



HAL
open science

Modéliser la diversité au cours du temps pour comprendre le contexte de l'utilisateur dans les systèmes de recommandation

Amaury L'Huillier

► **To cite this version:**

Amaury L'Huillier. Modéliser la diversité au cours du temps pour comprendre le contexte de l'utilisateur dans les systèmes de recommandation. Intelligence artificielle [cs.AI]. Université de Lorraine, 2018. Français. NNT: 2018LORR0182 . tel-01976800v2

HAL Id: tel-01976800

<https://theses.hal.science/tel-01976800v2>

Submitted on 19 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Modéliser la diversité au cours du temps pour comprendre le contexte de l'utilisateur dans les systèmes de recommandation

THÈSE

présentée et soutenue publiquement le 20 novembre 2018

pour l'obtention du

Doctorat de l'Université de Lorraine
(mention informatique)

par

Amaury L'Huillier

Composition du jury

<i>Rapporteurs :</i>	Max Chevalier	Professeur, Université Paul Sabatier
	Catherine Berrut	Professeur, Université Grenoble Alpes
<i>Examineurs :</i>	Miguel Couceiro	Professeur, Université de Lorraine
	Francis Rousseaux	Professeur, Université de Reims Champagne-Ardenne
<i>Directrice :</i>	Anne Boyer	Professeur, Université de Lorraine
<i>Co-directeur :</i>	Sylvain Castagnos	Maître de conférences, Université de Lorraine

Mis en page avec la classe thesul.

Table des matières

Introduction	1
1 Introduction générale	1
2 Problématique	5
3 Contributions	8
3.1 Modélisation de la diversité au cours du temps	8
3.2 Nouvelle définition du contexte et conception d'un algorithme de détection de ces contextes : DANCE	9
3.3 L'évolution temporelle de la diversité et les contextes implicites au service de la recommandation	11
4 Présentation du plan de thèse	12
Chapitre 1	
État de l'art	13
1.1 Les systèmes de recommandation	15
1.1.1 Formalisation du problème de la recommandation	15
1.1.2 Les principales familles d'algorithmes	16
1.1.3 Évaluation des recommandations	20
1.1.4 L'utilisateur : un humain derrière la machine	23
1.1.5 Discussion	26
1.2 Le contexte	26
1.2.1 Définition et intérêts	26
1.2.2 Intégration du contexte dans le processus de recommandation	29
1.2.3 Limites	35
1.2.4 Discussion	36
1.3 La diversité	37
1.3.1 Définitions et intérêts	37
1.3.2 Intégration de la diversité dans la recommandation	39
1.3.3 Le besoin en diversité	41

1.3.4	Diversité, nouveauté et serendipité	43
1.4	Bilan	45

Chapitre 2	
Jeux de données utilisés	47

2.1	Corpus Lastfm	48
2.2	Corpus Deezer	49
2.3	Attributs collectés	49
2.4	Limites	52

Chapitre 3	
Modéliser l'évolution temporelle de la diversité	55

3.1	Principe	55
3.2	Formalisme	57
3.3	Résultats de la modélisation temporelle de la diversité	62
3.3.1	Observations sur l'évolution temporelle de la diversité	62
3.3.2	Influences des paramètres sur la modélisation	65
3.3.3	Adaptation aux items de natures différentes	71
3.4	Discussion	71

Chapitre 4	
Diversité et contexte : DANCE	73

4.1	Contexte implicite	73
4.2	Détection des contextes implicites	75
4.2.1	Performances <i>a posteriori</i>	77
4.2.2	Performances en temps réel	86
4.2.3	Robustesse du modèle DANCE	92
4.2.4	Amélioration des performances	98
4.3	Discussion	103

Chapitre 5	
Étude utilisateur : exploration des relations existantes entre les attributs des items et le contexte	107

5.1	But de l'étude	107
5.2	Protocole	108
5.3	Résultats et discussions	112

Chapitre 6**Recommandations, diversité et contextes implicites****123**

6.1	Approches de recommandation	123
6.1.1	Recommandation collaborative à base de CI	124
6.1.2	Recommandation basée sur le contenu des CI	132
6.1.3	Anticiper les besoins des utilisateurs	134
6.2	Présenter les recommandations	136
6.2.1	Explication des recommandations	136
6.2.2	Les actions des utilisateurs	140
6.3	Discussions	142

Chapitre 7**Conclusions et perspectives**

7.1	Conclusions	147
7.2	Perspectives	149
7.2.1	Perspectives à court terme	149
7.2.2	Perspectives à moyen terme	150

Annexes**153****Annexe A Boîtes à moustaches et tests de comparaisons multiples, chapitre 5 153**

A.0.1	Comparaisons multiples du test de Kruskal-Wallis	160
-------	--	-----

Bibliographie**189**

Table des figures

1.1	Matrice utilisateur-item permettant de représenter les préférences des utilisateurs	15
1.2	Exemple de modèles de recommandation hybrides [Burke, 2002]	20
1.3	Adaptation du modèle <i>TAM</i> par [Jones, 2010]	23
1.4	Matrice permettant de représenter les préférences contextuelles des utilisateurs en lien avec l’environnement et le média [Adomavicius and Tuzhilin, 2011]	30
1.5	Caractère observable des dimensions contextuelles [Adomavicius and Tuzhilin, 2011]	32
1.6	Représentation hiérarchique de l’information contextuelle	33
1.7	Pré-filtrage, post-filtrage et filtrage contextuel [Adomavicius and Tuzhilin, 2011]	34
1.8	Modèle temporel de similarité diversité [Castagnos et al., 2010]	42
3.1	Illustration du fonctionnement de notre modèle	56
3.2	Impact du coefficient sur la formule de similarité	60
3.3	Évolution de la diversité au cours du temps de l’attribut “Danceability”. Corpus de Deezer, utilisateur 1, historique de taille 3, coefficient α fixé à 10	63
3.4	Évolution de la DR pour 3 attributs et DR totale. Corpus de Lastfm, utilisateur 4, historique de taille 3, coefficient α fixé à 10	63
3.5	Comportements remarquables de l’évolution de la diversité. Corpus de Lastfm, utilisateur 8, historique de taille 3, coefficient α fixé à 10	64
3.6	Évolution de la DR et de la dérivée de DR (tous attributs compris). Corpus de Deezer, utilisateur 200, historique de taille 3	65
3.7	Impact des données manquantes. Corpus de Lastfm, utilisateur 8, historique de taille 3, coefficient α fixé à 10	66
3.8	Impact du coefficient α sur l’évolution de la DR. Corpus de Lastfm, utilisateur 25, historique de taille 3	67
3.9	Impact de la taille de l’historique. Corpus de Deezer, utilisateur 50, historique de taille 3, coefficient α fixé à 10	68
3.10	Schéma représentant les deux coefficients d’éloignement	68
3.11	Impact des coefficients d’éloignement. Corpus de Lastfm, utilisateur 2, historique de taille 3, coefficient α fixé à 10	69
3.12	Impact du poids des attributs. Corpus de Lastfm, utilisateur 2, historique de taille 3, coefficient α fixé à 10	70
4.1	Exemple de découpage en contextes implicites ¹	75
4.2	Exemple d’utilisation de la méthode <i>change point analysis</i> [Killick and Eckley, 2014]	76
4.3	Détection <i>a posteriori</i> et en temps réel	77
4.4	Détections obtenues sans le paramètre τ	78
4.5	Contextes implicites détectés si $\tau = 0,15$	78

4.6	Répartition des tailles de contextes sur le corpus de Lastfm	82
4.7	Exemple de découpage	87
4.8	Répartition des tailles de contextes implicites obtenue sur le corpus de Lastfm	89
4.9	Performance de détection en fonction du nombre de données manquantes sur le corpus de Lastfm	93
4.10	Performance de détection en fonction du nombre de données manquantes sur le corpus de Lastfm	94
4.11	Performances et nombre de générations de l’algorithme génétique	99
5.1	Capture d’écran de la phase d’annotation sur l’opinion et les émotions	110
5.2	Capture d’écran de la phase d’annotation sur les situations contextuelles	110
5.3	Répartition du nombre d’annotations des musiques	112
5.4	Exemples de boîtes à moustaches obtenues	114
5.5	Émotions ressenties	118
5.6	Émotions initiatrices	118
5.7	Activité	118
5.8	Jour	118
5.9	Énergie	118
5.10	Environnement	118
5.11	Lieu	118
5.12	Météo	118
5.13	Saison	119
5.14	Semaine	119
5.15	Social	119
5.16	Pourcentage de réponses pour chacune des affirmations	120
6.1	Schéma du fonctionnement de l’approche de recommandation collaborative basée sur les CI	126
6.2	Cas d’un attribut variant fortement au sein d’un contexte implicite dans le parcours d’un utilisateur sur un réseau social professionnel	133
6.3	Représentation des transitions de CI à l’aide d’un processus Markovien	135
6.4	Explications des recommandations sur le site Amazon ²	137
6.5	Explications des recommandations sur le site Amazon	139
6.6	Explications des recommandations sur le site Amazon	140
6.7	Schéma des explications des choix pouvant être données par un utilisateur	141

Introduction

1 Introduction générale

En l'espace d'un demi-siècle, Internet est passé d'une théorie basée sur la transmission de paquets permettant à deux ordinateurs distants de communiquer au plus impressionnant système d'information que l'humanité ait connue. Pour prendre conscience de l'ampleur de la situation, il suffit de regarder les chiffres colossaux concernant ses usages. Durant le temps nécessaire pour lire les premières phrases de cette introduction, plus de 643 000 Go de données ont été échangés, près de 900 000 requêtes Google ont été effectuées et plus d'un million de vidéos ont été visionnées³. . . Si nous avions stocké depuis l'aube de l'humanité, l'ensemble des écrits et des communications jusqu'en 2003, cela représenterait 5 milliards de gigabits. Ce nombre représente actuellement la quantité d'informations générée en deux jours sur Internet⁴. Au-delà de ces nombres impressionnants qui ne cessent de croître chaque seconde, se poser la question "Qu'est-ce qu'on ne peut pas trouver sur Internet ?" est tout autant révélateur. En effet, si notre question respecte les limites éthiques et juridiques imposées par notre société, force est de constater que la réponse est : "tout". Par exemple, n'importe quelle personne pouvant se connecter à Internet peut accéder à des cours en ligne proposés par une université prestigieuse, effectuer un achat sur un site de e-commerce depuis son smartphone, accéder de manière illimitée à des services multimédias à la demande (jeux vidéo, musique, vidéo), *etc.*

Malgré l'offre disponible sur Internet, les utilisateurs ne parviennent pas toujours à trouver ce qu'ils recherchent et peuvent passer à côté d'informations pertinentes tant le volume de données disponibles est considérable. Selon les estimations réalisées par [Bosch et al., 2016], plus de 4,46 milliards de pages web étaient disponibles à la moitié de l'année 2016. Par conséquent, dégager l'information pertinente pour un utilisateur s'apparente à trouver une aiguille dans une meule de foin. Ce problème est d'autant plus complexe que la difficulté à prendre une décision augmente avec le nombre de choix possibles, en d'autres termes, plus un utilisateur a de choix, plus il aura de difficulté à se décider [Schwartz, 2005].

Certains outils tels que les moteurs de recherche ou la recherche par critères ont été mis au point afin d'assister les utilisateurs dans leurs démarches de recherche. Ces outils reposent sur la création manuelle d'une requête pouvant utiliser des opérateurs logiques (et, ou, *etc.*) et/ou sur le choix d'un ensemble de critères en rapport avec les caractéristiques (les métadonnées) des ressources (également appelées items). Par exemple, il est possible d'effectuer la recherche "appareil photo (et) compact (et) voyage" afin d'espérer obtenir des résultats concernant des appareils photo compacts adaptés aux voyages. Une fois la requête formulée (et/ou les critères définis) et la recherche effectuée, l'utilisateur peut consulter les résultats qui sont alors proposés. Si ce dernier n'est pas satisfait, il doit (s'il le peut) modifier sa requête afin d'obtenir d'autres résultats ou en-

3. <http://www.internetlivestats.com/>

4. Dominique Cardon, À quoi rêvent les algorithmes, nos vies à l'heure des *big data*, Édition du Seuil et La République des Idées, octobre 2005.

core affiner ses critères de recherche si les résultats sont trop nombreux ou pas assez pertinents. Ce processus de reformulation qui s'apparente à un processus d'essai-erreur est contraignant pour les utilisateurs mais, également néfaste du point de vue du système. Premièrement, les utilisateurs doivent fournir un effort cognitif notable afin de créer la première requête et devront selon les résultats obtenus repenser leur requête ou alors se contenter d'informations de moindre qualité (dans le sens où les ressources pertinentes existent bien, mais n'ont pas été atteintes). Le choix des mots-clés et/ou des critères étant déterminant pour obtenir les résultats attendus, réussir à formuler au premier essai la requête permettant de trouver les informations recherchées n'est pas une chose aisée et demande un certain niveau d'expérience et de réflexion. De plus, reformuler une requête nécessite de comprendre les caractéristiques de la recherche (opérateurs, mots-clés, ordre des mots-clés) ayant mené à des résultats non satisfaisants et ensuite de tenter de créer une nouvelle requête permettant d'obtenir des résultats plus opportuns. Deuxièmement, sachant qu'il existe de nombreux sites internet proposant les mêmes services (sites d'actualités, recettes de cuisine, e-commerce, forums de discussion, . . .), un utilisateur n'hésitera pas à quitter un site web pour un autre si ce dernier ne lui offre pas rapidement satisfaction. Dans notre cas, la satisfaction d'un utilisateur est représentée par le fait qu'il parvient à trouver rapidement et efficacement l'information qu'il est venu chercher.

Bien qu'indispensables afin d'interagir avec un système d'information quel qu'il soit (pages web, articles de commerce, ressources multimédias, ressources pédagogiques, *etc.*), ces outils traditionnels se heurtent à certaines limites. L'une des principales difficultés, malgré les progrès actuels dans le domaine du traitement automatique des langues, est liée au problème de la barrière sémantique. Cette barrière sémantique est la raison pour laquelle il est difficile de s'adresser à un système informatique de la même manière qu'à un conseiller dans un magasin ou à un bibliothécaire. La prise en charge du langage naturel par un système informatique est au cœur des recherches actuelles et les applications grand public en émergeant remportent un franc succès à travers les assistants intelligents qui commencent à s'immiscer dans notre quotidien^{5 6 7}. Cependant, bien que ces assistants intelligents soient capables de répondre à des questions complexes telles que "A quelle heure est la prochaine séance d'Avatar au cinéma de Nancy?" ou encore "Quel temps fera-t-il demain à Nancy?", ils seront tout aussi limités qu'un moteur de recherche lorsqu'il s'agira de trouver quel est l'appareil photo le plus adapté à un utilisateur ayant des besoins spécifiques. Même si l'interaction avec le système se fait par la parole, la compréhension des liens existant entre les caractéristiques d'un appareil photo et son utilisation requiert une exploitation importante des connaissances. Ainsi, même si ces assistants intelligents offrent un nouveau mode d'interaction par le biais de la reconnaissance vocale, le traitement de la masse d'information disponible pour dégager les informations pertinentes constitue toujours un défi scientifique majeur.

Afin de sortir de ce paradigme requête/résultats, les premiers systèmes de recommandation (SR) [Resnick et al., 1994] ont été mis au point. Leur but est simple, proposer aux utilisateurs des ressources susceptibles de les intéresser sans que ces derniers aient besoin de les rechercher par eux-mêmes. Les recommandations peuvent alors jouer différents rôles : suggérer des ressources (films, musiques, produits d'un catalogue, pages web. . .), proposer des alternatives par rapport à la ressource en cours de consultation (proposer un appareil photo ayant des caractéristiques proches de celui consulté, mais appartenant à une autre marque), proposer des ressources complémentaires (recommander une housse protectrice à quelqu'un qui consulte des appareils photo),

5. <https://support.microsoft.com/fr-fr/help/17214/windows-10-what-is>

6. <http://www.apple.com/fr/ios/siri/>

7. <http://www.alexa.com/>

etc. Ces systèmes offrent ainsi une nouvelle manière d'accéder à l'information puisque ce ne sont plus les utilisateurs qui recherchent l'information, mais c'est l'information qui est proposée aux utilisateurs.

En vue de fournir des recommandations personnalisées, les SR doivent s'adapter en fonction des ressources, des utilisateurs et de leurs interactions avec le système. Si par exemple nous nous plaçons dans le cas d'un SR basé sur les traces d'usages, quatre grandes étapes sont nécessaires à son opérationnalisation [Castagnos, 2008] :

- collecter les traces d'usage des utilisateurs (notes attribuées aux items, temps passé sur un item, ajout à une liste de favoris...);
- transformer et synthétiser ces traces sous forme d'information de plus haut niveau (profils utilisateurs);
- utiliser les profils utilisateurs afin de déterminer les recommandations adaptées à chaque profil à l'aide de méthodes d'apprentissage automatique;
- présenter les recommandations aux utilisateurs de manière appropriée.

Chaque système de recommandation peut varier sur une ou plusieurs des étapes ci-dessus et, de ce fait, il existe de nombreuses approches de recommandation ayant chacune leurs forces, leurs faiblesses et leurs perspectives [Adomavicius and Tuzhilin, 2005, Candillier et al., 2007] [Jannach et al., 2010, Lops et al., 2011, Zhang et al., 2017]. Par exemple, certains systèmes exploitent les caractéristiques des ressources alors que d'autres mettent à profit la communauté d'utilisateurs du système afin de calculer les recommandations. On parle alors de SR à base de contenu [Lops et al., 2011] dans le premier cas, et de SR sociaux [Su and Khoshgoftaar, 2009] dans le second cas. Un système basé sur le contenu exploite les caractéristiques (métadonnées) des ressources afin de trouver des ressources ayant des caractéristiques similaires à celles ayant été appréciées par l'utilisateur que l'on cherche à recommander. Ces systèmes reposent donc sur l'hypothèse suivante : "si un utilisateur apprécie une ressource, les ressources possédant des caractéristiques similaires seront également appréciées". Un SR social repose quant à lui sur l'hypothèse suivante : "les utilisateurs ayant des préférences cohérentes entre elles, les items plaisant à un utilisateur plairont également aux utilisateurs qui lui ressemblent". Les SR sociaux vont alors déterminer dans un premier temps les utilisateurs qui sont similaires à l'utilisateur courant, afin de lui proposer dans un second temps des ressources que ces derniers ont appréciées. Une autre approche sociale consiste à utiliser la confiance existante entre les utilisateurs ainsi que sa propagation au sein de la communauté d'utilisateurs du système afin de fournir des recommandations [Massa and Avesani, 2007, Haydar et al., 2015]. Les SR sociaux sont également appelés SR collaboratifs du fait qu'ils exploitent la communauté d'utilisateurs d'un système pour fournir les recommandations. Ces deux approches (à base de contenu et sociale) sont les plus couramment utilisées et peuvent être combinées entre elles afin d'obtenir des systèmes hybrides [Burke, 2002] permettant de profiter des avantages de chacune des deux approches tout en minimisant leurs faiblesses respectives. Rapidement adoptés par les utilisateurs, les SR sont devenus des outils indispensables afin de les assister dans leur démarche d'accès à l'information. Par ailleurs, certains acteurs du secteur ont même placé les systèmes de recommandation au cœur de leur service, par exemple dans le domaine de la musique⁸, du cinéma (films et séries)⁹ ou encore des actualités¹⁰.

Au fil des années, les SR mis au point ont permis d'atteindre une très haute précision dans la tâche de prédiction de l'intérêt qu'un utilisateur peut avoir pour des ressources. Pour un

8. <http://www.pandora.com>, <https://support.deezer.com/hc/fr/articles/201331091-Flow>

9. <https://www.netflix.com/>

10. <https://yahoo.tumblr.com/post/96977561949/a-new-publisher-solution-yahoo-recommends>

SR, calculer les recommandations revient à prédire les intérêts d'un utilisateur pour des ressources et ensuite à proposer les ressources ayant le plus grand intérêt. Cet intérêt est généralement exprimé sous la forme d'une note, mais certaines pratiques issues des réseaux sociaux ont également amené les utilisateurs à s'exprimer sous une forme binaire comme le bouton "j'aime"/"je n'aime pas"¹¹, ou encore via la notation "pouce vert"/"pouce rouge"¹². Afin d'évaluer la qualité d'un SR (c'est-à-dire d'évaluer la qualité de ses prédictions), il est courant de comparer l'intérêt prédit par le système pour un couple utilisateur/ressource et l'intérêt réel (exprimé) de cet utilisateur pour cette ressource. Plus un système minimise cette erreur, plus il est précis et donc capable de prédire les préférences d'un utilisateur. Cependant, un système peut être extrêmement précis sans pour autant satisfaire pleinement l'utilisateur. Par exemple, un SR de musiques ou de films qui ne recommande que des items populaires ou encore des items pour lesquels les préférences sont déjà connues aura une grande précision. Les recommandations proposées seront cependant sclérosées et ne suffiront pas à satisfaire l'utilisateur sur le long terme. Bien qu'il soit pertinent de proposer une musique déjà appréciée à un utilisateur¹³, dans certains cas, proposer uniquement des items déjà connus n'apportera que peu de satisfaction chez l'utilisateur. Par ailleurs, certains travaux de la littérature ont démontré que la précision n'est pas le seul facteur déterminant la qualité d'une recommandation [McNee et al., 2006, thenextweb.com, 2012] et qu'une différence de précision de 10 % entre deux algorithmes de recommandation n'est pas perçue par les utilisateurs [Jones, 2010]. Très récemment, Netflix qui a organisé de nombreux concours visant à améliorer la précision des recommandations a abandonné son système de notation par notes (étoiles) au profit d'un système binaire de pouce haut/pouce bas [thenextweb.com, 2017], démontrant alors la place relative de la précision dans les recommandations ainsi que la difficulté de l'évaluation de l'intérêt par l'utilisateur. Dans le même temps, de nombreux travaux ont mis en avant la nécessité de prendre en compte les facteurs humains (contexte, besoin en diversité, culture, tempérament...) dans le calcul des recommandations [Jones, 2010, Chen et al., 2013, Castagnos et al., 2013b]. Ainsi, un utilisateur ne souhaitera sans doute pas regarder seul des films pour la famille le samedi soir et des films d'horreur le dimanche après-midi alors qu'il est en famille. Même si les films proposés peuvent lui plaire dans l'absolu, son contexte social (dont les modalités sont dans ce cas "être seul" ou "en famille") l'amènera à avoir des préférences différentes. Les SR doivent alors être en mesure d'adapter leurs recommandations non seulement par rapport aux préférences des utilisateurs, mais également par rapport à leurs contextes. Le contexte est un facteur complexe qui possède de nombreuses dimensions qui peuvent être exclusives dans certains cas ou composées dans d'autres. Dans l'exemple cité précédemment, nous n'avons mentionné que le contexte social (sa famille) et temporel (soir, après-midi) de l'utilisateur. Le contexte peut notamment être météorologique, émotionnel ou lié à l'activité de l'utilisateur, et sa prise en compte dans la phase de recommandation peut être déterminante afin de proposer les bonnes recommandations. Du fait de son rôle décisif dans la pertinence des recommandations, le contexte est passé d'une dimension à prendre en compte pour améliorer les recommandations [Leake and Scherle, 2001] à un champ d'investigation à part entière [Adomavicius and Tuzhilin, 2011] appelé CARS (acronyme de *Context-Aware Recommender Systems*). Cependant, les méthodes actuelles visant à intégrer le contexte dans le processus de recommandation se heurtent à certaines difficultés concernant le respect de la vie privée des utilisateurs. Afin de s'adapter au contexte social de l'utilisateur de l'exemple précédent, le système doit alors être en mesure de savoir quand, où et avec qui l'uti-

11. <https://www.facebook.com/>

12. <https://www.youtube.com>, <https://www.pandora.com>

13. Cet exemple fait d'ailleurs partie des spécificités de certains domaines de recommandation que nous développerons dans les parties suivantes de ce manuscrit.

lisateur réalise une tâche (visionner un film dans notre cas). Or, ces informations appartiennent à la vie privée de l'utilisateur et bien que leur exploitation permette d'améliorer la qualité des recommandations, des méthodes alternatives permettant la prise en compte du contexte doivent être développées. Dans cette thèse, nous proposerons une nouvelle approche de recommandation contextuelle respectueuse de la vie privée des utilisateurs intégrant la prise en compte d'un autre facteur humain, à savoir le besoin en diversité.

2 Problématique

Prendre en compte le contexte de l'utilisateur dans le processus de recommandation s'est imposé comme étant indispensable afin de proposer les bonnes recommandations au moment opportun [Adomavicius et al., 2011b, Verbert et al., 2012]. La définition du contexte couramment utilisée dans le domaine de la recommandation est celle donnée par [Abowd et al., 1999] comme étant "l'ensemble des informations permettant de caractériser la situation d'une entité". Dans le cas des SR, l'"entité" en question fait bien évidemment référence à l'utilisateur. Afin de caractériser le contexte d'un utilisateur, les approches traditionnelles de l'état de l'art cherchent à définir les dimensions permettant de décrire la situation de l'utilisateur [Zheng, 2016]. Par exemple, les conditions météorologiques (température, ensoleillement...), les conditions sociales (seul, en couple, entre amis...), l'état mental (heureux, triste, mélancolique...) sont autant de facteurs pouvant définir le contexte de l'utilisateur. Dans le but de collecter ces informations, il est soit possible de demander directement à l'utilisateur de préciser son contexte^{14 15} (par exemple en demandant à l'utilisateur de spécifier son activité, son humeur...), soit de procéder à la détermination du contexte en utilisant des capteurs connectés au système (ce qui est de plus en plus possible avec l'utilisation de *smartphones* et de tablettes) ou toute autre information à laquelle le système peut accéder. Par exemple, [Unger, 2015] utilisent l'accéléromètre, le capteur de lumière et le microphone d'un *smartphone* afin de définir le contexte de l'utilisateur qui écoute de la musique. Un capteur de bruit ambiant saturé et un accéléromètre indiquant un mouvement constant peuvent alors être révélateurs d'un contexte de soirée dansante. À l'ère des objets connectés, des informations peuvent également être obtenues via différents appareils tels que bracelets connectés portés par les utilisateurs et dont les données relevées peuvent être partagées entre différents services. Ainsi, si les musiques sont lues à partir d'un ordinateur au travers une application de musique sur laquelle est également connecté le *smartphone* de l'utilisateur, il est possible par un recoupement des données de déterminer le contexte de l'utilisateur. L'utilisation directe de ces informations n'est pas la seule option pour définir le contexte de l'utilisateur, il est possible de recouper certaines informations pour améliorer la définition du contexte. Entre autres, de nombreuses approches utilisent la position géographique des utilisateurs afin d'accéder à la météo par exemple [Adomavicius and Tuzhilin, 2011]. Une fois que le contexte d'un utilisateur a été défini, diverses techniques sont employées pour définir les recommandations qui seront adaptées au contexte. Parmi les plus répandues, nous pouvons citer la factorisation de matrice [Baltrunas et al., 2011b], l'utilisation de tenseurs [Hidasi and Tikk, 2012], les approches basées sur les graphes [Wu et al., 2015], la création d'items contextualisés [Baltrunas and Ricci, 2014].

Même si l'amélioration de la qualité des recommandations reposant sur l'exploitation de ces facteurs contextuels est à chaque fois démontrée en comparaison avec les méthodes n'exploitant pas cette dimension, l'utilisation de telles informations peut s'avérer problématique pour plusieurs raisons. Premièrement, peu importe la manière d'y accéder, les facteurs contextuels sont

14. <https://www.spotify.com>

15. <https://play.google.com>

constitués à partir de données sensibles qui relèvent de la vie privée de l'utilisateur. De ce fait, leur exploitation est sujette à débat et plusieurs études ont démontré la réticence des utilisateurs quant à l'utilisation de telles informations [Knijnenburg and Kobsa, 2013], [Cranor, 2005]. Deuxièmement, comme il est précisé dans la définition du contexte donné par [Abowd et al., 1999], il faut que la connaissance de ces informations s'avère pertinente. Or, plusieurs études ont démontré que tous les facteurs contextuels collectés et exploités à des fins de recommandations contextuelles n'étaient pas nécessairement pertinents [Baltrunas et al., 2011a], [Odic et al., 2013]. De plus, l'utilisation de facteurs contextuels non pertinents augmente la complexité de l'espace de recherche [Braunhofer and Ricci, 2016]. Troisièmement, les facteurs contextuels ne sont pas toujours accessibles et les systèmes qui en dépendent peuvent devenir désuets si les données sont partiellement ou totalement non observables [Adomavicius and Tuzhilin, 2011]. Quatrièmement, les informations contextuelles récupérées et la manière dont elles sont exploitées sont souvent dépendantes du domaine dans lequel est implanté le SR. Cela apporte donc un caractère *ad hoc* à ces approches qui seront la plupart du temps non transposables/extensibles d'un domaine applicatif à un autre. Enfin, le contexte est connu pour être une dimension très dynamique [Adomavicius and Tuzhilin, 2011], imposant de fait une collecte continue des informations nécessaires à la mise à jour des facteurs contextuels dans le but de s'adapter à toute évolution de ces derniers. Une telle collecte soulève deux problèmes touchant dans un cas les utilisateurs et dans l'autre le système. D'une part, une récupération constante et massive d'informations sur l'utilisateur soulève de nouveau la question du respect de la vie privée. D'autre part, un tel traitement implique nécessairement une charge de calcul non négligeable pour des systèmes devant être les plus réactifs possible.

En résumé, le contexte est primordial afin de fournir des recommandations pertinentes en vue d'assister l'utilisateur. Cependant, les méthodes actuelles reposent pour la plupart sur la collecte d'informations des utilisateurs et font face à certains problèmes que nous avons mentionnés. Mettre au point une approche alternative ayant un but similaire constitue donc le cadre de cette thèse. Pour ce faire, nous considérons que le contexte d'un utilisateur peut être défini non plus uniquement sur la base des dimensions contextuelles qui le caractérisent, mais sur la base des caractéristiques des ressources consultées au sein de ce contexte. Plus particulièrement, nous nous intéresserons à l'évolution temporelle des caractéristiques des ressources consultées par les utilisateurs, et ce au travers de la diversité. Nous expliquons dans la suite de cette section en quoi l'analyse du besoin en diversité dans le parcours des utilisateurs peut être une alternative à la définition et la prise en compte du contexte dans le processus de recommandation.

La diversité a été définie pour la première fois dans les SR comme étant la mesure opposée à la similarité [Smyth and McClave, 2001]. Plus deux ressources sont diverses et moins elles partagent de similitudes. La diversité est calculée sur la base des attributs caractérisant les ressources. Par exemple, deux films peuvent être comparés sur leur genre, leur durée, leur casting et leurs réalisateurs. . . . Il faut bien comprendre que diversité et précision ne sont pas deux dimensions opposées. Il est possible de proposer deux recommandations qui sont diverses entre elles en termes de caractéristiques, mais qui sont tout autant précises par rapport au besoin de l'utilisateur. La diversité a par la suite été présentée comme étant un moyen efficace de réagir face à une mauvaise recommandation [McGinty and Smyth, 2003]. En effet, si un utilisateur n'apprécie pas une recommandation, c'est prendre un risque que de lui recommander une ressource qui lui est très similaire, même si la prédiction d'intérêt calculée par le SR pour cette ressource est élevée. L'approche proposée cherche donc à maximiser la diversité entre la première et la seconde recommandation, tout en s'assurant que la seconde recommandation soit pertinente pour cet utilisateur. Par la

suite, la diversité est rapidement devenue une dimension importante et son introduction dans les listes de recommandations afin de proposer des alternatives a fait l'objet de nombreuses publications [Ziegler et al., 2005], [Radlinski and Dumais, 2006], [Boim et al., 2011]. Certaines études démontrent toutefois que la diversité, bien que nécessaire, peut susciter la méfiance des utilisateurs si elle n'est pas proposée au bon moment ou dans les bonnes proportions [Lathia, 2010, Castagnos et al., 2013a]. Inversement, il est démontré que l'absence de diversité dans certains cas peut provoquer une frustration chez l'utilisateur [Lathia et al., 2010, Zhang and Hurley, 2008]. Par exemple, un système ayant déterminé l'intérêt d'un utilisateur pour un groupe de musique ne doit pas exclusivement recommander des musiques de ce groupe afin de ne pas restreindre les recommandations proposées à l'utilisateur.

[Castagnos et al., 2010] ont démontré que le niveau de diversité optimal à proposer dans les recommandations n'est pas figé, mais doit évoluer au cours du temps. Dans leurs travaux, ils soulignent dans le domaine du e-commerce que la diversité doit être proposée en des proportions différentes en fonction de la phase du processus d'achat dans lequel se trouvent les utilisateurs. En effet, la quantité de diversité proposée doit être importante au début du processus afin de permettre à l'utilisateur de découvrir les alternatives possibles. Le système ne connaissant pas encore bien les besoins/attentes de l'utilisateur, il ne peut pas prendre le risque de le cantonner à certains produits ayant des caractéristiques particulières. Ensuite, au fur et à mesure que l'utilisateur navigue à travers les différents items et que le système comprend/définit les besoins de l'utilisateur, la similarité entre les items proposés doit augmenter. Cela se traduit par une diminution de la diversité permettant à l'utilisateur d'affiner son choix. Enfin, lorsque la décision d'achat approche, la quantité de diversité proposée à l'utilisateur doit accroître de nouveau afin d'augmenter sa confiance. En comparant son choix par rapport à des items ayant des caractéristiques différentes, il s'assure qu'il a choisi l'item qui répond bien à ses attentes. Maîtriser la quantité de diversité est alors fondamental pour proposer des recommandations qui seront utiles pour l'utilisateur en fonction de son besoin. La diversité est donc un facteur important dans la qualité des recommandations et son utilisation doit être maîtrisée en fonction du contexte et des utilisateurs. Cependant, même si diverses méthodes permettent de l'introduire dans les listes de recommandations, aucun modèle de l'état de l'art ne propose un moyen de mesurer l'évolution de la diversité des ressources consultées au cours du temps. Un tel modèle permettrait de connaître le besoin en diversité des utilisateurs en temps réel et d'adapter le niveau de diversité des recommandations en conséquence. S'il est démontré que le niveau de diversité souhaité par l'utilisateur est dépendant de son contexte¹⁶, il est également intéressant de se demander si l'analyse de l'évolution de la diversité au cours du temps ne pourrait pas être un moyen de définir le contexte de l'utilisateur. En effet, il est raisonnable de penser que, si le niveau de diversité change d'un contexte à l'autre, une variation de diversité est sous-jacente d'un changement de contexte. Il n'est alors plus nécessaire d'interroger l'utilisateur sur son contexte ou de définir ce dernier explicitement, mais uniquement d'analyser l'évolution des caractéristiques des items consultés afin de détecter des variations notables de diversité.

16. Un utilisateur qui cherche à se faire une idée du contenu d'un site web en termes de contenu consultera des items très divers et c'est ce niveau de diversité qui caractérisera son contexte.

Dans cette thèse, nous nous intéresserons à la problématique suivante : **est-il possible de fournir des recommandations contextuelles en se basant uniquement sur l'analyse de l'évolution temporelle de la diversité ?**

Afin d'apporter une réponse à cette problématique, nous nous fixons les objectifs suivants :

- Définir un modèle permettant de modéliser l'évolution temporelle de la diversité au cours du temps dans le parcours de consultations d'un utilisateur ;
- Proposer une nouvelle famille de contextes basée sur l'évolution de la diversité et une méthode d'identification de ces contextes ;
- Concevoir des approches de recommandation contextuelles.

Afin de tester les hypothèses que nous venons de présenter, nous devons utiliser des corpus de données qui respectent certaines conditions. Les données applicatives doivent former des séquences de consultations contiguës de ressources ayant un temps de consultation relativement faible. En effet, modéliser l'évolution de la diversité n'a de sens que si les consultations sont en lien les unes avec les autres. Cela est également important pour notre second objectif vis-à-vis duquel nous cherchons à déterminer le contexte sur la base des consultations effectuées. Ainsi, le domaine des films n'est pas le domaine applicatif de prédilection étant donné qu'il est peu probable qu'un utilisateur regarde plusieurs films à la suite, et donc qu'il reste dans le même contexte entre deux consultations. Enfin, les items consultés doivent posséder des attributs sur lesquels seront calculées les valeurs de diversité. Les réseaux sociaux professionnels, les sites d'offres d'emploi, les sites d'e-commerce ou d'e-éducation et les applications de musique sont des exemples de domaines applicatifs potentiels. Lorsque les internautes utilisent des sites web/applications de ces domaines, ils consultent de nombreux items et les préférences pour ces derniers sont dépendantes des contextes dans lesquels ils se trouvent. À titre d'exemple, un utilisateur qui souhaite animer une soirée dansante ne sera sans doute pas intéressé par son morceau de musique classique préféré bien qu'il l'écoute plusieurs fois par semaine.

Pour ces raisons, nous avons choisi d'utiliser des jeux de données issus d'applications musicales telles que la plateforme de musique à la demande Deezer¹⁷ et le site Last.fm¹⁸. Ces corpus sont constitués de 2 millions de consultations de musiques pour lesquelles nous avons obtenu 16 attributs détaillés dans le Chapitre 2. Nous avons choisi de travailler sur le domaine de la musique en ligne pour plusieurs raisons. Premièrement, le domaine de la musique en ligne est un domaine qui ne cesse d'attirer des utilisateurs ces dernières années et dans lequel la recommandation occupe une place majeure [Schedl et al., 2017]. Deuxièmement, les musiques possèdent la particularité de pouvoir être consultées de nombreuses fois par un même utilisateur, ce qui constitue du point de vue de la recommandation, un réel défi. En effet, il n'est pas seulement question de proposer à l'utilisateur un item qu'il peut apprécier, il est nécessaire de comprendre ses besoins afin de lui proposer également des morceaux qu'il connaît déjà. Troisièmement, il existe une relation importante entre le contexte d'un utilisateur et la musique qu'il souhaite écouter. Par exemple, peu de mariages ont été célébrés sur la musique *Sonate pour piano no 2 op. 35* de Chopin plus connue sous le nom de Marche funèbre. Ainsi, prendre uniquement en compte les préférences des utilisateurs n'est pas suffisant pour proposer des recommandations. La prise en compte du contexte des utilisateurs est primordiale afin de proposer des recommandations pertinentes. De plus, l'utilisation de plateforme de *streaming* de musique permet aux utilisateurs

17. <https://www.deezer.com/>

18. <https://www.last.fm/>

de consulter leurs musiques aux quatre coins du globe et donc dans de nombreux contextes auxquels les SR doivent s'adapter. Enfin, les musiques sont des ressources pour lesquelles nous avons pu obtenir de nombreuses caractéristiques sur lesquelles nous avons pu baser nos calculs de diversité.

3 Contributions

3.1 Modélisation de la diversité au cours du temps

La première contribution de cette thèse est la formalisation et la conception à notre connaissance du premier modèle de l'état de l'art permettant de modéliser en temps réel l'évolution de la diversité des ressources consultées par les utilisateurs. La diversité est calculée à chaque nouvelle consultation par rapport aux ressources précédemment consultées (appelées historique) en utilisant la métrique de diversité relative. La diversité relative est une métrique introduite par [Ziegler et al., 2005] permettant de quantifier la diversité apportée par une ressource relativement à un ensemble de ressources. Initialement utilisée afin de contrôler le niveau de diversité de chaque item d'une liste de recommandations par rapport aux autres items de cette liste, nous avons modifié le cadre d'utilisation de cette métrique. En calculant la diversité relative pour chaque nouvelle consultation d'un utilisateur par rapport à ses consultations précédentes, il est possible de modéliser l'évolution de la diversité au cours du temps. En fonction, de cette évolution, nous pouvons savoir si l'utilisateur spécifie ou diversifie ses consultations, soit en analysant indépendamment chaque attribut, soit en analysant un ensemble d'attributs. Ce modèle est générique et le formalisme que nous avons proposé permet un déploiement rapide dans n'importe quel domaine applicatif pour lequel il existe des séquences de consultations contiguës. Du fait de sa faible complexité algorithmique $O(n)$ (n étant la taille de l'historique utilisé dans le calcul de la diversité), notre approche peut être utilisée en temps réel rendant ainsi le modèle très réactif puisque la diversité de chaque nouvelle consultation est calculée presque instantanément. Outre les nombreux paramètres pouvant être employés dans le calcul de la diversité, notre modèle peut également fournir une modélisation de la diversité pour des ressources de natures différentes, mais possédant tout de même des attributs communs. Par exemple, sur un site de e-commerce, si un utilisateur consulte deux livres, puis un DVD, il est possible de calculer la diversité apportée par le DVD comparativement aux livres précédemment consultés sur la base du prix ou encore du genre. L'apport de ce modèle permet donc la validation du premier objectif fixé dans le cadre de cette thèse.

Cette contribution a fait l'objet de deux publications internationales :

- L'Huillier, A., Castagnos, S., and Boyer, A. (2014). Understanding Usages by Modeling Diversity over Time. volume 1181 of ACM UMAP 2014 Extended Proceedings, Aalborg, Denmark.
- L'Huillier, A., Castagnos, S., and Boyer, A. (2016a). Modéliser la diversité au cours du temps pour détecter le contexte dans un service de musique en ligne. *Revue des Sciences et Technologies de l'Information (TSI)*.

3.2 Nouvelle définition du contexte et conception d'un algorithme de détection de ces contextes : DANCE

Les travaux de [Castagnos et al., 2010] ont démontré dans le cas d'une démarche d'achat en ligne l'existence d'un lien entre le contexte dans lequel se trouve un utilisateur et la quantité de diversité consommée/désirée. Partant de ce constat, nous nous sommes alors demandé s'il était possible de détecter et de définir le contexte des utilisateurs en analysant l'évolution temporelle de la diversité. Pour répondre à cette question, nous avons procédé en deux étapes.

Dans un premier temps, nous avons démontré que les changements de contextes des utilisateurs pouvaient être déterminés sur la base de l'étude de l'évolution de la diversité. Dans ce but, nous avons proposé une nouvelle famille de contextes que nous appelons contextes implicites et que nous dissociions des contextes explicites. Les définitions que nous proposons sont les suivantes :

- **Contexte implicite** : ensemble des caractéristiques communes aux ressources consultées pendant un certain laps de temps correspondant à une situation donnée.
- **Contexte explicite** : ensemble des critères permettant de caractériser explicitement la situation dans laquelle se trouve l'utilisateur courant.

Nous considérons qu'un nouveau contexte implicite débute lorsqu'une variation significative de diversité caractérisée par le dépassement d'un seuil fixé par notre modèle est détectée. En nous basant sur cette définition, nous avons développé l'algorithme **DANCE** (acronyme pour *Diversity And Natural Context Elicitation*), dont le but est d'isoler les contextes implicites dans une séquence de consultations. Les expériences menées sur 2 corpus de musique ont mis en lumière une forte correspondance entre les changements de contextes implicites et certaines actions des utilisateurs telles que les changements de sessions. En utilisant le taux de recouvrement comme critère de performance de notre modèle, il est possible d'optimiser certains paramètres intervenant dans le calcul de la diversité et ainsi d'augmenter les performances de notre modèle. Les avantages de notre modèle par rapport aux modèles reposant sur l'exploitation du contexte explicite sont multiples. Premièrement, notre modèle se base sur notre approche de modélisation temporelle de la diversité et bénéficie des mêmes avantages. Le modèle **DANCE** est donc réactif et utilisable en temps réel étant donnée sa complexité faible ($O(n)$) et constante. Deuxièmement, comme le contexte implicite n'est défini que sur la base des attributs et de leurs évolutions, notre approche se veut respectueuse de la vie privée des utilisateurs. De plus, notre modèle s'avère également robuste car les performances de détection de contextes implicites ne se trouvent significativement impactées qu'à partir de 60 % d'informations manquantes (c'est-à-dire que certains attributs des items sont inconnus). Nous avons également confirmé le caractère adaptatif de notre modèle vis-à-vis d'items de natures différentes. Troisièmement, en plus de définir le contexte implicite à partir duquel les recommandations pourront être adaptées et contractualisées, les contextes implicites nous assurent un contrôle du niveau de diversité à proposer dans les recommandations. Ces derniers points feront l'objet des contributions discutées dans la section suivante. Cette troisième contribution a fait l'objet d'une publication internationale et d'une revue nationale :

- Castagnos, S., L’Huillier, A., and Boyer, A. (2015). Toward a robust diversity-based model to detect changes of context. In 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2015), Vietri sul Mare, Italy.
- L’Huillier, A., Castagnos, S., and Boyer, A. (2016b). The new challenges when modeling context through diversity over time in recommender systems. In Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP ’16, pages 341–344, New York, NY, USA. ACM.

Dans un second temps, nous avons cherché à mettre en avant lumière les relations existantes entre les attributs des items contenus dans les corpus utilisés dans ce manuscrit et de nombreux facteurs contextuels. Autrement dit, nous avons souhaité savoir si les attributs des musiques consultées sont dépendants du contexte explicite dans lequel se trouvent les utilisateurs. Les expériences développées dans ce but ont permis de valider l’hypothèse de cette relation, et de créer un jeu de données contenant plus de 1500 musiques pour lesquelles le contexte d’écoute souhaité est renseigné par des utilisateurs. Afin de permettre l’exploitation de ce corpus par d’autres travaux portant sur la recommandation contextuelle, nous avons mis à disposition de la communauté scientifique l’ensemble des données que nous avons pu collecter lors de cette étude utilisateur. Cette deuxième contribution a fait l’objet d’une publication internationale :

- L’Huillier, A., Castagnos, S., and Boyer, A. (2017). Are item attributes a good alternative to context elicitation in recommender systems? In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP ’17, pages 371–372, New York, NY, USA. ACM.

3.3 L’évolution temporelle de la diversité et les contextes implicites au service de la recommandation

La dernière contribution de cette thèse concerne l’exploitation des contributions précédentes afin de proposer diverses approches de recommandation, de générer des explications à destination des utilisateurs ou encore de s’assurer que l’utilisateur reste maître de ses choix.

Concernant les approches de recommandation, nous avons présenté 3 approches basées sur les contextes implicites. Ces approches peuvent être utilisées indépendamment les unes des autres, être combinées entre elles ou encore être combinées à d’autres approches de la littérature afin de former des systèmes hybrides. Deux des approches proposées exploitent les contextes implicites de la communauté d’utilisateurs d’un système afin de déterminer les items les plus adaptés au contexte de l’utilisateur à recommander (que nous appelons contexte courant). La différence majeure entre ces deux approches concerne le fait que l’une recherche les contextes implicites contenant les items du contexte courant alors que l’autre recherche les contextes implicites les plus similaires au contexte courant. De plus, cette seconde approche repose sur une phase de caractérisation permettant d’établir les profils des contextes implicites. La troisième approche de recommandation repose quant à elle sur l’hypothèse que les utilisateurs possèdent des comportements récurrents et qu’il est possible de prédire les futurs besoins des utilisateurs en utilisant des algorithmes d’apprentissage automatiques tels que les modèles de Markov. Les états de ces modèles peuvent alors être représentés par les items consultés ou encore par les contextes implicites. L’utilisation de l’évolution temporelle de la diversité permet de réordonner ou de limiter une liste de recommandations en vue de recommander les items adaptés au niveau de diversité.

Outre leur utilisation dans le but de calculer les recommandations, les contextes implicites permettent de fournir des explications aux utilisateurs. Ces explications peuvent, soit concerner la description du contexte courant de l'utilisateur afin de nous assurer que le contexte implicite fait sens du point de vue de l'utilisateur, soit, permettre d'expliquer les recommandations à l'utilisateur. De plus, nous avons montré en quoi l'utilisation de la diversité s'avère plus pertinente (dans certains) cas que la similarité pour présenter et expliquer les recommandations.

Les SR sont des outils dont le but est de restreindre de grandes quantités d'information afin de ne présenter aux utilisateurs que des items susceptibles de les intéresser. Cependant les SR peuvent malgré eux se sur-spécialiser dans les préférences connues des utilisateurs et enfermer petit à petit ces derniers dans un modèle de préférence dont l'utilisateur aura du mal à s'extraire [Abbassi et al., 2009]. Ce problème est notamment connu sous le nom de bulle de filtre (*filter bubble*). Afin de limiter l'apparition de ce problème, nous proposons d'exploiter notre modèle afin de comprendre les besoins réels des utilisateurs sans leur "imposer" les recommandations déterminées uniquement par rapport à leurs profils. Pour ce faire, certains items possédant un niveau de diversité spécifique peuvent être introduits dans les recommandations afin de contrôler et d'assurer notre compréhension du besoin réel des utilisateurs. En introduisant volontairement des items n'appartenant pas au contexte courant de l'utilisateur (déterminé à partir de l'évolution en diversité), nous permettons à l'utilisateur de sortir de son contexte courant ou alors de poursuivre son contexte s'il ne choisit pas ces items spécifiques. Ainsi, utiliser judicieusement la diversité dans les recommandations permet de s'assurer que ces dernières sont adaptées aux besoins réels des utilisateurs et qu'elles ne sont pas uniquement constituées d'items qui peuvent leur plaire en toutes circonstances. Autrement dit, nous cherchons à laisser le plus de contrôle possible à l'utilisateur, et ce, en ne limitant pas les recommandations au contexte implicite dans lequel se trouve l'utilisateur.

4 Présentation du plan de thèse

Le suite du manuscrit est organisée comme suit : dans le Chapitre 1 nous présenterons un état de l'art des systèmes de recommandation afin de définir le cadre de cette thèse ainsi que les enjeux qui en découlent. Par la suite, nous présenterons dans le Chapitre 2 les corpus de données que nous avons utilisés dans nos travaux. Dans le Chapitre 3, nous exposerons l'approche que nous avons développée afin de modéliser l'évolution temporelle de la diversité dans le parcours de consultations des utilisateurs. La présentation de la notion de contexte implicite, ainsi que la présentation et la formalisation du modèle DANCE feront l'objet du chapitre 4. L'étude utilisateur que nous avons menée afin d'étudier les liens qui existent entre le contexte explicite des utilisateurs et les attributs des items est présentée dans le Chapitre 5. Dans le chapitre 6 nous proposerons plusieurs approches de recommandation que nous avons développées sur la base de l'exploitation de nos modèles. Enfin, nous clôturerons ce manuscrit par un bilan du travail réalisé ainsi que par la présentation des perspectives offertes.

Chapitre 1

État de l'art

Sommaire

1.1	Les systèmes de recommandation	15
1.1.1	Formalisation du problème de la recommandation	15
1.1.2	Les principales familles d'algorithmes	16
1.1.3	Évaluation des recommandations	20
1.1.4	L'utilisateur : un humain derrière la machine	23
1.1.5	Discussion	26
1.2	Le contexte	26
1.2.1	Définition et intérêts	26
1.2.2	Intégration du contexte dans le processus de recommandation	29
1.2.3	Limites	35
1.2.4	Discussion	36
1.3	La diversité	37
1.3.1	Définitions et intérêts	37
1.3.2	Intégration de la diversité dans la recommandation	39
1.3.3	Le besoin en diversité	41
1.3.4	Diversité, nouveauté et serendipité	43
1.4	Bilan	45

Introduction

Dans l'introduction de ce manuscrit, nous avons montré que les SR sont devenus des outils indispensables afin d'assister les utilisateurs face à la quantité incommensurable de données à laquelle ils peuvent accéder. Nous avons également expliqué que bien que performants, ces outils se heurtent encore à de nombreux défis. Afin de comprendre les enjeux de cette thèse vis-à-vis des approches existantes, nous présentons dans ce chapitre l'état de l'art que nous avons élaboré. À l'issue de ce chapitre, les réponses aux questions suivantes auront été apportées :

- Quelles sont les différentes approches de recommandation ? Quels sont leurs avantages et inconvénients ?
- Quels sont les rôles du contexte et de la diversité dans le processus de recommandation ?
- De quelles manières les approches de recommandation actuelles intègrent-elles ces deux dimensions et quelles sont leurs limites ?
- Quelles sont les pistes de recherches dégagées afin de répondre aux problématiques abordées dans ce manuscrit ?

Après avoir énoncé la problématique générale à laquelle s'intéresse la recommandation, nous présenterons les deux principales familles d'algorithmes que sont le filtrage par contenu et le filtrage collaboratif. Nous montrerons que chaque famille d'algorithmes possède des forces et des faiblesses, et qu'il est possible de créer des approches hybrides permettant de bénéficier des avantages de ces deux familles. Nous aborderons ensuite un point crucial des SR qui est la phase d'évaluation. Ce point nous permettra d'expliquer les raisons pour lesquelles les approches de recommandation proposées ces dernières années tendent à se recentrer sur les besoins des utilisateurs plutôt que sur certaines métriques basées exclusivement sur la tâche de prédiction d'intérêt. Nous expliciterons alors par quels moyens les approches de recommandation ont dû évoluer afin de prendre en compte les facteurs humains dans leur processus de recommandation.

Nous nous intéresserons par la suite à deux facteurs humains en particulier, à savoir le **contexte** et le besoin en **diversité**. Nous définirons ces deux facteurs humains et nous nous intéresserons à la manière dont ils sont intégrés et exploités dans les SR de l'état de l'art. Notre intérêt particulier pour ces deux facteurs humains est motivé par deux raisons : premièrement, étant donné leur impact positif sur la qualité des recommandations, ces deux facteurs humains sont au cœur des recherches actuelles. Dans une époque où l'accès à Internet et donc à des services possédant des SR (musiques, *VOD*, e-commerce...) peut se faire de n'importe où (du moment que l'utilisateur possède une connexion internet), prendre en compte le contexte permet de s'assurer que les recommandations correspondent bien aux besoins (voire, aux contraintes) immédiat(e)s des utilisateurs. La diversité intervient quant à elle à de nombreux niveaux et permet entre autres de proposer des alternatives, couvrir l'intégralité d'un catalogue d'items, éviter la monotonie des recommandations... Cependant, bien que la prise en compte de ces deux facteurs soit indéniablement bénéfique (en ce qui concerne la qualité des recommandations), leur intégration dans le processus de recommandation reste complexe et soulève certains défis scientifiques. Ces défis sont d'autant plus importants en ce qui concerne l'intégration du contexte de l'utilisateur puisque ce dernier est déterminé en exploitant des données sensibles appartenant à la vie privée des utilisateurs. Or, la préservation et le respect de la vie privée des utilisateurs occupent à l'heure actuelle une place primordiale dans l'actualité politique mondiale au travers de la mise en place du RGPD¹⁹. Le RGPD est une réglementation européenne entrée en vigueur au mois de mai 2018 visant à fournir une transparence ainsi qu'un contrôle aux utilisateurs vis-à-vis des informations les concernant qui sont collectées par les sites internet, notamment au travers des cookies. Proposer des approches respectueuses de la vie privée des utilisateurs tout proposant de bonnes recommandations devient plus que jamais un sujet auquel la recherche se doit d'apporter des solutions. Concernant la diversité, malgré ses multiples utilités, son utilisation et son exploitation se sont principalement limitées à une métrique à maximiser en tout temps. Or, comme le montrent des travaux de la littérature [Lathia, 2010, Castagnos et al., 2013a], la diversité est une dimension qui évolue au cours du temps et dont la quantité à inclure dans les recommandations doit être maîtrisée tant quantitativement que temporellement. Deuxièmement, une lecture ap-

19. <https://www.eugdpr.org/>

profondie de l'état de l'art montre que ces deux dimensions sont la plupart du temps présentées et traitées indépendamment. Cependant, certains travaux de la littérature nous laissent penser qu'il existe un lien entre le contexte d'un utilisateur et la diversité de ces consultations. Ce lien sera discuté dans la Section 1.1.5.

1.1 Les systèmes de recommandation

1.1.1 Formalisation du problème de la recommandation

Comme nous l'avons présenté dans l'introduction de ce manuscrit, les SR ont pour but de restreindre la quantité d'information à présenter aux utilisateurs [Ricci et al., 2011]. Contrairement aux moteurs de recherche, leur but est de présenter aux utilisateurs des items susceptibles de les intéresser sans qu'ils aient besoin de solliciter le système afin de les obtenir. Avant d'en arriver au moment où les items sont présentés, un SR doit tout d'abord déterminer quels sont les items les plus pertinents à recommander, c'est-à-dire calculer les recommandations. Le calcul des recommandations comprend l'utilisation de méthodes très simples telles que le calcul des items les plus couramment consultés jusqu'à l'utilisation d'approches personnalisées exploitant de nombreux facteurs et tenant compte de nombreuses dimensions. Le problème de la recommandation peut être résumé par l'Équation 1.1 suivante [Adomavicius and Tuzhilin, 2005] :

$$\forall_u \in U, i'_u = \arg \max v(u, i) \text{ pour } i \in I \quad (1.1)$$

Si l'on considère l'ensemble des utilisateurs comme étant l'ensemble U et l'ensemble des items comme l'ensemble I , alors le but d'un SR est de maximiser la valeur retournée par la fonction d'utilité v qui lie un utilisateur avec un item. Autrement dit, un SR cherche à déterminer les ressources les plus utiles pour chaque utilisateur. Dans de nombreux cas, la fonction d'utilité est une fonction dont le but est de prédire une note qui représente la préférence prédite d'un utilisateur pour un item. Dans ce cas, il est possible de représenter v sous la forme de l'équation :

$$v : U \times I \rightarrow R \quad (1.2)$$

Dans l'Équation 1.2, l'ensemble des items I d'un système est lié à l'ensemble des utilisateurs U par un ensemble de valeurs appartenant à R (R étant le diminutif de *Rate*). Si cette fonction d'utilité est utilisée, les items recommandés par un SR seront alors les items possédant les préférences prédites les plus hautes.

Les préférences des utilisateurs sont la plupart du temps, représentées sous la forme d'une matrice à deux dimensions. Comme le montre la Figure 1.1 ci-dessous, la première dimension représente les items, la seconde les utilisateurs. L'intersection de ces deux dimensions représente les préférences des utilisateurs.

Grâce à la matrice a dans laquelle les préférences sont représentées sous forme de notes allant de 1 à 5, nous pouvons retrouver le profil utilisateur de chacun des utilisateurs. Pour cela, il suffit de récupérer la ligne correspondante à l'utilisateur pour lequel on souhaite obtenir les préférences. Par exemple l'utilisateur 5 (u_5) possède une faible préférence pour l'item 2 (i_2) et une forte préférence pour l'item 3 (i_3). Les préférences des utilisateurs peuvent être obtenues de deux manières, explicitement ou implicitement [Jawaheer et al., 2014]. Il est question de préférences explicites lorsque ces dernières sont obtenues en "interrogeant" directement l'utilisateur. Soit le système demande à l'utilisateur sa préférence (ex. : "Avez-vous apprécié ce film?"), soit l'utilisateur renseigne par lui-même sa préférence en attribuant une note ou un vote. Inversement,

	i_1	i_2	i_3	i_4
u_1		3	5	2
u_2	1	5	2	4
u_3		3		
u_4			5	2
u_5		1	4	

FIGURE 1.1 – Matrice utilisateur-item permettant de représenter les préférences des utilisateurs

une préférence implicite n'est pas directement fournie par l'utilisateur et doit être déterminée en analysant les traces d'usage des utilisateurs. Par exemple, une durée de consultation importante sur un item ou de multiples consultations de ce dernier permettent d'indiquer la préférence d'un utilisateur pour cet item. Ces deux types de préférences possèdent chacun leurs avantages et leurs inconvénients. Alors que les préférences explicites possèdent l'avantage de se passer d'interprétation, leur obtention nécessite cependant un effort cognitif de la part des utilisateurs. Au contraire, obtenir les préférences de manière implicite ne demande aucune implication de l'utilisateur, mais nécessite d'inférer les comportements des utilisateurs et peut donc constituer une source d'incertitude. Certains travaux suggèrent par ailleurs d'utiliser conjointement les deux types de préférences, ce qui permet d'améliorer les performances des SR contrairement au cas où l'un ou l'autre est utilisé seul [Gai and Qiang, 2016].

1.1.2 Les principales familles d'algorithmes

Historiquement, les algorithmes de recommandation sont classés en deux familles : les algorithmes de filtrage par contenu [Lops et al., 2011, Aggarwal, 2016a] et les algorithmes de filtrage collaboratif [Ekstrand et al., 2011, Adomavicius and Tuzhilin, 2005]. Alors que les premiers exploitent les caractéristiques des items, les seconds requièrent les données d'usage des utilisateurs sans connaissance des caractéristiques des items²⁰.

Filtrage par contenu

Le principe du filtrage par contenu repose sur le postulat suivant : si un utilisateur apprécie certaines ressources, alors les ressources qui sont similaires à ces dernières seront également appréciées par cet utilisateur. En 1992, lorsque cette approche fut présentée pour la première fois dans un système de notes (mémos) scientifiques, la similarité entre les ressources était calculée sur la base du nombre d'occurrences de mots scientifiques présents dans les items du système [Foltz and Dumais, 1992]. Ainsi, les mémos usant du même vocabulaire que les mémos appréciés par un utilisateur lui étaient recommandés.

La pierre angulaire de cette méthode de filtrage repose sur les informations, également appelées caractéristiques ou encore attributs, que le système possède sur les items. Dans la majorité des cas, les attributs correspondent aux caractéristiques techniques décrivant les items. Par exemple, dans le cadre d'un site de e-commerce de vêtements, les items sont des vêtements caractérisés par des attributs tels que leur prix, leur taille, leur couleur, leur matière. . . Les attributs peuvent également correspondre à des informations externes récupérées via d'autres services tels que Wiki-

20. Les caractéristiques des items peuvent être exploitées afin de réordonner une liste de recommandations par exemple, mais n'interviennent pas directement dans le calcul des recommandations.

pédia²¹ afin d'enrichir les informations connues sur les items [Gabrilovich and Markovitch, 2007]. Les attributs peuvent être de différents types tels que le type binaire, numérique ou encore textuel, et ces derniers vont conditionner la manière dont seront représentés les items. La forme la plus courante étant une représentation sous la forme d'un vecteur dans lequel les dimensions représentent les attributs. Lorsque les attributs sont numériques ou binaires, la représentation pose relativement peu de problèmes. Cependant, il est très fréquent que ces attributs soient de nature textuelle et requièrent alors une phase de traitement afin d'être exploitables. Ce traitement peut être plus ou moins complexe en fonction des techniques employées [Lops et al., 2011]. Une mesure très utilisée pour les attributs textuels est la mesure TF-IDF (*Term frequency, inverse document frequency*). Cette mesure permet d'obtenir la représentation d'une ressource sous forme vectorielle où chaque dimension représente un terme et la valeur de chaque dimension représente le poids de ce terme. La mesure de TF-IDF, n'est pas calculée indépendamment pour chaque document (ressource textuelle) mais est calculée en comparant la fréquence d'apparition d'un terme dans un document par rapport à la fréquence d'apparition de ce terme dans un ensemble de documents. Plus le poids d'un terme est important et plus ce terme est important dans ce document relativement aux autres documents de la collection à laquelle il appartient. La mesure TF-IDF peut être appliquée sur des items entièrement textuels ou sur des attributs tels que les mots-clés associés à des musiques dans des *playlists* [Jannach et al., 2015]. À l'ère du web 2.0 (également appelé web social), les commentaires, discussions, critiques à propos des ressources deviennent des attributs de plus en plus qualitatifs si nous sommes en mesure de les exploiter correctement. Même si de grands progrès ont été réalisés dans le domaine du traitement de la langue naturelle, l'exploitation de données textuelles comporte encore de nombreux défis [de Gemmis et al., 2015, Lin et al., 2018]. Par exemple, l'analyse des sentiments d'un commentaire est complexe et ne peut pas être effectuée uniquement sur les mots pris indépendamment. Un commentaire peut dégager un avis négatif alors que les termes utilisés sont positifs. Ainsi, un commentaire tel que "me tenir éveillé est le seul exploit de ce film" est très négatif malgré l'utilisation du terme "exploit" possédant une connotation positive.

De manière générale, un SR basé sur le contenu peut être divisé en 3 principaux modules définis par [Lops et al., 2011] qui sont : l'analyseur de contenu (*content analyzer*), le créateur de profil (*profile learner*) ainsi que le filtre (*filtering component*). Le but du premier module est d'analyser le contenu des items et d'effectuer certaines transformations nécessaires afin de représenter ces derniers sous une forme manipulable par le système (par exemple en utilisant une représentation vectorielle). Le second module vise à construire le profil de l'utilisateur dont le but est de représenter les préférences de l'utilisateur sur la base des attributs. Enfin, le troisième module vise à comparer le profil de l'utilisateur et les items du système afin de déterminer quels sont les items qui correspondent le mieux au profil que l'on cherche à recommander.

Les avantages du filtrage par contenu sont principalement sa transparence, sa capacité à expliquer les recommandations ainsi que sa capacité à ne pas souffrir du problème du démarrage à froid²² pour les nouveaux items [Lops et al., 2011]. Ainsi, il est possible d'expliquer aux utilisateurs comment les recommandations ont été obtenues (par exemple : "parce que vous avez aimé *Nocturne Op. 9, No. 2 in E b major, Andante* de Frédéric Chopin") ainsi que les raisons pour lesquelles elles sont pertinentes (par exemple : "nous vous proposons d'autres pièces pour piano"). En outre, l'arrivée d'un nouvel utilisateur qui ne possède pas de profil utilisateur peut être problématique dans les premières interactions étant donné que ses préférences ne sont pas

21. <https://fr.wikipedia.org/>

22. Le démarrage à froid (*cold-start*) représente le cas dans lequel le système ne possède pas suffisamment d'informations à propos d'un utilisateur ou d'un item.

connues. L'ajout d'un item ne cause en revanche pas de problème de démarrage à froid, car il est toujours envisageable d'en indexer le contenu. Compte tenu du fait que cette méthode repose sur l'exploitation des attributs des items, l'ajout de nouveaux items (sous réserve de connaître les attributs) ne nécessite pas de temps d'adaptation. Cependant, les approches basées sur le contenu sont particulièrement sensibles à la sur-spécialisation ainsi qu'à leur capacité limitée à pouvoir exploiter de manière optimale l'ensemble des attributs dont ils disposent. L'approche par contenu visant à maximiser la similarité entre le profil de l'utilisateur et les recommandations, l'utilisateur ne se verra pas proposer des recommandations ayant des caractéristiques différentes de celles qu'il apprécie. Par exemple, une approche par contenu ne proposera que des musiques de type rock à un utilisateur n'ayant écouté que du rock²³. Or, même si cela est cohérent avec les préférences des utilisateurs, ce dernier ne se verra pas recommander des musiques d'autres genres s'il n'initie pas par lui-même la consultation de ce genre d'items.

Filtrage collaboratif

Contrairement à l'approche par contenu qui n'exploite que les attributs des items afin de déterminer les items à recommander, le filtrage collaboratif exploite les préférences des autres utilisateurs afin de calculer les recommandations. Dès lors, il ne s'agit plus de positionner un utilisateur au sein d'un ensemble d'items, mais au sein d'une communauté d'utilisateurs (dans un espace à n dimensions, n étant le nombre d'items). L'approche collaborative fait l'hypothèse que les préférences des utilisateurs sont cohérentes entre elles²⁴ et qu'il est possible d'utiliser les préférences de la communauté pour déterminer les recommandations d'un utilisateur (ou d'un groupe d'utilisateurs). La première utilisation d'une méthode de recommandation impliquant la communauté d'utilisateurs date du début des années 90 [Goldberg et al., 1992] même si l'appellation de filtrage collaboratif n'est utilisée que deux années plus tard [Resnick et al., 1994]. Par la suite, les approches collaboratives ont évolué et se sont divisées en deux catégories : les approches basées sur la mémoire et les approches basées sur les modèles.

Les approches basées sur la mémoire utilisent l'ensemble des données du système afin de calculer les recommandations et sont divisées en deux catégories qui sont les approches orientées utilisateurs et les approches orientées items. À titre d'exemple, une des approches orientées utilisateurs consiste à déterminer pour chaque utilisateur les k utilisateurs qui lui sont le plus similaires, ces utilisateurs sont appelés les k plus proches voisins (kNN pour k Nearest Neighbor). Le calcul de similarité est basé sur les vecteurs de notes attribuées par les utilisateurs aux items et peut être réalisé selon différentes mesures dont les plus connues sont le coefficient de corrélation de *Pearson* et la similarité Cosinus [Ticha, 2015]. Une fois que les k plus proches voisins d'un utilisateur sont définis, les notes de ces derniers sont exploitées afin de prédire les préférences non connues de l'utilisateur. Les prédictions les plus hautes, également appelées top- N seront alors des items à recommander. Selon ce même principe de calcul des plus proches voisins, les approches collaboratives orientées items proposées par [Linden et al., 2003, Sarwar et al., 2001] constituent la deuxième catégorie d'approches basées sur la mémoire. Dans [Sarwar et al., 2001], les kNN sont calculés à l'échelle des items et non plus des utilisateurs. Connaissant les items les plus similaires (kNN) calculés en fonction des notes affectées par tous les utilisateurs confondus, il est possible de choisir parmi ces derniers ceux à recommander. Même si ces méthodes s'avèrent efficaces pour déterminer les recommandations, elles souffrent de problèmes de passage à l'échelle étant donné que les calculs de similarité (entre utilisateurs ou items) doivent être effectués à chaque nouveau

23. Un système utilisant des calculs de distances sémantiques pourra cependant proposer des genres similaires [Diefenbach et al., 2016].

24. La notion de cohérence fait dans ce cas davantage référence au cas inter-utilisateur qu'au cas intra utilisateur.

cycle de recommandations. Or, les systèmes actuels sont amenés à gérer de grandes quantités d'utilisateurs et d'items et ces approches peuvent devenir limitantes à une époque où la réactivité est essentielle.

Les approches basées sur un modèle visent quant à elles à réduire la quantité d'informations possédée par le système afin de créer un modèle de représentation des données [Breese et al., 2013]. L'avantage par rapport aux approches basées sur la mémoire est qu'il n'est plus nécessaire d'effectuer des calculs sur l'ensemble des données pour déterminer les recommandations d'un utilisateur, le modèle peut être directement exploité et interrogé ce qui limite considérablement les temps de calcul. Parmi les différentes approches exploitant des modèles, nous pouvons notamment citer les approches par factorisation de matrices [Koren et al., 2009, Aleksandrova et al., 2017] ainsi que les approches de clustering [Wei et al., 2012]. Le principe de la factorisation de matrices repose sur l'extraction de facteurs latents de la matrice utilisateur-item ce qui permet de réduire le nombre de dimensions nécessaires à la représentation des données. Afin d'extraire ces facteurs latents, des méthodes telles que l'analyse en composante principale, la décomposition en valeur singulière ou encore l'analyse latente sémantique sont utilisées [Hofmann, 2004]. Les approches de clustering cherchent pour leur part à isoler des groupes d'utilisateurs ou des groupes d'items partageant des caractéristiques communes. De ce fait, sachant qu'un utilisateur ou un item appartient à un *cluster* donné, calculer ses k plus proches voisins n'est plus nécessaire et il suffit d'exploiter les entités qui se trouvent dans ce même *cluster* [Sarwar et al., 2002, Esslimani et al., 2009, Juniarta et al., 2018].

L'un des principaux avantages du filtrage collaboratif réside dans sa capacité à proposer à un utilisateur des ressources ayant des caractéristiques différentes de son profil utilisateur. En effet, compte tenu du fait que ce ne sont plus les caractéristiques des items qui sont exploitées, mais les préférences de la communauté d'utilisateurs, un utilisateur pourra très bien se voir proposer des films de science-fiction alors qu'il n'a jusque-là exprimé aucun avis positif ou négatif sur ce genre de film. Cette recommandation lui est faite, car des utilisateurs qui possèdent des profils similaires au sien ont également apprécié des films de science-fiction ou encore parce que les films appréciés par cet utilisateur sont proches (en termes de distance dans l'espace de représentation des items et des préférences qui leur sont associées) de ces films de science-fiction qui lui sont recommandés. Cependant, comme toute approche, cette dernière souffre de certaines limitations telles que le problème du démarrage à froid [van den Oord et al., 2013]. Dans le cas d'un nouvel utilisateur, bien que le système puisse tout de même proposer des recommandations si le nombre de notes connues de l'utilisateur est faible, le risque de proposer des recommandations non pertinentes est important. Ce même phénomène se produit également lors qu'un nouvel item est introduit dans le système. De plus, le faible nombre de préférences connues étant donné le nombre important d'utilisateurs et d'items [Grčar et al., 2006], ainsi que la difficulté à pouvoir fournir des explications par rapport aux recommandations proposées [Herlocker et al., 2000] constituent d'autres limites de cette approche.

Les avantages sans les inconvénients : l'hybridation

Les deux familles de recommandation que nous venons de voir possèdent chacune leurs avantages et inconvénients. Cependant, il est possible de combiner ces deux approches afin de créer des systèmes hybrides [Burke, 2002, Ticha, 2015] dans le but de bénéficier des avantages des deux approches tout en diminuant les inconvénients. En effet, comme nous l'avons vu, le filtrage collaboratif est très sensible au démarrage à froid, alors que le filtrage par contenu est beaucoup moins impacté par ce phénomène. Proposer un modèle qui utilise tout d'abord une approche par contenu afin d'obtenir les préférences de l'utilisateur puis une approche collaborative une fois

que les informations obtenues sont suffisantes s'avère être un bon moyen d'éviter le problème du démarrage à froid et la sclérose des recommandations. L'hybridation ne se limite pas à ce cas d'usage et il existe de nombreuses combinaisons. Par exemple, 6 modèles d'hybridation ont été proposés par [Burke, 2002]. La Figure 1.2 ci-dessous illustre deux de ces modèles hybrides : le modèle par permutation et le modèle par mixage. Les approches utilisées pour présenter l'hybridation sur cette figure sont des approches collaboratives et par contenu, mais il est également possible d'utiliser des approches de recommandation démographiques [Nguyen et al., 2006], basées sur l'utilité ou encore sur les connaissances [Tarus et al., 2017].

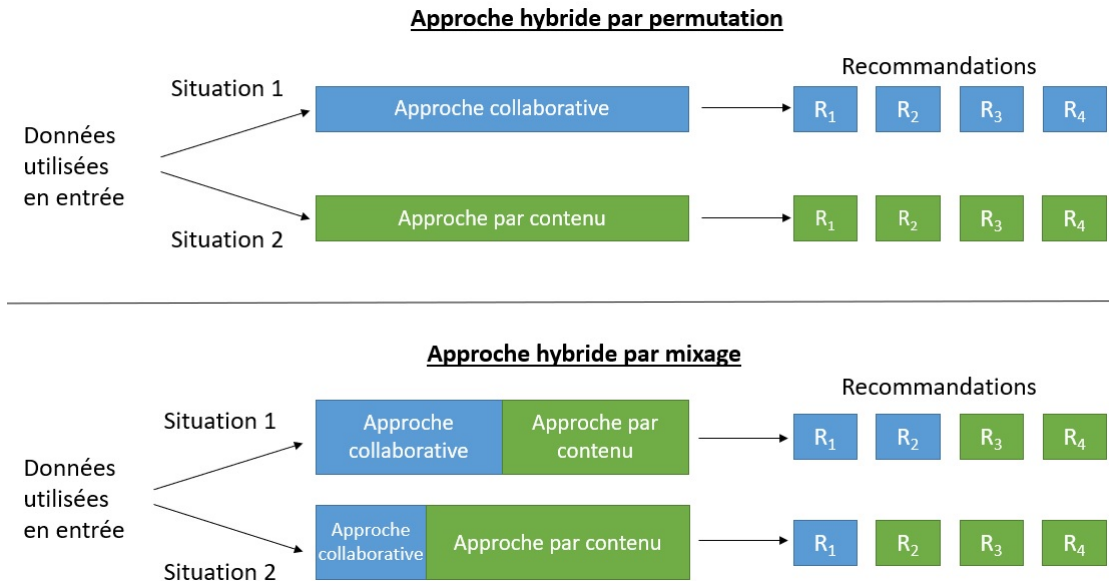


FIGURE 1.2 – Exemple de modèles de recommandation hybrides [Burke, 2002]

Dans le cas de l'approche par mixage, les différentes approches de recommandation impliquées dans le modèle hybride sont utilisées conjointement à chaque cycle de recommandation, mais avec des poids différents en fonction de la situation. Par exemple, une liste de recommandations peut être composée de 80 % d'items provenant d'une approche par contenu et 20 % d'items provenant d'une approche collaborative. Dans le cas de l'approche par permutation, les approches sont utilisées indépendamment les unes des autres en fonction des situations. Du fait du nombre de modèles d'hybridation et des diverses combinaisons d'approches de recommandation envisageables, l'hybridation permet de créer des systèmes adaptés à de nombreux cas d'usage en exploitant au mieux les forces des approches utilisées.

1.1.3 Évaluation des recommandations

Afin de connaître les performances d'un SR, il est nécessaire d'évaluer les recommandations produites. Pour savoir si un SR est plus performant que les SR auxquels il est comparé, ce dernier peut être évalué sur une ou plusieurs dimensions que nous détaillerons par la suite. Indépendamment de ces dimensions, il existe deux situations dans lesquelles le SR peut être évalué : en situation d'usage ou hors ligne. Lors d'une évaluation en situation d'usage, le SR est implanté sur une plateforme ouverte à des utilisateurs suivant un protocole expérimental. Des méthodes de A/B testing sont par exemple utilisées afin de comparer les résultats obtenus avec les différentes approches utilisées (l'approche testée ainsi que les algorithmes de l'état de l'art auxquels

l'approche testée est comparée). De plus, il est également possible d'interroger directement les utilisateurs sur leur satisfaction et/ou leur perception. Lors d'une évaluation hors ligne, les utilisateurs ne sont pas impliqués et des jeux de données issus de différentes applications réelles sont exploités. La satisfaction des utilisateurs ne pouvant pas être mesurée, ces approches nécessitent de recourir à des métriques permettant d'évaluer les recommandations produites. Chacune de ces deux situations possède des forces et des faiblesses. Les tests en situation d'usage permettent davantage d'éprouver la qualité des recommandations et d'interagir avec les utilisateurs. Cependant, ces tests sont longs à déployer et nécessitent d'obtenir des utilisateurs investis. Les situations hors ligne offrent l'avantage de pouvoir être testées indéfiniment, mais également de pouvoir tester de nombreuses hypothèses dans des conditions toujours identiques, ce qui constitue un atout pour la communauté scientifique. Cependant, les corpus hors ligne contiennent des données figées dans le temps ne permettant pas d'obtenir de retours des utilisateurs. Ainsi, il est possible de conclure qu'une recommandation est pertinente si elle a été consultée par l'utilisateur dans le corpus de données étudié. En revanche, si cette ressource (celle recommandée) n'a pas été consultée, nous ne pouvons pas conclure qu'il s'agit d'une recommandation non pertinente pour autant. Pour leurs avantages et leur facilité d'usage, les protocoles hors ligne sont devenus le cadre de test le plus répandu.

Pendant de nombreuses années, les SR ont principalement été évalués uniquement sur la base de leur capacité à pouvoir prédire les préférences des utilisateurs au travers d'une métrique que l'on appelle la précision. La précision correspond à la distance existante entre la préférence (exprimée sous forme de note) prédite par le système et la note réellement attribuée par l'utilisateur. Les métriques les plus connues et utilisées sont la MAE (*Mean Absolute Error*, erreur moyenne absolue) et la RMSE (*Root-Mean-Square Error*, erreur quadratique moyenne). Pour effectuer ces prédictions, un corps de données est divisé en 2 parties : une base d'apprentissage et une base de test. La base d'apprentissage sera utilisée pour apprendre les paramètres nécessaires aux calculs des prédictions (calcul des k plus proches voisins, initialisation des profils utilisateurs...). Puis, ces prédictions sont ensuite évaluées à l'aide de la base de test dans laquelle les préférences réelles des utilisateurs ont été volontairement supprimées.

$$MAE = \sum_{i=1}^n \times \frac{|nr_i - np_i|}{N} \quad (1.3)$$

$$RMSE = \sqrt{\sum_{i=1}^n \times \frac{(nr_i - np_i)^2}{N}} \quad (1.4)$$

Les équations 1.3 et 1.4 permettent de connaître l'erreur moyenne commise par un SR sur l'ensemble des prédictions effectuées N , où np_i représente la note prédite par le SR sur l'item i et nr_i la note réelle (celle de l'utilisateur). Un SR est plus précis qu'un autre dès lors que son erreur moyenne est plus faible. Le fait d'élever l'erreur au carré (Équation 1.4) permet de donner plus d'importance aux erreurs conséquentes et de minimiser l'impact des faibles erreurs.

Selon les domaines applicatifs ou les jeux de données testés, il n'est pas toujours possible de se baser sur la prédiction des préférences. Dans certains cas, le système peut avoir affaire à des items dont il connaît déjà la note donnée par l'utilisateur. Ce cas peut par exemple être illustré par les *playlists* de musiques qui sont devenues des outils de plus en plus exploités par les utilisateurs et pour lesquelles les SR sont de plus en plus utilisés [Bonnin and Jannach, 2014]. Il ne s'agit alors plus de prédire une note, mais de prédire quelles sont les musiques qui peuvent intéresser l'utilisateur étant donnée la *playlist* qu'il est en train de construire. La recommandation de la musique suivante (mode radio) constitue un autre exemple dans lequel la recommandation ne

se base plus sur une prédiction de préférences puisqu'il est possible de recommander à un utilisateur sa chanson favorite et ce plusieurs fois dans une même semaine. Dans ces exemples, il ne s'agit plus d'estimer une préférence, mais d'estimer les besoins des utilisateurs afin de proposer les recommandations les plus adéquates. Autrement dit, contrairement à certains SR qui cherchent à estimer les notes, ces systèmes cherchent à construire des listes d'items potentiellement pertinents. Dans ce cas, étant donné que les préférences (les notes) sont connues, des métriques telles que la MAE ou la RMSE ne sont pas adaptées. L'évaluation des SR se fera alors en générant à chaque nouvelle consultation de l'utilisateur une liste de recommandations et en comparant les items effectivement consultés par l'utilisateur à la liste de recommandations. Afin de mesurer les performances, il existe différentes métriques telles que la précision et le rappel [Cleverdon and Kean, 1968] mais également d'autres formes dérivées telles que :

- précision : correspond au rapport entre le nombre de fois où un des items de la liste de recommandations a effectivement été consulté/choisi par l'utilisateur et le nombre de fois où le système a proposé des recommandations ;
- rappel : correspond au rapport entre le nombre de fois où un des items de la liste de recommandations a effectivement été consulté/choisi par l'utilisateur et le nombre de fois où l'utilisateur a consulté un item ;
- F-mesure : la F mesure est une métrique qui permet de combiner à la fois les scores de précision et de rappel. Elle se calcule comme suit :

$$F - mesure = 2 \times \frac{précision \times rappel}{précision + rappel} \quad (1.5)$$

- le hit-ratio²⁵ : correspond au rapport entre le nombre de fois où un des items de la liste de recommandations a effectivement été consulté/choisi par l'utilisateur et le nombre de fois où le système a proposé des recommandations ;
- le MRR (*Mean Reciprocal Rank*) : correspond à la moyenne de la position dans la liste de recommandations d'un item recommandé qui a effectivement été consulté/choisi.

Le but d'un SR consiste alors à maximiser le hit-ratio tout en minimisant la valeur du MMR. En effet, ces deux métriques doivent toujours être comparées ensemble. Dans le cas (peu probable) où une liste de recommandations contient l'ensemble des items du système, le hit-ratio sera égal à 1 (le choix futur de l'utilisateur sera forcément dans la liste de recommandations). Cependant, la valeur de la MRR correspondante serait anormalement faible. Ces deux métriques peuvent être utilisées en situation d'usage ou bien hors ligne. Cependant, les performances ne pourront pas être comparées entre les deux situations étant donné que dans le cas hors ligne, la liste de recommandations peut comporter plusieurs dizaines voire centaines d'items, ce qui n'est pas le cas en situation réelle. De plus, en situation réelle, la liste de recommandations est proposée à l'utilisateur qui peut sélectionner un item alors qu'en situation hors ligne, nous rappelons que les actions des utilisateurs sont immuables. Dès lors, même si une liste de recommandations proposée par une approche est plus pertinente que les items réellement consultés par l'utilisateur, cette approche ne pourra être évaluée ni positivement ni négativement. Nous ne pouvons dans ce cas, rien conclure quand à la pertinence des recommandations.

25. La définition du hit-ratio est identique à celle de la précision. La précision et le rappel étaient à l'origine utilisés dans le domaine de la recherche d'informations et ces définitions ont été adaptées au domaine de la recommandation. Malgré cette adaptation, le terme de hit-ratio est de plus en plus utilisé.

Alors que ces métriques sont largement utilisées dans la communauté, de nombreuses études telles que [McNee et al., 2006, Jannach and Adomavicius, 2016] mettent en lumière leurs limites en tant qu'indicateurs de la qualité d'un SR. Les auteurs soutiennent que les recommandations s'adressent à des utilisateurs qui ont des objectifs particuliers qui dépendent de la situation dans laquelle ils se trouvent et que la précision seule n'est pas suffisante pour juger de la qualité d'un SR. Ainsi, l'évaluation d'un SR ne peut plus être résumée à une seule métrique, étant donné que le processus d'évaluation dépend de la tâche qu'effectuent les utilisateurs. Comme nous l'avons évoqué précédemment avec le domaine de la musique, la tâche de recommandation de la musique suivante n'est pas la même tâche que la recommandation de musiques pour la création de *playlists* et le processus d'évaluation ne peut être identique. De plus, il a été démontré qu'il est possible d'augmenter la précision (donc de minimiser l'erreur commise) d'un SR en recommandant exclusivement des items populaires [Jannach et al., 2013]. Nous allons nous intéresser à présent aux raisons pour lesquelles certaines métriques encore très largement répandues actuellement peuvent s'avérer inefficaces pour mesurer la qualité d'un SR. Nous verrons notamment que cela est dû au fait que les recommandations sont soumises aux plus complexes des processus qui soient : le raisonnement et la prise de décision humaine.

1.1.4 L'utilisateur : un humain derrière la machine

Les processus d'évaluation des SR sont au cœur des discussions actuelles [Ricci et al., 2015] [Jannach et al., 2017, Bellogín et al., 2017]. Ces dernières sont majoritairement orientées autour du principal acteur des SR : l'utilisateur. Toutefois, les premiers constats relatifs à la prise en compte d'autres dimensions que la précision datent d'environ une décennie [McNee et al., 2006]. Dans leur papier, les auteurs expliquent que la précision n'est pas une métrique autosuffisante et que d'autres facteurs sont à prendre en compte dans le processus d'évaluation. Les facteurs dont il est question concernent l'utilisateur et notamment sa capacité à accepter les recommandations. Par la suite et dans le même ordre d'idée, d'autres travaux se sont focalisés sur l'importance des facteurs humains dans le processus de recommandation. Par exemple, Jones en 2010 [Jones, 2010] propose une adaptation du TAM (Technology Acceptance Model) dans le domaine de la recommandation. Le modèle TAM [Davis, 1989], est un modèle mis au point à l'origine afin de comprendre la manière dont des utilisateurs perçoivent un système informatique. La révision proposée par [Jones, 2010], illustrée par la Figure 1.3, permet d'illustrer les deux dimensions expliquant la manière dont les utilisateurs acceptent le système.

Nous pouvons constater que de nombreux facteurs interviennent sur les deux axes de ce modèle qui peuvent être résumés à travers deux questions :

- Est-ce que le système est facile d'utilisation ? (Facilité d'utilisation)
- Est-ce que le système est utile pour l'utilisateur ? (Utilité)

Dès lors, il apparaît évident qu'un SR ne peut être évalué sur sa seule capacité à prédire les préférences d'un utilisateur aussi précises soient ces prédictions. En effet, comme nous pouvons le constater sur la Figure 1.3, la précision ne constitue qu'un seul des facteurs intervenant dans l'utilité d'un SR qui, elle-même intervient conjointement avec la facilité d'utilisation dans l'acceptation du système par l'utilisateur. Ainsi, un SR qui serait très précis, mais qui ne tiendrait pas compte du contexte de l'utilisateur pourrait ne pas être accepté par l'utilisateur. Par exemple si un SR de musique n'est pas en mesure de proposer les bonnes musiques au bon moment (par exemple, en proposant une musique classique à un utilisateur qui souhaite mettre de l'ambiance dans une fête), ce dernier pourra être jugé inutile et ne sera donc pas être accepté. Suite à ce constat, certaines études font également état des dimensions qu'un SR doit prendre en compte

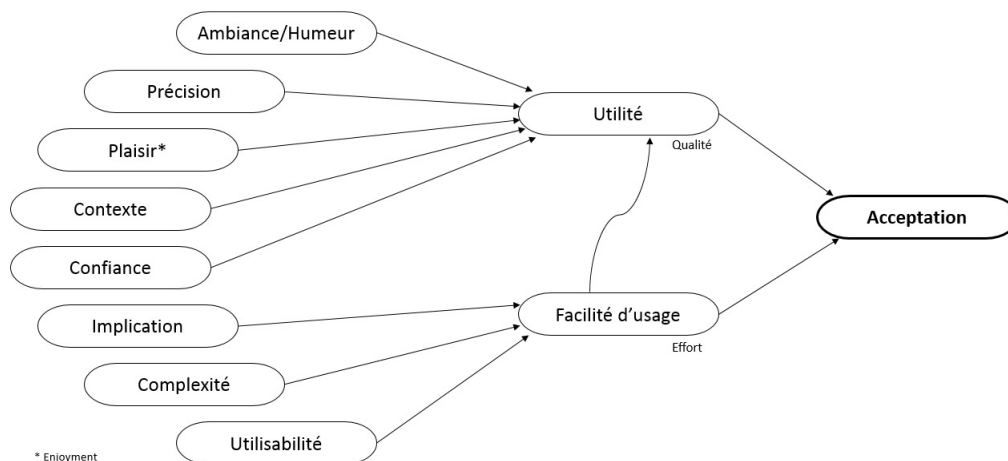


FIGURE 1.3 – Adaptation du modèle *TAM* par [Jones, 2010]

afin de proposer des recommandations de qualité [Shani and Gunawardana, 2011]. Parmi ces dimensions, nous pouvons notamment citer :

- **Couverture** : la métrique de couverture est utilisée afin de savoir si l'ensemble des items d'un catalogue peut être recommandé. Par exemple, il existe un biais bien connu par la communauté tendant à amener les algorithmes à recommander en majorité les ressources les plus populaires au détriment des ressources moins populaires [Celma, 2008]. La couverture d'un SR peut par ailleurs être mesurée à l'aide métrique de diversité agrégée [Adomavicius and Kwon, 2012] (*cf. infra* 39).
- **Confiance système** (en anglais *Confidence*) : lors du calcul des recommandations, le SR détermine les recommandations sur la base des informations dont il dispose sur l'utilisateur courant (son profil utilisateur) et sur la communauté d'utilisateurs ou les items. En fonction de ces informations, un système peut être plus ou moins sûr de la qualité des recommandations générées. Par exemple, si l'utilisateur courant possède un profil utilisateur qui n'est pas cohérent par rapport à ceux des autres utilisateurs (problème des *Grey-Sheep users* [Ghazanfar and Prügel-Bennett, 2014, Gras et al., 2016]) ou que certaines informations déterminantes dans le calcul des recommandations ne sont pas connues, alors la confiance du SR vis-à-vis de ces recommandations peut être faible. Il est alors possible d'utiliser des indicateurs de confiance représentant la probabilité que les recommandations intéressent/soient utiles pour l'utilisateur.
- **Confiance utilisateur** (en anglais *Trust*) : représente la confiance qu'attribue un utilisateur envers les recommandations qui lui sont proposées. Si les recommandations sont incohérentes ou que l'utilisateur ne parvient pas à comprendre pourquoi ces recommandations lui sont proposées, le système peut alors perdre la confiance de l'utilisateur qui dans certains cas abandonnera le système. Dans ce cas, il est question de confiance envers le système et non de confiance entre les utilisateurs qui constituent également un champ de recherche dans le domaine de la recommandation [Haydar et al., 2015].
- **Nouveauté** : un SR doit être en mesure de s'adapter à de nouveaux utilisateurs (sur lesquels il ne possède aucune information) comme à de nouveaux items. Nous avons vu précédemment que cela pouvait poser certains problèmes de démarrage à froid en ce qui concerne les approches collaboratives.
- **Sérendipité** : recommandation due au hasard utile pour l'utilisateur courant alors que

ce dernier ne recherchait pas spécifiquement cet item en particulier. La sérendipité d'une recommandation peut être vue comme sa capacité à surprendre positivement l'utilisateur. Cela se produit notamment avec l'utilisation d'approches collaboratives [Herlocker et al., 2004] étant donné que les préférences d'autres utilisateurs (les voisins) sont utilisées afin de calculer les recommandations et que le contenu n'est alors pas limité au profil de l'utilisateur. Les approches basées sur le contenu peuvent également être adaptées afin de maximiser cette dimension [Jaquinta et al., 2008].

- **Diversité** : ne proposer que des ressources similaires en termes d'attributs apportera certes une forte précision. Cependant, cela pourra avoir comme conséquence d'enfermer l'utilisateur dans ses préférences et de le priver de découvrir d'autres items. La diversité comporte également de nombreux avantages que nous présenterons dans la Section 1.3.
- **Utilité** : proposer des recommandations pertinentes à un moment donné est différent de proposer des items qui peuvent plaire à l'utilisateur dans l'absolu. Par exemple, proposer un film d'horreur à un utilisateur qui cherche un film à regarder avec ses enfants est loin d'être utile (bien que ce film puisse lui plaire dans l'absolu). La question de l'utilité est très liée au contexte que nous développerons dans la Section 1.2.
- **Robustesse** : capacité à fournir des recommandations de qualité malgré la présence de données frauduleuses, inexactes ou incomplètes.
- **Respect de la vie privée** : bien que la satisfaction de l'utilisateur soit la finalité des SR, cela ne justifie en rien l'utilisation de données personnelles lui appartenant. Il est donc important pour un SR d'identifier les informations suffisantes et nécessaires à la recommandation en respectant au maximum la vie privée de l'utilisateur et en veillant à toujours obtenir son consentement.
- **Capacité d'adaptation** : les utilisateurs ne sont pas des entités statiques dont les préférences sont stables au cours du temps. Il est possible que ce qui plaît à un utilisateur aujourd'hui ne lui plaise plus le lendemain ou encore, ce qui plaît à un utilisateur dans une situation donnée n'est pas nécessairement ce qui lui plaira dans une autre situation.
- **Passage à l'échelle** : compte tenu du nombre important d'utilisateurs et d'items que peuvent avoir à gérer les SR à l'heure actuelle, il est nécessaire de mettre au point des approches pouvant être déployées en situation réelle d'usage. Les solutions proposées doivent être en mesure de traiter de grands volumes de données dans des délais très courts car les recommandations sont utiles la plupart du temps uniquement dans une situation donnée.

Certaines métriques telles que la couverture et le passage à l'échelle ne concernent pas directement l'utilisateur. Cependant, ne pas les prendre en compte peut avoir un impact négatif sur l'acceptation d'un SR par l'utilisateur à long terme ou dans certaines circonstances. Inversement, certaines dimensions telles que la diversité, l'utilité, la nouveauté et le contexte sont des dimensions qui impactent directement l'utilité des recommandations et dont la prise en compte doit constituer la priorité des SR.

Afin de prendre en compte ces nombreuses dimensions, certaines approches dites approches multi-objectifs ou encore multicritères ont été proposées [Adomavicius and Tuzhilin, 2005] [Rodriguez et al., 2012, Lacerda, 2017]. Le principe de ces approches est de proposer des méthodes de décision permettant de trouver les items ou les ensembles d'items qui répondent à plusieurs critères tels que ceux que nous avons présentés. Par exemple, un SR multicritères peut avoir pour objectif de maximiser l'utilité des recommandations en cherchant à la fois à proposer le plus d'items nouveaux tout en diversifiant les items de la liste de recommandations proposée. Dans ce cas, le SR cherche alors à optimiser 3 critères qui sont : l'utilité, la diversité et la

nouveauté. Sachant qu'il n'est pas possible de maximiser certaines dimensions sans affecter la prise en compte d'autres dimensions, ces approches cherchent alors à déterminer les meilleurs compromis possible. Parmi les techniques permettant d'y parvenir, nous pouvons notamment mentionner [Ribeiro et al., 2014, Adomavicius et al., 2011a] :

- trouver les solutions optimales selon un front de Pareto ;
- déterminer une combinaison linéaire à partir des critères et réduire le problème à l'optimisation d'un seul critère ;
- optimiser le critère le plus important et utiliser les autres critères comme des contraintes ;
- optimiser un critère, se servir de cette optimisation comme contrainte puis répéter cette optimisation pour les autres critères.

Le front ou encore optimum de Pareto constitue l'ensemble des états pour lesquels il n'est pas possible d'améliorer un critère sans dégrader les autres. Deux solutions du front de Pareto posséderont des critères différents, mais si une solution maximise davantage un des critères pris en compte par rapport à une autre solution, cette deuxième solution sera nécessairement obtenue au détriment d'au moins un autre critère. De plus, il n'est pas possible de savoir si une solution du front de Pareto est meilleure qu'une autre étant donné que chaque "avantage" donné à un critère se traduit par la pénalisation d'un autre.

1.1.5 Discussion

Les approches basées sur le contenu et les approches collaboratives sont les deux principales familles d'algorithmes de recommandation. Ces dernières possèdent chacune leurs forces et leurs faiblesses en ce qui concerne la prédiction des préférences des utilisateurs. De plus, ces deux approches peuvent être combinées afin de former des systèmes hybrides permettant de profiter des avantages de chacune. Or, malgré une amélioration constante de leur capacité à prédire les préférences des utilisateurs, les SR connaissent actuellement une remise en question majeure quant à leur aptitude à répondre aux réels besoins des utilisateurs. Dans [Jannach and Adomavicius, 2016, Aggarwal, 2016b], les auteurs proposent des standards permettant de mesurer de la manière la plus objective qui soit les performances des SR afin de s'assurer que les recommandations soient les plus pertinentes possible. En effet, les recommandations ne peuvent être résumées à une prédiction de préférences et d'autres dimensions telles que les facteurs humains doivent être considérées. Afin de prendre en considération ces nombreuses dimensions, certaines approches s'intéressent à la mise au point de modèles multi-objectifs dont le but est de prendre en compte plusieurs dimensions dans le calcul des recommandations [Adomavicius et al., 2011a]. Ces approches sont alors plus à même de couvrir les besoins des utilisateurs que les approches ne se basant que sur un seul critère à maximiser tel que la précision. Cependant, malgré les bénéfices apportés par la prise en compte de plusieurs dimensions, nous pensons que certaines de ces dimensions doivent être réévaluées, notamment en ce qui concerne la manière dont elles sont prises en compte.

Dans nos travaux, nous nous focalisons sur le besoin en diversité ainsi que sur le contexte qui sont deux facteurs humains dont l'intégration dans les SR est primordiale afin de garantir la qualité des recommandations. Nous soutenons que le niveau de diversité à proposer dans les recommandations ne peut pas être résumé à un score à maximiser et que la prise en compte du contexte de l'utilisateur ne doit pas s'effectuer au détriment du respect de sa vie privée. Dans les sections suivantes, nous présentons ces deux facteurs humains ainsi que les arguments qui nous ont amenés à cette prise de position.

1.2 Le contexte

Nous avons vu précédemment que les SR ne pouvaient pas se contenter d'estimer les préférences d'un utilisateur s'il souhaite l'assister efficacement. Par exemple, ce n'est pas parce qu'un utilisateur affiche des préférences fortes pour certaines musiques électroniques qu'il est pertinent de lui proposer des musiques de ce genre dans chaque liste de recommandations. S'il n'apprécie ce genre de musique que lorsqu'il se trouve à son travail, lui recommander ce genre de musique le soir n'aura pas d'intérêt puisqu'il préfère écouter du jazz lorsqu'il est chez lui. Cet exemple illustre plusieurs dimensions présentées par [Shani and Gunawardana, 2011] qui sont l'utilité et la capacité d'adaptation. Ces deux dimensions peuvent être regroupées dans une unique dimension : le contexte.

1.2.1 Définition et intérêts

Le contexte est une notion intelligible que nous utilisons quotidiennement : “dans quel contexte cet événement s'est-il produit?”, “Il faut situer cette phrase dans son contexte” ou encore “Pris dans ce contexte, je suis d'accord avec toi”. Pourtant, cette notion est difficile à définir, car elle est spécifique au domaine applicatif. Certains auteurs ayant analysé plus de 150 définitions du contexte ont par ailleurs déclaré "Il est difficile de trouver une définition pertinente et satisfaisante pour chaque discipline" [Bazire and Brézillon, 2005]. Selon ces auteurs, la définition du contexte dépend de nombreux facteurs qui sont eux-mêmes dépendants de l'utilisation que l'on souhaite faire du contexte. De nombreuses questions se posent alors lorsque l'on souhaite apporter une définition du terme “contexte” : quel contexte est pertinent pour mon cas? S'agit-il du contexte de la personne, du contexte de la tâche, du contexte des interactions ou encore du contexte de la situation? Quand est-ce qu'un contexte commence et quand est-ce qu'il s'arrête?... Pour ces raisons, nous comprenons qu'apporter une seule définition du contexte dans le domaine des SR s'avère complexe car il peut être perçu selon différents points de vue et les SR peuvent intervenir dans de nombreuses situations.

Cependant, nous pouvons nous baser sur une définition plus générale afin d'en dégager les principales composantes. Pour ce faire, nous proposons de nous reposer sur la définition du contexte donnée par le TLFi²⁶.

Contexte : ensemble de circonstances liées, situation où un phénomène apparaît, un événement se produit.

Au travers de cette définition, nous pouvons tout d'abord extraire le fait que le contexte se compose d'un ensemble de circonstances. Même si une personne ne se trouve que dans un seul et même contexte, l'ensemble de circonstances utilisées pourra donner lieu à des définitions de contextes différents. Ainsi, le contexte d'un utilisateur peut être défini par sa position géographique uniquement, sa position géographique et la météo, sa position géographique et la météo ainsi qu'une infinité d'autres circonstances. Un autre point important dans la notion de contexte souligné par la définition est le fait que les circonstances sont liées. C'est d'ailleurs cette liaison entre les différentes circonstances qui rend le contexte si complexe à définir, d'autant plus que ces circonstances peuvent dans certains cas être dépendantes²⁷ les unes des autres comme nous le verrons par la suite. Enfin, cette définition nous apprend également que ces circonstances vont

26. Trésor de la Langue Française informatisé <http://atilf.atilf.fr/>

27. La terme de liaison fait référence au fait qu'un contexte est défini par la liaison de plusieurs circonstances (en famille & en vacances & le soir). La notion de dépendance fait référence au fait que certaines circonstances sont dépendantes entre elles (neigeux & température basse).

servir de cadre aux événements. Dès lors, lorsque nous nous intéressons au contexte, nous ne pouvons plus simplement affirmer "cet événement s'est produit" mais nous sommes obligés de préciser "dans le contexte ... cet événement s'est produit". Étant donné que nous nous intéressons au contexte dans le domaine des SR, une définition appropriée à ce domaine est celle donnée par [Castagnos, 2008] qui se rapproche de ce que certains auteurs définissaient avant même l'apparition des SR comme étant des actions situées [Suchman, 1987] :

Contexte dans un SR : ensemble des circonstances d'utilisation et des attributs de l'utilisateur pouvant influencer sur son comportement.

En plus de couvrir la définition précédente, cette définition rajoute la notion d'"utilisateur" et précise que ce qui caractérise l'utilisateur (ses attributs) fait également partie du contexte. Par exemple, un attribut de l'utilisateur peut être son humeur/état mental jouant un rôle dans son contexte. Selon qu'il soit heureux ou triste, un utilisateur n'aura pas les mêmes préférences en ce qui concerne le film ou la musique qu'il souhaite consulter [Chen et al., 2016].

Dans la définition donnée par le TLFi, il est question de "circonstances". Dans le cadre des SR, ces circonstances ont été divisées en deux notions qui sont les conditions et les dimensions contextuelles (également appelées facteurs contextuels) [Adomavicius and Tuzhilin, 2011]. Une dimension contextuelle correspond à un groupe de conditions contextuelles qui partagent une certaine logique. Autrement dit, les conditions contextuelles sont les valeurs possibles de dimensions contextuelles. Dans le cas où la météo est la dimension contextuelle, ensoleillé, pluvieux et neigeux sont 3 exemples de conditions contextuelles.

Pour certains auteurs, il existe même plusieurs formes de contexte. [Dourish, 2004] par exemple distingue 2 visions du contexte qui sont la vision représentationnelle et la vision interactionnelle. La vision **représentationnelle** se base sur 4 postulats :

- le contexte est une forme d'information ;
- le contexte est délimitable ;
- le contexte est stable ;
- le contexte est indépendant de l'activité sous-jacente.

Selon cette vision, le contexte peut être décrit en utilisant un ensemble d'attributs qui est connu *a priori* et la structure de ces attributs contextuels n'évolue pas au cours du temps. Inversement, selon la vision **interactionnelle**, les caractéristiques contextuelles ne sont plus figées dans le temps, mais sont définies dynamiquement. Ainsi, plutôt que de dire que le contexte agit comme un ensemble de conditions sous lesquelles une activité se déroule, cette vision assume qu'il y a une relation cyclique entre le contexte et l'activité, où l'activité donne lieu à un contexte et où le contexte influence l'activité.

Au final, le contexte peut être défini en le rattachant à 3 concepts [Kaminskas and Ricci, 2012], qui sont :

- l'utilisateur (informations démographiques, état mental, niveaux d'activité...) ;
- l'environnement (position géographique de l'utilisateur, météo, heure de la journée, période de la semaine et/ou de l'année...) ;
- le multimédia (les autres informations auxquelles est exposé l'utilisateur, par exemple dans le domaine de la musique, la pochette d'un album ou une courte biographie du groupe de musique).

Le contexte est donc une notion qui peut être définie de différentes manières et certains auteurs affirment que deux contextes ne sont jamais exactement les mêmes [Zimmermann et al., 2007].

En effet, étant donné les nombreuses dimensions physiques et mentales qui interviennent dans le contexte, obtenir deux contextes identiques est très complexe, d'autant qu'un utilisateur donné observé à l'instant t n'est plus exactement le même à un instant $t + 1$ (lorsqu'il vient d'effectuer une action telle que consulter, noter ou encore acheter un item). Même si fournir une unique définition de ce que représente exactement le contexte d'un utilisateur s'avère complexe, l'exploitation des données contextuelles qui sont accessibles par un système permet néanmoins d'adapter les recommandations au contexte. Bien qu'il puisse exister différentes définitions du contexte, l'adaptation des recommandations à ce dernier (quelle que soit la définition utilisée) permet d'améliorer la qualité des recommandations.

Résumé : le contexte s'articule autour de 3 concepts (utilisateur, environnement, multimédia) auxquels sont associés des attributs appelés conditions contextuelles. Ces conditions contextuelles possédant un lien logique entre elles sont regroupées au sein d'une même dimension contextuelle. Le contexte peut être défini comme étant une dimension fixe (vision représentationnelle) ou encore évolutive (vision interactionnelle).

1.2.2 Intégration du contexte dans le processus de recommandation

Les différentes représentations du contexte que nous avons présentées montrent que le contexte est une notion complexe à définir, car sa définition dépend des usages et du domaine dans lesquels nous y faisons référence. Cela montre également que cette notion peut évoluer et qu'il est possible de proposer de nouvelles définitions du contexte. Cependant, outre la définition utilisée, l'exploitation du contexte demeure critiquable au vu des informations utilisées pour l'intégrer dans le processus de recommandation. Afin d'argumenter nos propos, nous allons nous intéresser aux moyens par lesquels les SR s'adaptent au contexte des utilisateurs. Pour cela, nous allons présenter les trois étapes majeures des SR exploitant le contexte qui sont : la représentation, la récupération et enfin l'exploitation des informations contextuelles.

Représenter l'information contextuelle

D'après l'Équation 1.2 présentée dans la Section 1.1.1, nous avons vu que les utilisateurs sont liés aux items à travers des notes représentant leurs préférences et qu'un utilisateur n'est lié à un item qu'au travers d'une seule valeur représentant sa préférence. Cependant, nous avons expliqué dans la section précédente qu'un utilisateur pouvait avoir des préférences différentes en fonction de son contexte et que de ce fait, un utilisateur et un item ne sont plus liés par une préférence unique, mais pas des préférences qui sont dépendantes du contexte. Dans ce cas, la fonction d'utilité prend alors la forme suivante :

$$v = U \times I \times C \rightarrow R \quad (1.6)$$

La prise en compte du contexte ajoute dans ce cas une nouvelle dimension à la représentation du rapport entre un utilisateur et un item. Dès lors que la dimension contextuelle est incluse dans le modèle de préférence d'un utilisateur, ce dernier ne possède plus une, mais plusieurs préférences pour un seul item. Par exemple, un utilisateur qui apprécie une musique en soirée avec des amis et lorsqu'il se réveille témoigne d'une préférence forte dans ces deux contextes pour cette musique. Inversement, lorsqu'il lit un article scientifique, il n'apprécie pas écouter cette même musique et possède alors une préférence faible pour cette dernière dans ce contexte. Dans ce cas, il existe 3 préférences pour cet utilisateur et ce seul item.

Alors que les préférences des utilisateurs étaient représentées dans une matrice à deux dimensions (cf. *supra* Figure 1.1, Section 1.1.1, p.15) lorsque le contexte n'était pas pris en compte, l'inclusion de ce dernier a pour conséquence d'ajouter des dimensions supplémentaires. Une représentation possible des préférences contextuelles des utilisateurs pourra être la suivante :

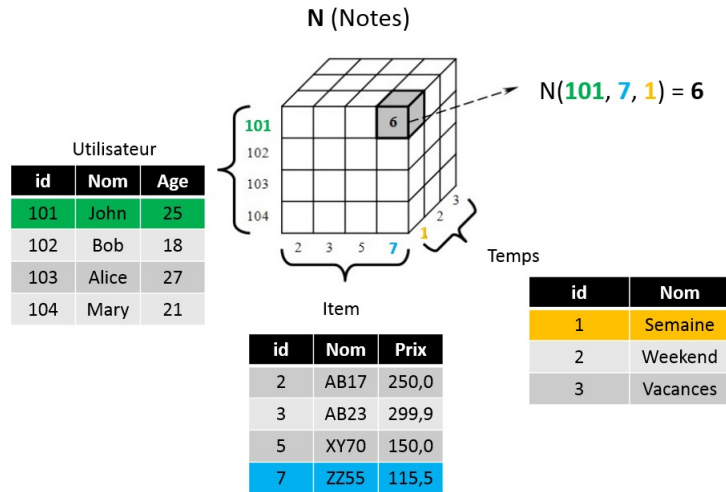


FIGURE 1.4 – Matrice permettant de représenter les préférences contextuelles des utilisateurs en lien avec l’environnement et le média [Adomavicius and Tuzhilin, 2011]

Dans la Figure 1.4, les profils utilisateurs ne contiennent qu’une seule dimension contextuelle qui est le moment de la semaine et qui constitue une dimension liée à l’environnement. Cette dimension s’ajoute à deux dimensions permettant de représenter les préférences de l’utilisateur : la dimension utilisateur et la dimension item (qui sont les deux dimensions utilisées lorsque le contexte n’est pas pris en compte). Dans ce cas, nous avons une dimension contextuelle qui ajoute une dimension à la matrice de préférence et cette nouvelle dimension est définie par le nombre de conditions contextuelles qui la compose, c’est-à-dire 3 dans notre cas (semaine, vacance et weekend). Ainsi, un utilisateur peut posséder 3 préférences pour un item qui sont chacune dépendante des conditions contextuelles. Si un système définit le contexte comme étant composé de 10 dimensions contextuelles (météo, période de la journée...), le profil utilisateur sera alors contenu dans une matrice contenant 12 dimensions. Si nous considérons que chaque dimension contextuelle possède au moins 2 conditions (sans quoi l’ajout d’une dimension n’apporte aucune information), il pourra donc exister 22 préférences différentes²⁸ pour un utilisateur vis-à-vis d’un item et il sera impossible d’obtenir une représentation graphique tridimensionnelle comme dans la Figure 1.4 ci-dessus. De plus, nous n’avons parlé jusqu’à présent que des concepts “utilisateur” et “environnement”. Or le contexte est également constitué du concept “média” qui ajoute des dimensions supplémentaires dans la représentation des préférences des utilisateurs. Quel que soit le nombre de dimensions de la matrice de préférence, l’accès aux préférences d’un utilisateur repose sur le même principe : il suffit de sélectionner dans chaque dimension de cette matrice les conditions pour lesquelles nous souhaitons connaître les préférences (c’est-à-dire pour un contexte donné), puis, en recoupant chacune de ces dimensions, nous obtenons les préférences contextuelles si des préférences dans ce contexte sont connues. Par exemple, si l’on souhaite connaître la note

28. Considérer en plus les combinaisons de ses conditions contextuelles augmentera davantage cette espace de représentation. Par exemple, s’il pleut **et** qu’il est seul un utilisateur peut après un item alors qu’il peut ne pas l’apprécier lorsqu’il pleut **et** qu’il se trouve avec ses amis.

attribuée par l'utilisateur nommé "John" pour l'item "ZZ55" en "semaine", il suffit de chercher la valeur contenue dans la matrice aux coordonnées correspondantes à ces informations. Dans ce cas, il s'agit de la préférence contenue aux index 101 (id de l'utilisateur), 7 (id de l'item) et 1 (id de la condition contextuelle choisie) et pour lesquels la valeur correspondante est 6. La recherche peut être effectuée en utilisant l'ensemble des dimensions ou à partir d'un sous-ensemble de dimensions/conditions afin d'augmenter le nombre de préférences alors retrouvées. En effet, plus le nombre de dimensions/conditions utilisées est faible, plus le nombre de préférences contextuelles retrouvées sera important. Inversement, plus la recherche sera spécifique, c'est-à-dire concernant beaucoup de conditions contextuelles, moins le nombre de préférences retrouvées sera important. Par exemple, le nombre de préférences obtenues en recherchant uniquement les items appréciés dans la condition "ensoleillé" sera forcément égal sinon supérieur aux préférences obtenues dans le contexte "ensoleillé & après-midi & famille & heureux".

Il convient de préciser que même si un utilisateur peut avoir des préférences différentes, voire complètement opposées en fonction des contextes, tout contexte différent ne donnera pas nécessairement lieu à des préférences différentes. Étant donné qu'un contexte est différent d'un autre à partir du moment où l'une des conditions contextuelles change, il est alors possible que cette condition n'impacte pas les préférences. Par exemple, qu'il pleuve, vente ou neige, un utilisateur peut tout autant apprécier une même musique du moment qu'il est avec ses amis en soirée. Cependant il peut ne pas vouloir écouter cette musique lorsqu'il est seul et qu'il pleuve ou neige.

Récupérer l'information contextuelle

Afin de pouvoir être incluses dans les modèles de préférences puis exploitées, les préférences contextuelles des utilisateurs doivent tout d'abord être récupérées par le système. Pour ce faire, il existe 3 méthodes principales permettant de déterminer le contexte des utilisateurs [Adomavicius and Tuzhilin, 2011].

- la collecte explicite : les informations concernant le contexte de l'utilisateur sont obtenues directement auprès de l'utilisateur, en lui demandant de renseigner son contexte. Par exemple, des sites de musique en ligne tels que Rockola²⁹ ou MoodOfTheDay³⁰ demandent explicitement à l'utilisateur de renseigner son humeur afin de proposer des recommandations de musique qui y sont adaptées.
- la collecte implicite : les informations sont obtenues à partir des interactions entre le système et les utilisateurs. Par exemple lorsqu'un utilisateur utilise un service (via le web ou une application mobile), ce dernier peut obtenir des informations contextuelles grâce à la géolocalisation de l'utilisateur, l'heure à laquelle l'utilisateur utilise le système, le bruit environnant, la quantité de lumière captée par l'appareil photo... A l'ère du web social, les utilisateurs fournissent de plus en plus de contenu et il est alors possible d'exploiter les annotations fournies par les utilisateurs pour déterminer le contexte de consultation des items. Par exemple [Hariri et al., 2012] ont exploité les "tags" associés à des *playlists* pour définir le contexte des musiques dans un premier temps et proposer des recommandations contextualisées dans un second temps.
- l'inférence : contrairement aux deux méthodes précédentes, le système va récupérer dans un premier temps des informations qui ne concernent pas directement le contexte de l'utilisateur. Puis dans un deuxième temps, le système va déterminer/inférer à partir de méthodes

29. <http://www.rockola.fm/>

30. <http://www.moodoftheday.fr/gallery/by-mood/>

statistiques (comme les classifieurs naïfs de Bayes par exemple) des situations caractéristiques en utilisant une base d'apprentissage constituée en amont. Par exemple, il est possible d'inférer l'identité (adulte, enfant, *etc.*) d'une personne qui regarde la télévision ainsi que certains critères démographiques en fonction de la fréquence à laquelle les chaînes sont "zappées" [S. Zufryden et al., 1993].

Le contexte possède la particularité d'être très dynamique dans certains cas. Si l'on se place dans le cas d'un utilisateur qui écoute de la musique à partir d'une application de musique, il est possible que son contexte évolue rapidement. Cet utilisateur peut par exemple partir de son bureau pour rentrer chez lui après une journée de travail alors même qu'il ne s'est pas déconnecté du service de musique puisqu'il utilise une application sur son *smartphone*. Nous avons donc affaire dans ce cas à au moins 3 contextes : au travail, en voiture et à la maison. Les contextes mentionnés ici ne font référence qu'au contexte géographique et il faut garder à l'esprit que le contexte est constitué de nombreuses dimensions telles que l'humeur, la météo, l'entourage de l'utilisateur... et tout changement dans l'une des dimensions, c'est-à-dire tout changement de condition donne lieu à un contexte différent. Un contexte étant défini sur la base de l'ensemble des conditions contextuelles des dimensions contextuelles considérées, chaque modification d'une condition donne lieu à un contexte différent. Par exemple, si l'utilisateur commence sa journée de travail par une mauvaise nouvelle, mais qu'il obtient par la suite une promotion, son état mental pourra donner lieu à deux contextes différents malgré le fait qu'il soit toujours dans le même contexte environnemental. Par conséquent, peu importe la définition sur laquelle il repose, le contexte est défini en fonction des informations récupérées par le système et par la mise à jour de ces dernières. Le degré de connaissance des informations permettant de définir le contexte ainsi que l'évolution de cette connaissance dans le temps correspond à ce que [Adomavicius and Tuzhilin, 2011] nomment le caractère observable du contexte. Ce caractère observable du contexte est résumé dans la Figure 1.5 ci-dessous :

Evolution des facteurs contextuels	Connaissance que possède le SR des facteurs contextuels		
	Complètement observable	Partiellement observable	Non observable
Statique	Tout est connu du contexte	Connaissances partielles et statiques du contexte	Connaissances latentes du contexte
Dynamique	Les connaissances du contexte sont dynamiques	Connaissances partielles et dynamiques du contexte	Rien n'est connu du contexte

FIGURE 1.5 – Caractère observable des dimensions contextuelles [Adomavicius and Tuzhilin, 2011]

Alors que dans la meilleure situation possible, la connaissance du contexte de l'utilisateur est complète et dynamique, ce qui permet d'adapter les recommandations à chaque changement du contexte, dans le pire scénario le système ne possède aucune connaissance du contexte de l'utilisateur et ne pourra donc pas s'y adapter. Cependant, dans la majorité des cas d'usage, un SR ne se trouve pas dans ces deux extrêmes, mais plutôt dans les autres configurations. Par exemple, même si un SR est en mesure (d'un point de vue technique) de collecter de nombreux facteurs contextuels, et ce de manière dynamique, l'obtention des informations peut être limitée par des raisons techniques, éthiques ou encore par des choix de l'utilisateur. Un utilisateur peut ainsi choisir de désactiver le partage de sa position géographique empêchant alors un système de connaître pleinement son contexte.

De la même manière que le contexte peut être défini de différentes manières, il peut l'être sur plusieurs niveaux/échelles représentant généralement la précision de la connaissance du contexte. Par exemple, dans le cas où une personne entreprend une démarche d'achat sur un site de e-commerce, nous pouvons distinguer le contexte où la personne recherche un item pour elle-même ou alors pour faire un cadeau. Ces deux situations différentes constituent alors les deux modalités du facteur contextuel "intention d'achat". Chacune de ces deux modalités peut être divisée en une suite de nouvelles modalités qui précisent davantage le contexte de l'utilisateur. Ainsi, dans le cas où il s'agit d'un cadeau, différentes situations peuvent être déterminées en fonction de la personne à qui est destiné le cadeau : pour le conjoint, un ami, un parent... et dans le cas d'un achat personnel, il peut être effectué pour le travail ou pour d'autres motifs. Ces différents niveaux d'information sont hiérarchiques et sont représentés à l'aide d'ontologies qui seront ensuite exploitées afin de fournir des recommandations contextuelles en utilisant les différents niveaux de représentation des informations contextuelles [Chen and Chen, 2014a]. La Figure 1.6 ci-dessous représente par exemple l'organisation des informations contextuelles pour trois dimensions contextuelles différentes :

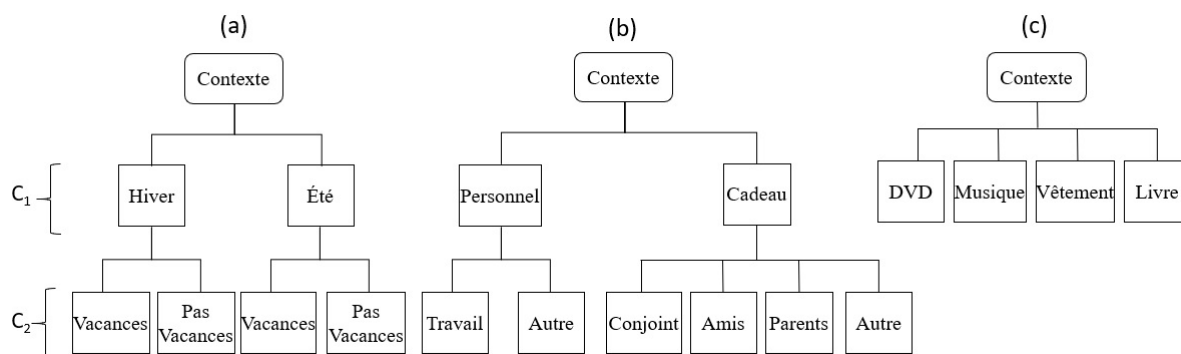


FIGURE 1.6 – Représentation hiérarchique de l'information contextuelle [Panniello and Gorgoglione, 2011]

Cette figure illustre les représentations hiérarchiques de 3 dimensions contextuelles. Dans le premier cas (a), il s'agit de la période de l'année, dans le second cas (b) de l'intention d'achat de l'utilisateur (que nous avons présenté précédemment) et dans le troisième cas (c), la dimension contextuelle est uniquement composée de la catégorie des items d'un catalogue. Outre l'exemple de représentation hiérarchique proposée par les auteurs ([Panniello and Gorgoglione, 2011]) sur cette figure, il est intéressant de noter que l'utilisation de la catégorie des items (c'est-à-dire un de leurs attributs) en tant que dimension contextuelle constitue une approche originale. De plus, leurs travaux montrent que ce facteur contextuel permet une amélioration de la qualité des recommandations par rapport à une approche ne prenant pas en compte le contexte. Cependant, les auteurs ne semblent pas accorder une importance particulière à ces résultats qui démontrent pour la première fois que le contexte peut être défini sans information contextuelle se rapportant à l'utilisateur et à son environnement. L'idée d'exploiter les caractéristiques des items récemment consultés afin de définir le contexte des utilisateurs plutôt que d'utiliser des facteurs contextuels sera également exploitée dans les travaux de [Hariri et al., 2014] dans le domaine de la musique en exploitant des algorithmes de bandit. Cependant, leur approche de validation demeure critiquable dans la mesure où les changements de contextes que le modèle essaye de détecter ne sont pas de véritables changements de contextes. En effet, pour simuler des changements de contextes (sur lesquels ils ont évalué leur approche), les auteurs ont décidé de fusionner deux profils utili-

sateurs et ont considéré qu'être en mesure de détecter les moments où les profils ont été échangés (lorsque l'on passe des préférences d'un utilisateur à l'autre) constituait un moyen d'évaluer la capacité de leur approche quant à la détection de changements de contextes. Or, nous pensons que bien qu'intéressante dans une certaine mesure, cette méthode de validation demeure limitée, car la complexité des changements de comportements (sous l'influence du contexte) d'un unique utilisateur, ne peut être réduite aux préférences de deux utilisateurs distincts.

Au final, même si les ontologies possèdent l'avantage d'offrir une représentation facilement exploitable des informations contextuelles, elles possèdent le défaut d'être *ad hoc* et donc limitées à leurs domaines d'usage. Par exemple, une ontologie des informations contextuelles utilisées dans le domaine du e-commerce sera difficilement transposable au domaine de la musique. En effet, des dimensions contextuelles telles que l'intention d'achat n'interviennent que dans le domaine spécifique du e-commerce.

Exploiter l'information contextuelle

Une fois que les informations contextuelles ont été définies et que les préférences contextuelles des utilisateurs ont été obtenues, ces dernières sont alors exploitées pour calculer les recommandations les plus adaptées aux contextes des utilisateurs. Il existe 3 modèles principaux permettant d'exploiter les informations contextuelles [Adomavicius and Tuzhilin, 2011] : le pré-filtrage, le post-filtrage et le filtrage contextuel. Ces modèles sont illustrés dans la Figure 1.7 ci-dessous :

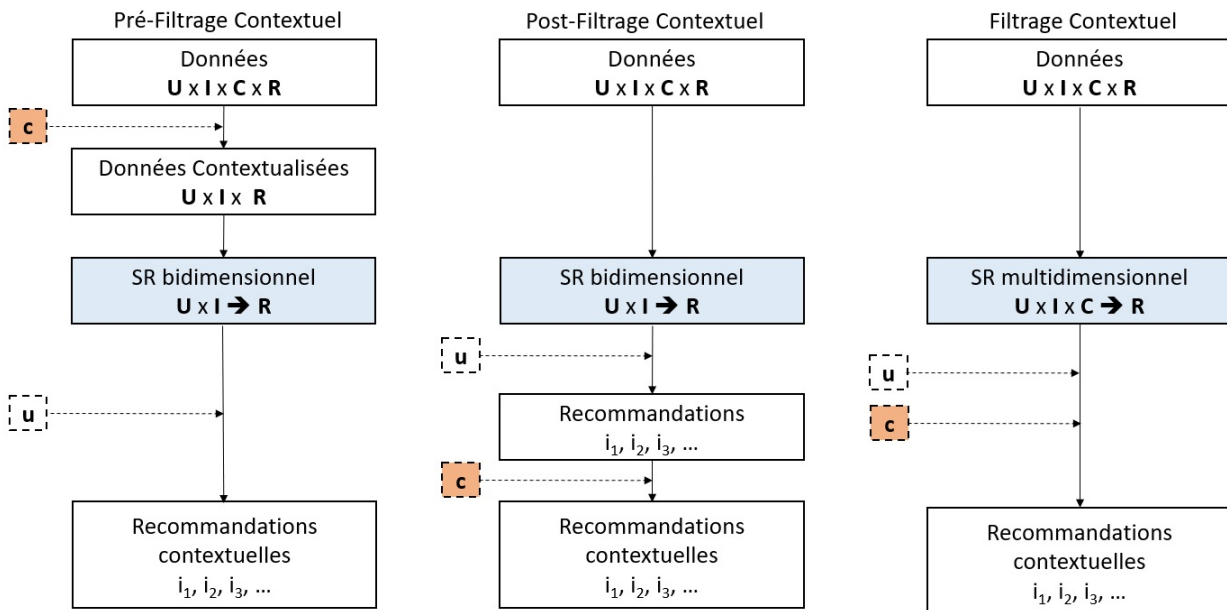


FIGURE 1.7 – Pré-filtrage, post-filtrage et filtrage contextuel [Adomavicius and Tuzhilin, 2011]

Sur cette figure, les deux flèches en pointillés accompagnées des lettres c et u représentent respectivement les moments où interviennent le contexte (c) et les préférences de l'utilisateur (u) dans les différents modèles de recommandation. Le pré-filtrage consiste à filtrer les items qui seront utilisés en amont du processus de recommandation en fonction du contexte courant de l'utilisateur. Par exemple, dans le cas où le système recherche les k plus proches voisins d'une ressource, le système va dans un premier temps rechercher les ressources pour lesquelles les pré-

férences sont connues dans des contextes similaires et ensuite procéder au calcul des plus proches voisins. Cette méthode possède l’avantage de réduire l’espace de recherche étant donné que l’on procède à une restriction du nombre de ressources qui seront impliquées dans les calculs. Cependant, il est nécessaire de recalculer à chaque nouveau contexte les nouveaux voisins et cela possède un impact non négligeable sur les temps de calcul. Inversement, le post-filtrage consiste pour sa part à filtrer les ressources en aval du processus de recommandation pour ne garder par la suite que les ressources qui sont adaptées au contexte désiré. L’avantage de cette méthode est que, les items étant filtrés en aval du calcul des recommandations, il n’est alors pas nécessaire de procéder à un nouveau calcul des recommandations si le contexte est modifié. Enfin, le filtrage contextuel fait intervenir le contexte en tant qu’argument de la fonction de recommandation, contrairement aux deux cas précédents dans lesquels il intervenait soit avant soit après. Dans ce cas, il n’est plus question d’exploiter un SR à deux dimensions, mais d’exploiter l’ensemble des dimensions du contexte dans le calcul des recommandations. Par exemple, la factorisation de matrice qui est une approche très exploitée dans la recommandation peut être généralisée à l’aide de tenseurs permettant de créer des modèles multidimensionnels constitués par l’ajout des dimensions contextuelles [Karatzoglou et al., 2010].

Ainsi, comme nous venons de le voir, le pré-filtrage et le post-filtrage contextuel sont des cadres dans lesquels peuvent être utilisées n’importe lesquelles des approches de recommandation puisqu’il s’agit de filtrer les données avant ou après le processus de recommandation. Concernant les approches impliquées dans le filtrage contextuel, nous pouvons notamment citer l’utilisation de factorisation de matrice [Baltrunas et al., 2011b], de tenseurs [Karatzoglou et al., 2010, Hidasi and Tikk, 2012], de techniques de séparations d’utilisateurs ou d’items³¹ [Said et al., 2011, Zheng et al., 2014a], les approches basées sur l’exploitation de graphes [Wu et al., 2015] ou encore les algorithmes de bandits [Bouneffouf et al., 2012]. Cependant, bien que ces approches contextuelles améliorent indéniablement la qualité des recommandations, nous soutenons que ces approches souffrent de nombreuses limitations.

1.2.3 Limites

Malgré des avantages indiscutables en termes de recommandation, la prise en compte du contexte et notamment l’exploitation des informations contextuelles possèdent certaines limites. Premièrement, même si le fait d’ajouter des dimensions contextuelles dans le modèle de préférence des utilisateurs permet d’augmenter la quantité d’informations disponible, cela augmente également la quantité d’informations manquante. Par exemple, le jeu de données de MovieLens qui contient 20 millions de notes attribuées par 138 000 utilisateurs sur 27 000 films³² contient 99.463 % de données manquantes. Rajouter des dimensions contextuelles aux préférences aura alors pour conséquence d’augmenter l’espace de recherche tout en augmentant le nombre de valeurs inconnues. Certains travaux tels que ceux de [Odic et al., 2011, Campos et al., 2014] ont par ailleurs démontré que l’utilisation d’un nombre trop important de facteurs pouvait nuire aux performances du système et que les utilisateurs ne sont pas nécessairement d’accord quant à la pertinence de certains facteurs contextuels utilisés dans les recommandations. De plus, rappelons que l’utilisation d’ontologies pour représenter hiérarchiquement les données contextuelles possède l’inconvénient d’être *ad-hoc* et dépendant des domaines dans lesquels sont implantés les SR.

31. Le principe des approches par séparation est de séparer un item ou un utilisateur par rapport aux dimensions contextuelles. Par exemple, dans le cas de la dimension “moment de la journée”, chaque item sera séparé en 3 items distincts qui seront $item_{matin}$, $item_{après-midi}$ et $item_{soir}$.

32. <https://grouplens.org/datasets/movielens/>

Deuxièmement, exceptés les travaux de [Panniello and Gorgoglione, 2011, Hariri et al., 2014], la majorité des approches que nous avons pu trouver dans l'état de l'art reposent sur l'exploitation de facteurs contextuels propres à l'individu et à son environnement. Or, ces facteurs contextuels sont issus d'informations contextuelles appartenant au domaine de la vie privée des utilisateurs. Des informations telles que la position géographique [Baltrunas et al., 2012], l'environnement social [Codina et al., 2016] et l'activité de l'utilisateur [Zou et al., 2016] constituent par exemple des données sensibles qui ne devraient pas être récupérées. Obtenir ces informations nécessite la plupart du temps de procéder soit à une collecte explicite, soit à une collecte implicite de la part du système. Dans le cas d'une collecte explicite, ce sont les utilisateurs qui sont à l'origine des facteurs contextuels en renseignant manuellement ces données. Ils peuvent donc décider quelles sont les informations qu'ils donnent ainsi que des moments où ils souhaitent les donner. Cependant, les utilisateurs ne peuvent pas penser à tous les facteurs contextuels et il serait relativement complexe et fastidieux de renseigner à chaque fois le contexte dans lequel ils se trouvent. De plus, un utilisateur peut renseigner son contexte à chaque consultation tout comme ne jamais le renseigner ce qui aura pour conséquence de priver le modèle des données contextuelles sur lesquelles il repose. Inversement, lorsque la collecte est implicite, des informations sont récupérées en permanence par le système. Même si le problème du respect de la vie privée n'est pas spécifique aux systèmes de recommandation contextuels, étant donné qu'en 2005, Cranor [Cranor, 2005] avait montré que la récupération d'informations personnelles constituait une pratique de plus en plus critiquée par les utilisateurs, la collecte d'informations contextuelles repousse davantage les frontières de la vie privée des utilisateurs. Plus récemment, [Knijnenburg and Kobsa, 2013] ont démontré que le fait de questionner les utilisateurs par rapport à leur contexte est une pratique qui n'est pas encouragée, car elle peut générer de la méfiance envers le système. Bien que certaines approches proposent d'encrypter les données des utilisateurs [Nikolaenko et al., 2013], il en résulte que, selon [Tang and Wang, 2015], la majorité des méthodes employées repose sur des mécanismes cryptographiques qui bien que simples à mettre en place, tendent à augmenter les temps de calcul et à diminuer la précision des recommandations. Même si ces derniers proposent une approche novatrice permettant d'améliorer l'aspect "sécurité" des données tout en préservant la qualité des recommandations intégrant le contexte, la préservation de la sécurité des données reste encore un défi majeur puisque des données sensibles sont tout de même collectées. Enfin la mise en place du RGPD déployé cette année permet aux utilisateurs de refuser que certaines informations personnelles soient utilisées par le système. Bien que permettant aux utilisateurs d'obtenir un contrôle sur leurs données, ces mesures constituent un obstacle au SR se basant sur ces informations et des alternatives doivent être proposées.

Troisièmement, outre l'aspect vie privée, la phase de récupération d'informations possède un certain nombre de défauts. Par exemple, il est possible que ces dernières ne soient pas récupérables pour des raisons matérielles ou alors parce que l'utilisateur décide de bloquer l'utilisation de certaines informations ou de certains capteurs (comme l'utilisation de systèmes physiques ou logiciels permettant de cacher/désactiver les webcams ou les micros des ordinateurs portables). Sans facteurs contextuels, les systèmes échoueront *de facto* à fournir des recommandations contextuelles. Quatrièmement, en plus de poser certains problèmes quant au respect de la vie privée des utilisateurs, la recommandation contextuelle souffre d'autres problèmes peu traités dans la littérature. Par exemple, les travaux de [Jiang and He, 2016] sont parmi les rares exemples dans lesquels les auteurs s'interrogent sur l'impact énergétique et temporel de l'architecture de recommandation utilisée. A notre époque, les *smartphones* sont devenus de véritables ordinateurs possédant de nombreuses applications fonctionnant en arrière-plan qui transmettent des données relatives au contexte aux serveurs d'application qui les exploitent afin de fournir des recommandations. Par exemple, un utilisateur peut écouter de la musique pendant qu'il surfe sur un site de e-commerce

tout en recevant des notifications *push*³³ de la part de son application d'actualité, tandis que son application de rencontre en ligne met à jour ses coordonnées afin d'actualiser sa position. Dans ce cas, 4 services échangent de nombreuses informations et consomment donc de la bande passante ainsi que de la batterie. Mettre à jour de manière permanente les données contextuelles des utilisateurs ne peut qu'augmenter cette consommation. Si ces applications ne sont pas suffisamment prévoyantes vis-à-vis de la consommation qu'elles engendrent, les utilisateurs ne manqueront pas de faire savoir leur mécontentement comme cela a pu être le cas avec l'application mobile de Facebook³⁴ par exemple. Une des raisons pour lesquelles l'application provoquait une surconsommation de la batterie était entre autres l'utilisation de la localisation afin d'offrir l'expérience la plus adaptée possible au contexte des utilisateurs.

1.2.4 Discussion

Malgré les bénéfices apportés en termes d'adaptation aux réels besoins des utilisateurs, les approches de recommandation contextuelles se heurtent entre autres à des problématiques de respect et de préservation de la vie privée des utilisateurs. Plutôt que de mettre aux points des approches visant à augmenter la sécurité des informations sensibles obtenues chez les utilisateurs (qui posséderaient tout de même certains inconvénients), nous soutenons que le contexte peut être défini autrement qu'à travers des facteurs contextuels portant sur les individus et leur environnement. Dès lors, lorsque nous ferons référence au contexte tel que nous venons de le voir (c'est-à-dire défini à partir d'informations sur l'utilisateur son environnement) nous emploierons le terme de **contexte explicite**³⁵. Nous avons par ailleurs présenté plusieurs travaux dans lesquels l'adaptation au contexte des utilisateurs se base sur les caractéristiques des items consultés et non sur les facteurs contextuels des utilisateurs. Cependant, nous pensons que ces travaux ne constituent que les prémises d'une nouvelle famille de contextes et que cette idée peut être davantage développée. Nous soutenons à ce propos que l'étude du comportement des utilisateurs vis-à-vis de la diversité et plus particulièrement de son évolution au cours du temps constitue une piste prometteuse.

1.3 La diversité

Parmi l'ensemble des dimensions devant être prises en compte par les SR, la diversité est, avec le contexte, celle qui a reçu le plus d'attention ces dernières années [Kunaver and Požrl, 2017]. La diversité peut faire référence à deux concepts : la diversité en tant que métrique et la diversité en tant que facteur humain (on parle dans ce cas de besoin en diversité). Dans un premier temps, nous présentons la diversité afin de la définir et de comprendre ce qu'elle représente. Nous présenterons également ses différents usages dans les SR. Dans un second temps, nous nous intéressons à la manière dont la diversité est intégrée dans les SR. Nous montrons dans un troisième temps, que les travaux s'intéressant à la diversité se sont essentiellement focalisés sur la diversité en tant que métrique sans pour autant chercher à s'intéresser au besoin en diversité des utilisateurs qui sont pourtant les destinataires des recommandations. En nous appuyant sur certains travaux s'intéressant à l'évolution des besoins en diversité des utilisateurs au cours du

33. Le principe des notifications push est de recevoir des alertes sur un *smartphone* même si l'utilisateur n'utilise pas l'application en question

34. <https://www.nytimes.com/2017/11/24/technology/personaltech/facebook-battery-drain.html>

35. La notion d'explicite n'est pas lié aux préférences explicites et implicites. Le terme d'explicite fait référence au fait que le contexte de l'utilisateur est explicitement défini grâce aux dimensions/conditions contextuelles

temps, nous mettons en lumière les problématiques pour lesquelles nous tenterons d'apporter des réponses dans la suite de ce manuscrit.

1.3.1 Définitions et intérêts

Diversité : Caractère de ce qui est opposé, contradictoire.

Selon la définition donnée par le TLFi³⁶, la diversité fait référence à ce qui est opposé. Pour qu'il puisse y avoir "opposition", il est nécessaire qu'au moins deux entités soient présentes. Par exemple, un item ou un utilisateur ne peut pas être divers s'il est pris seul. Il est toujours question d'être divers par rapport à quelque chose, comme il est toujours question d'être similaire par rapport à quelque chose (une entité ou un groupe d'entités). C'est précisément selon cette notion d'opposition que [Smyth and McClave, 2001] ont défini pour la première fois le terme de diversité dans les SR. Ils définissent la diversité comme étant la mesure opposée à la similarité. La diversité de deux items est définie par [Smyth and McClave, 2001] selon l'Équation 1.7 suivante :

$$Diversité(a, b) = 1 - similarité(a, b) \quad (1.7)$$

D'après cette équation, nous comprenons que plus deux items sont divers et moins ils sont similaires, et réciproquement. La similarité pouvant être calculée de différentes manières, il peut alors exister différentes mesures de diversité pour les entités considérées. La diversité de deux items calculée à partir de l'ensemble des préférences des utilisateurs d'une communauté ne représentera pas la même diversité que celle calculée à partir des caractéristiques des items. Par exemple, il est possible que deux musiques possèdent des caractéristiques musicales très proches et qu'elles soient alors peu diverses de ce point de vue, mais qu'elles soient très diverses en ce qui concerne les utilisateurs ayant apprécié chacune d'elle. Ainsi, lorsqu'il est question de discuter d'une valeur de diversité, il convient de préciser comment cette dernière a été calculée afin de savoir ce que représente cette diversité. En effet, l'Équation 1.7 définit uniquement la diversité par rapport à la similarité et cette équation est valable peu importe la manière dont est calculée la similarité.

Le calcul de la diversité n'est pas uniquement exclusif à deux items et peut être réalisé sur des ensembles d'items. [Smyth and McClave, 2001] ont également proposé une formule permettant de calculer la diversité présente à l'intérieur d'une classe composée de m ressources. La formule est donnée par l'Équation 1.8.

$$Diversité(i_1, \dots, i_m) = \frac{\sum_{j=1..m-1} \sum_{k=j..m} (1 - similarité(i_j, i_k))}{\frac{m}{2} * (m - 1)} \quad (1.8)$$

[Smyth and McClave, 2001] ont également proposé une formule permettant de quantifier la diversité apportée par un item i relativement à une classe C d'items. Ils ont nommé cette mesure diversité relative (DR) et cette dernière se calcule selon l'Équation 1.9.

$$DR(i, C) = \begin{cases} 0 & \text{si } C = \{\}, \\ \frac{\sum_{j=1..m} (1 - similarité_A(i, c_j))}{m} & \text{sinon.} \end{cases} \quad (1.9)$$

36. Trésor de la Langue Française informatisé <http://atilf.atilf.fr/>

Même si la diversité est définie comme étant la mesure opposée à la similarité, la diversité peut se baser sur des métriques indépendantes de la similarité. Par exemple, le coefficient de Gini dont la formule est donnée dans l'équation 1.10 a été utilisé [Fleder and Hosanagar, 2007] afin de quantifier la diversité des ventes dans le cas d'un site de e-commerce. $L(u)$ représente la courbe de Lorenz indiquant le pourcentage de revenus générés par les 10% des items les moins vendus durant une période de temps donnée.

$$G = 1 - 2 \int_0^1 L(u) du \quad (1.10)$$

Le coefficient de Gini est utilisée afin de savoir s'il existe des inégalités dans des distributions (dans le cas de [Fleder and Hosanagar, 2007], les distributions représentent les items vendus). La valeur obtenue permet de savoir si certains items sont proposés tout le temps ($G=1$) ou alors jamais ($G=0$)³⁷. Autrement dit plus le coefficient de Gini est important et moins la diversité des items proposés sera importante.

La diversité est utilisée afin de remplir différentes fonctions dans les SR. Ces fonctions dépendent de la vision de la diversité adoptée. Parmi ces différentes visions, nous présentons tout d'abord la diversité extrinsèque et la diversité intrinsèque. La diversité extrinsèque a pour rôle de combler une incertitude de langage ou une incertitude concernant le profil de l'utilisateur lorsque ce dernier n'est pas suffisamment complet ou qu'une ambiguïté est présente [Radlinski et al., 2009]. Par exemple, lorsqu'un utilisateur fait des recherches en utilisant le terme "jaguar", ne sachant pas si l'utilisateur parle de l'animal ou de la voiture, il est préférable de présenter des résultats couvrant les deux sujets. Les caractéristiques de l'animal étant très distinctes des caractéristiques de la voiture, présenter les deux résultats aura pour conséquence d'introduire de la diversité (calculée dans ce cas sur les attributs des items) et donc de pallier l'incertitude liée à l'ambiguïté soulevée. La diversité intrinsèque, quant à elle, vise à minimiser les redondances dans un ensemble d'items [Clarke et al., 2008]. Lorsqu'une liste de recommandations est proposée à l'utilisateur, il est plus intéressant de proposer des résultats qui sont divers entre eux, mais qui sont tout aussi pertinents en ce qui concerne les besoins de l'utilisateur. Par exemple, si ce dernier écoute plusieurs morceaux de musique de "rock psychédélique", lui proposer uniquement des musiques de Pink Floyd, donc d'un seul artiste appartenant à ce genre, aura pour conséquence de minimiser la diversité intrinsèque. Cela n'est pas forcément souhaitable, car la liste recommandée limite les choix de l'utilisateur à un artiste unique. Cependant, toujours maximiser la diversité intrinsèque peut également être néfaste dans certains cas. Si l'utilisateur souhaite justement spécialiser ses consultations, et que le système diversifie les artistes proposés, l'utilisateur ne trouvera pas les recommandations satisfaisantes. Nous détaillerons cet aspect dans la Section 1.3.3 p. 41 lorsque nous aborderons la question du besoin en diversité. La diversité intrinsèque peut également être utilisée afin de réagir par rapport à une mauvaise recommandation [Billsus and Pazzani, 2000]. Les auteurs proposent une approche de recommandation basée sur le contenu qui, lorsqu'une recommandation n'est pas appréciée/refusée, va réagir en déterminant le ou les items maximisant la diversité (tout en restant liés aux besoins de l'utilisateur) par rapport à l'item apprécié/refusé. L'hypothèse étant la suivante : si un utilisateur n'apprécie pas un item, alors les items très similaires ont une faible probabilité d'être acceptés.

[Adomavicius and Kwon, 2012] proposent également deux autres visions de la diversité dans les SR. La première dite diversité agrégée fait référence à la diversité des items qui sont recommandés sur l'ensemble des recommandations effectuées par un système. Par exemple, si un SR tend à ne recommander qu'une faible proportion d'items (les plus populaires par exemple) à

37. Le coefficient de Gini est une valeur comprise dans l'intervalle $[0, 1]$

l'ensemble des utilisateurs par rapport à l'ensemble des items dont il dispose, ce système possédera une faible diversité agrégée. Cette métrique peut par ailleurs être utilisée afin de mesurer la couverture d'un SR, c'est-à-dire sa capacité à couvrir l'ensemble des items d'un catalogue [Ludewig and Jannach, 2018]. Dans ce cas, la diversité portera sur le nombre d'items différents ayant été recommandé et l'Équation 1.8 ne pourra pas être utilisée. La seconde est appelée diversité individuelle et permet de mesurer la quantité de diversité présente au sein d'une liste de recommandations. Un SR de musique ne proposant que des items appartenant à un même artiste à chaque liste qu'il génère possédera dans ce cas une diversité individuelle faible. L'équation 1.8 peut être utilisée afin de mesurer la diversité individuelle d'un SR étant donné qu'il s'agit de quantifier la diversité présente dans une liste d'items.

Nous précisons que la diversité individuelle et la diversité agrégée ne sont pas liées. Un SR peut très bien recommander des listes de recommandations dans lesquelles les items sont très similaires les uns des autres (diversité individuelle faible) tout en proposant pour chaque utilisateur des items différents (diversité agrégée forte). Inversement, un SR peut proposer uniquement un ensemble de quelques items qui sont toujours les mêmes pour tous les utilisateurs (diversité agrégée faible), mais étant très divers les uns des autres (diversité individuelle forte).

1.3.2 Intégration de la diversité dans la recommandation

[Bradley and Smith, 2001] sont les premiers à proposer un algorithme incluant la métrique de diversité dans les recommandations. L'algorithme qu'ils ont développé est un algorithme glouton borné dont le but est de proposer aux utilisateurs des items similaires à l'item cible (la requête de l'utilisateur) tout en étant divers entre eux (diversité intrinsèque). Le principe de leur algorithme est le suivant : premièrement, l'algorithme sélectionne un ensemble d'items de taille K où les items sont les plus similaires à l'item cible en se basant sur les attributs des items. Puis dans un second temps, l'algorithme procède à un reclassement des recommandations dans le but d'obtenir une liste de recommandations maximisant la diversité. Pour ce faire, l'algorithme crée un sous-ensemble appelé top- N (où $N < K$) en sélectionnant itérativement l'item offrant la meilleure qualité calculée selon l'Équation 1.11. Ainsi, chaque item ajouté à la liste de recommandations (top- N) est l'item qui apporte le plus de diversité par rapport aux items déjà présents dans la liste.

$$Qualité(i, t, C) = Similarité(i, t) \times DR(t, C) \quad (1.11)$$

Utilisant ce principe de reclassement des recommandations, plusieurs adaptations seront présentées afin de proposer un top- N à la fois précis par rapport à la requête et suffisamment divers. [Radlinski et al., 2009]. [Zhang and Hurley, 2008] utilise une approche binaire afin de conserver une similarité raisonnable tout en maximisant la diversité dans le top- N proposé à l'utilisateur. Certains auteurs tels que [Schafer et al., 2002, Jahrer et al., 2010, Candillier et al., 2013] parleront de méta-systèmes dont le principe repose sur la sélection d'items de différentes listes obtenues à partir de différents RS utilisant des métriques de similarités différentes. Alors que [Schafer et al., 2002, Jahrer et al., 2010] s'intéressent aux items communs entre les différentes approches utilisées, [Candillier et al., 2013] se focalisent sur la diversité et sélectionnent donc les items permettant de la maximiser.

Les algorithmes proposés visent alors à résoudre ce que la communauté appelle le dilemme (ou encore compromis) diversité-précision. Le terme de dilemme est utilisé car la précision est souvent opposée à la métrique de similarité. Cela étant dû au fait qu'il est complexe d'obtenir des listes de recommandations dont la précision est identique, comparé au cas où seule la précision est prise en compte.

L'introduction de la diversité a également été investiguée dans le cadre de systèmes à base de filtrage collaboratif. La différence majeure par rapport au filtrage par contenu est qu'il n'est pas possible de se servir des caractéristiques des items et, de ce fait, les métriques de diversité sont uniquement calculées à partir des notes attribuées par la communauté d'utilisateurs. La métrique nommée similarité intra-liste (plus couramment appelée ILS *Intra List Similarity* proposée par [Ziegler et al., 2005] et utilisée pour le reclassement du top-N fut la première approche visant à intégrer de la diversité dans le filtrage collaboratif. Certains travaux tels que ceux de [Said et al., 2012], [Boim et al., 2011] et [Castagnos et al., 2014] s'attacheront à adapter les méthodes de clustering afin de pouvoir injecter de la diversité dans un cadre collaboratif. Une autre approche proposée par [Vargas and Castells, 2014] consiste à recommander non plus des ressources aux utilisateurs mais de recommander des utilisateurs aux ressources. Cette approche a pour conséquence d'augmenter la quantité de diversité présente dans les recommandations. Les items présents dans le catalogue d'items exploité par un SR ne sont pas tous consultés uniformément. Alors qu'une petite proportion items est très souvent consultée et par la même occasion souvent recommandée, la majeure partie des items est souvent peu consultée mais également peu recommandée. Cet ensemble d'items peu consultés au regard des items les plus populaires est appelé longue traîne. Certains travaux se sont alors intéressés à tirer profit de cette inégalité de répartition connue également sous le nom de phénomène de longue traîne dans leurs approches de recommandation [Celma, 2008, Park and Tuzhilin, 2008].

Les algorithmes permettant d'intégrer de la diversité dans les recommandations peuvent être divisés en deux catégories : ceux se basant dans un premier temps sur des approches de recommandations basées sur le contenu, collaboratives ou hybrides sans se préoccuper de la diversité. Puis, dans un second temps, ces algorithmes cherchent à déterminer quels sont les items qui maximisent la diversité. Autrement dit, ces algorithmes vont procéder à un reclassement des top-N items à recommander afin que la liste de recommandations finalement présentée à l'utilisateur maximise la diversité. L'autre catégorie d'algorithmes s'intéresse pour leur part à prendre en compte la diversité directement dans leur processus de calcul des recommandations. Tout comme le contexte, nous pouvons alors parler d'algorithmes basés sur un post-filtrage et d'algorithmes de modélisation de la diversité. Alors que les algorithmes de post-filtrage sont relativement nombreux dans la littérature, les algorithmes visant à modéliser la diversité sont moins plus restreints [Kunaver and Požrl, 2017].

[Foulonneau et al., 2014] proposent également plusieurs stratégies permettant d'inclure de la diversité dans les recommandations, à savoir :

- augmenter la taille de la liste des items recommandés ;
- ajouter des items choisis aléatoirement ;
- mélanger les stratégies de recommandation (systèmes hybrides) ;
- trouver des clusters proches des clusters d'intérêt d'une personne donnée, ce qui permet de recommander des items différents tout en minimisant le risque de proposer des items n'intéressent pas l'utilisateur [Onuma et al., 2009] ;
- sélectionner des items de longue traîne ayant le moins de chance de déplaire [Celma, 2008] ;
- compenser le manque de notation (*sparsity*) en notant des clusters d'items plutôt que de les noter individuellement [Park and Tuzhilin, 2008].

Il existe de nombreuses approches permettant de diversifier les recommandations mais également de nombreuses métriques visant à quantifier cette dimension. Cependant, maximiser la quantité de diversité n'est pas nécessairement la garantie de proposer des recommandations de qualité, c'est-à-dire satisfaisant au mieux les besoins des utilisateurs. Ainsi, bien qu'un SR soit

capable d'introduire davantage de diversité dans ses recommandations par rapport à un autre SR, nous ne pouvons pas conclure que ce dernier est moins performant que le premier. En revanche, il est tout à fait juste d'affirmer que le premier SR est plus performant que le second dans la tâche d'introduction de la diversité. Depuis qu'elle a été montrée comme étant une dimension importante devant être prise en compte, la diversité est devenue un score à maximiser tout comme l'a été la précision avant elle. Or, comme le présentent [Kunaver and Požrl, 2017] dans leur conclusion à propos de la diversité dans les SR, il existe encore certains défis à propos de la diversité devant être relevés. Parmi ces défis, nous pouvons notamment noter que la perception de la diversité par les utilisateurs occupe une place importante et les auteurs déclareront qu'il est préférable de s'intéresser à cette notion plutôt que de développer de nouvelles approches permettant d'introduire de la diversité dans les recommandations. La perception des utilisateurs vis-à-vis de la diversité est également liée au besoin en diversité représentant la propension que possèdent les utilisateurs à désirer ou non de la diversité dans les recommandations. Nous allons donc dans la section suivante nous intéresser au besoin en diversité qui constitue un facteur humain devant être pris en compte dans le modèle *TAM* (*cf. supra* Figure 1.3, p. 23) afin que les utilisateurs acceptent et adoptent un SR.

1.3.3 Le besoin en diversité

La diversité permet d'éviter de proposer des items trop redondants, de pallier certaines incertitudes, réagir à la suite d'une mauvaise recommandation. . . . Bien qu'elle puisse également servir à mesurer la capacité d'un SR à couvrir l'intégralité d'un catalogue d'items, la diversité est avant tout destinée aux utilisateurs. Plusieurs études telles que [Candillier et al., 2011, Castagnos et al., 2014] confirment que la diversité est effectivement une dimension des recommandations perçue par les utilisateurs. Certains travaux montreront par ailleurs montrés qu'un manque de diversité peut générer de la frustration auprès des utilisateurs [Lathia, 2010]. Pour autant, chercher à maximiser la diversité lors de chaque recommandation (ou liste de recommandations) proposée n'est pas un objectif devant être atteint. [Lathia et al., 2010] expliquent à ce propos que la diversité ne doit pas être maximisée à chaque cycle de recommandation sous peine d'instaurer une incompréhension des utilisateurs. Dans le même temps, il est démontré que la quantité de diversité à proposer dans les recommandations ne doit pas être figée et impacte significativement la perception qu'ont les utilisateurs vis-à-vis d'un système [Castagnos et al., 2013b]. Dans leurs travaux, [Castagnos et al., 2013a] proposent de classer les travaux s'intéressant à la diversité en deux catégories : ceux qui s'intéressent à **pourquoi** et ceux qui s'intéressent à **comment** diversifier les recommandations. Alors que ces deux axes sont relativement étudiés, nous remarquons qu'un troisième axe n'a pour sa part reçu que peu d'attention, il s'agit de l'axe temporel de la diversité qui s'illustre à la question du **quand**. Les premiers éléments de réponses quant à la prise en compte de l'aspect temporel dans le but d'adapter le niveau de diversité à proposer aux utilisateurs ont été fournis par ces mêmes auteurs quelques années auparavant.

[Castagnos et al., 2010] démontrent, dans une étude utilisateur visant à étudier leur comportement dans une démarche d'achat sur un site de e-commerce, que le niveau de diversité que doit proposer un SR ne doit pas être le même au cours du temps. Suite à ce constat, ils proposent alors une courbe théorique représentée dans la Figure 1.8. Cette figure représente le niveau théoriquement adéquat de diversité (en termes d'attributs des items) que doit proposer un SR au fur et à mesure du temps dans le parcours de consultation des utilisateurs. Nous pouvons constater qu'il existe 3 étapes principales pour lesquelles le niveau de diversité doit être adapté. Dans un premier temps, sachant que les utilisateurs ont besoin de se faire une idée des produits présents dans le catalogue, le système doit maximiser le niveau de diversité proposé. Cela permet de ne

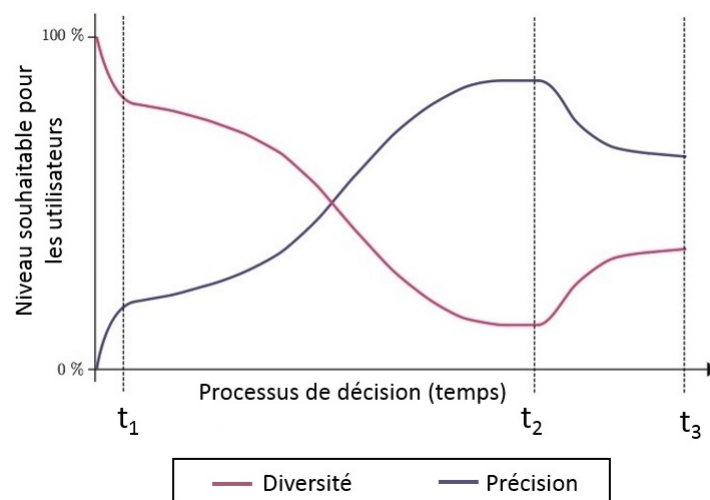


FIGURE 1.8 – Modèle temporel de similarité diversité [Castagnos et al., 2010]

pas enfermer l'utilisateur dans une catégorie de produits ou selon certaines caractéristiques alors que les besoins de l'utilisateur ne sont pas encore déterminés. Autrement dit, cette phase permet à l'utilisateur de “balayer” un large spectre d'items, ce qui lui permet alors de prendre connaissance des différentes alternatives possibles. À mesure que l'utilisateur consulte des produits, le niveau de diversité proposé doit alors diminuer progressivement. Cette diminution a pour but d'aider l'utilisateur à spécialiser ses besoins. Par exemple, le SR pourra proposer les mêmes produits, mais avec des caractéristiques différentes (une couleur différente, une taille différente) contrairement à la première phase où le système recommandera par exemple des items de nature différente. Enfin, lorsque la décision d'achat approche, le système devra augmenter le niveau de diversité afin de permettre à l'utilisateur de conforter son choix.

Adapter le niveau de diversité des recommandations en fonction de certaines situations a également été proposé dans les travaux de [Hasan et al., 2014]. Les auteurs proposent d'adapter le niveau de diversité à proposer aux utilisateurs en fonction de leurs actions et plus particulièrement lorsque ces derniers reformulent la requête leur permettant d'obtenir des résultats. Cependant, même si la prise en compte de l'action de l'utilisateur permet de définir la quantité de diversité à introduire dans les recommandations et permet d'améliorer la qualité de ces dernières, ce processus nécessite l'intervention de l'utilisateur. Or, en plus d'avoir à agir pour reformuler sa requête initiale, l'utilisateur doit également penser à la manière dont il doit reformuler sa requête ce que nécessite alors un effort cognitif. Ainsi, les travaux de [Castagnos et al., 2010, Hasan et al., 2014] nous indiquent que le niveau de diversité à proposer dans les recommandations est dépendant de la tâche que l'utilisateur est en train de réaliser. Étant donné que l'intention d'achat constitue une des dimensions du contexte, nous pouvons par extension supposer qu'il doit exister un lien entre la diversité et le contexte dans lequel se trouvent les utilisateurs.

Le niveau de diversité optimal à inclure dans les recommandations n'est pas uniquement dépendant de l'action réalisée par l'utilisateur. Le niveau de diversité à proposer est également dépendant des utilisateurs. [Di Noia et al., 2014] ont par exemple présenté une approche exploitant l'historique des utilisateurs pendant une période donnée afin de savoir si ces derniers diversifient ou non leurs consultations. Puis, en se basant sur la propension qu'ont les utilisateurs à diversifier leurs consultations, leur algorithme adapte le niveau de diversité ce qui mène à une augmentation

de la satisfaction des utilisateurs. [Tintarev et al., 2013] démontrent par exemple qu'il existe une relation entre l'ouverture d'esprit des utilisateurs et le niveau de diversité que ces derniers désirent. Les auteurs démontrent que plus un utilisateur possède un score important en termes d'ouverture d'esprit, plus ce dernier sera enclin à revoir des recommandations diverses. Certains travaux proposent des SR permettant aux utilisateurs de contrôler le niveau de diversité qu'ils souhaitent obtenir dans les recommandations [Aytekin and Karakaya, 2014]. [Ekstrand et al., 2014] et [Javari and Jalili, 2015] ont par ailleurs proposé une approche possédant un système de *feedback* concernant la précision ainsi que la diversité des recommandations dans le but d'adapter les recommandations en conséquence.

Dès lors, la diversité ne doit plus constituer uniquement une métrique nécessaire qu'un SR se doit maximiser mais doit être perçue comme une composante évolutive qui doit être maîtrisée précisément. Nous soutenons fortement le point de vue selon lequel la consultation d'un item par un utilisateur ne constitue pas seulement une trace d'usage mais s'inscrit dans un cadre temporel dont la prise en compte est déterminante pour fournir les recommandations suivantes. Contrairement à l'approche proposée par [Di Noia et al., 2014] qui s'intéresse à attribuer un score de diversification aux utilisateurs, nous soutenons qu'une prise en compte du niveau diversité de l'historique récent s'avère être une piste prometteuse qui n'a à notre connaissance pas encore été explorée.

1.3.4 Diversité, nouveauté et serendipité

Avant de clore cette section sur la diversité, nous présentons deux autres concepts avec lesquels la diversité ne doit pas être confondue : la sérendipité et la nouveauté. Comme nous venons de le voir, la diversité fait référence à ce qui est opposé, ce qui est différent. La nouveauté pour sa part, fait référence à ce qui n'a pas encore été présenté à l'utilisateur. La plupart du temps, la nouveauté se situe à l'échelle d'un item en tant qu'entité unique, c'est-à-dire lorsqu'un item ou des items n'ont encore jamais été présentés à un utilisateur. La nouveauté peut également se situer à l'échelle des caractéristiques des items (ce qui est alors exclusif aux approches basées sur le contenu). Par exemple, si un utilisateur n'a jamais consulté de film de science-fiction, lui proposer un film de ce genre relève à la fois de la nouveauté en termes d'item mais aussi en termes de caractéristiques. Au contraire, si l'utilisateur avait déjà consulté des films de science-fiction, cette recommandation n'apporterait de la nouveauté que du point de vue de l'item. Il a été démontré qu'il existe un lien important entre la satisfaction de l'utilisateur et la présence de nouveauté dans les recommandations [Ekstrand et al., 2014]. Par ailleurs, certains auteurs tels que [Kapoor et al., 2015] s'intéressent à l'évolution du besoin en nouveauté chez les utilisateurs et expliquent que la nouveauté ne doit pas nécessairement être maximisée à chaque recommandation. Au contraire, la quantité de nouveauté à présenter ne doit pas être quelque chose de fixé pour chaque utilisateur, mais doit évoluer au cours du temps et cette quantité de nouveauté peut être propre à chaque utilisateur. Bien que ne devant pas être confondue avec la diversité, la nouveauté peut être rattachée avec la diversité dans une certaine mesure. Cela peut par exemple se produire lorsqu'il est question de comparer un item par rapport au profil d'un utilisateur en termes de caractéristiques, ce qui n'est donc valable que pour les approches basées sur le contenu. Si un item présenté à un utilisateur est très divers par rapport à son profil (donc des caractéristiques), alors cet item risque également d'être nouveau. Cependant, il est tout à fait possible de présenter un item à l'utilisateur qui soit nouveau mais qui ne soit pas du tout divers par rapport à son profil. Par exemple, recommander le dernier morceau du groupe préféré d'un utilisateur constitue bien une recommandation nouvelle (du point de vue de

la musique et non de l'artiste) si ce dernier ne l'a pas encore consultée, mais ne constitue pas une recommandation diverse pour autant. Ainsi, bien que pouvant être liées dans une certaine mesure, ces deux dimensions ne doivent pas être confondues et sont traitées indépendamment. [Vargas and Castells, 2011] ont par exemple proposé une méthode d'unification de 3 dimensions qui sont : la diversité, la nouveauté et la précision qu'ils ont testées sur un corpus de films (MovieLens) et de musiques (Lastfm).

La sérendipité correspond quant à elle à la découverte par le hasard de quelque chose qui s'avère pertinent sans que ce soit ce que l'on ait recherché. Par exemple, la célèbre découverte de la pénicilline par Alexander Fleming est le fruit de la sérendipité. En effet, sa découverte est due au désordre régnant dans le laboratoire de ce célèbre inventeur³⁸. Dans le cadre de la recommandation, un exemple de découverte par sérendipité correspond par exemple au cas où un utilisateur qui consulte sur un site de e-commerce uniquement des t-shirts se voit recommander une paire de chaussures et décide de l'ajouter à son panier. Du point de vue de l'utilisateur, il s'agit de sérendipité puisqu'il s'est vu proposer des chaussures qu'il a appréciées alors qu'il consultait à l'origine des t-shirts. Cependant, il convient de préciser que la véritable source de sérendipité ne peut être que le fruit du hasard et donc que toute autre manière de l'amener n'est pas réellement de la sérendipité du point de vue du système. En effet, lors de l'exemple précédent, il est peu probable que l'utilisateur se soit vraiment fait recommander cette paire de chaussures en particulier à un moment précis. De nombreuses études s'intéressent à mettre en place des approches favorisant ce genre de *scenari* [Kotkov et al., 2016]. Les approches collaboratives sont par exemple un moyen de générer de la sérendipité. Dans le cas où le système repose sur le calcul des plus proches voisins en se basant sur les utilisateurs, les items recommandés sont uniquement constitués par les items ayant des préférences importantes pour les voisins considérés. [Candillier et al., 2011] présentent une approche nommée *blogart* dans un jeu de données contenant les parcours des utilisateurs sur la plateforme Overblog³⁹. Leur approche consiste à retourner des articles sélectionnés aléatoirement du blog sur lequel est consulté l'article.

1.4 Bilan

Les SR sont devenus des outils incontournables afin d'assister les utilisateurs dans leurs recherches d'information et ces derniers n'ont cessé de s'améliorer au fil des années. Ces améliorations se sont essentiellement basées sur la prédiction des préférences absolues des utilisateurs et de ce fait, les SR se sont paradoxalement éloignés de leur but : assister les utilisateurs. En effet, en se focalisant exclusivement sur la métrique de précision, les SR ont omis que ces dernières s'adressent à des êtres humains dont les processus décisionnels sont complexes. En réaction à ces constats, certaines approches proposées se sont davantage focalisées sur l'humain que représente l'utilisateur afin de placer ce dernier au coeur du processus de recommandation. Dans ce but, certains travaux cherchent à définir et intégrer les facteurs humains qui jouent un rôle dans l'acceptation des recommandations tout au long du processus de recommandation. Parmi les facteurs humains ayant un impact sur la qualité des recommandations [Jones, 2010, Shani and Gunawardana, 2011], la diversité et le contexte sont les deux principalement étudiés. Alors que la prise en compte du contexte permet d'adapter les

38. <https://www.acs.org/content/acs/en/education/whatischemistry/landmarks/flemingpenicillin.html>

39. <https://www.over-blog.com/>

recommandations à de nombreuses dimensions déterminées par la situation courante de l'utilisateur (entourage, environnement, météo, état mental...), la diversité possède quant à elle de nombreuses utilités. Elle permet notamment de réagir suite à une mauvaise recommandation et aide les utilisateurs à discerner les alternatives possibles [McGinty and Smyth, 2003], de permettre aux utilisateurs à découvrir de nouvelles ressources, de couvrir l'ensemble du catalogue d'items [Vargas and Castells, 2014] et également d'assister les utilisateurs lorsqu'ils doivent prendre une décision [Castagnos et al., 2010]. Cependant, comme nous l'avons relevé dans cet état de l'art, un certain nombre de problématiques sont encore présentes tant pour le contexte que la diversité et de nombreuses pistes d'amélioration sont envisageables.

Malgré des améliorations indéniables de la qualité des recommandations fournies en prenant en compte le contexte [Adomavicius and Tuzhilin, 2005, Adomavicius et al., 2011b], la majorité de ces approches nécessitent d'effectuer une collecte des données personnelles afin de définir le contexte des utilisateurs. Or, comme nous l'avons décrit, cette étape soulève de nombreux problèmes éthiques puisque cette démarche transgresse le respect de la vie privée des utilisateurs. Même si certaines approches alternatives ont été proposées [Hosseinzadeh Aghdam et al., 2015] [Hariri et al., 2015], les méthodes de validation utilisées ou encore l'impossibilité de fournir des explications aux utilisateurs vis-à-vis de la contextualisation des recommandations sont des points discutables. De ce fait, parvenir à mettre au point de nouvelles méthodes permettant une prise en compte du contexte ne reposant pas sur une collecte d'informations sensibles et permettant une explication des recommandations est un défi qui reste encore non résolu.

En ce qui concerne la diversité, nous avons montré que bien que très utile dans de nombreux cas, elle pouvait susciter la méfiance des utilisateurs si elle est proposée en de mauvaises proportions ou à des moments inopportuns. Cependant, lors de la constitution de notre état de l'art, nous n'avons pas trouvé de travaux proposant une approche permettant de connaître et d'adapter le niveau de diversité à proposer aux utilisateurs en fonction de leurs besoins réels. En effet, dans la majorité des approches proposées s'intéressant à la diversité, il est question de maximiser cette dernière en toutes circonstances tout en s'assurant que les items proposés soient pertinents. Or, tout comme le contexte des utilisateurs impose de reconsidérer les préférences courantes de l'utilisateur que l'on recommande, le niveau de diversité à proposer doit également être adapté au cours du temps et ne pas être maximisé en permanence. Certains travaux ont par ailleurs démontré que le niveau de diversité souhaité par les utilisateurs dans les recommandations était dépendant du contexte de ces derniers [Castagnos et al., 2009]. Sur la base de ces travaux, nous pensons que le corollaire de ce postulat constitue un moyen de proposer une nouvelle approche de recommandation contextuelle. Autrement dit, nous faisons l'hypothèse que l'étude du niveau de diversité peut être utilisée afin de déterminer le contexte de l'utilisateur.

Dans cet état de l'art, nous n'avons pas mentionné certaines approches telles que les approches de recommandation basées sur les sessions [Ludewig and Jannach, 2018] ou encore celles prenant en compte la dimension temporelle (*Time-aware recommender systems* également appelés TARS) [Campos et al., 2014, de Borba et al., 2017]. Premièrement, nous avons écarté les approches à base de session, car ces dernières sont définies par la durée s'écoulant entre la connexion (ou le moment où l'utilisateur commence à interagir avec un service) d'un utilisateur et sa déconnexion (ou le moment où l'utilisateur cesse d'interagir avec un service). Or, nous soutenons l'idée selon laquelle il peut exister plusieurs contextes au sein d'une même session. Par exemple, un utilisateur peut rechercher des cadeaux pour différentes personnes durant la même session. Deuxièmement, les approches basées sur la prise en compte de l'aspect temporel s'intéressent majoritairement à l'évolution des préférences des utilisateurs dans le temps. Par exemple, ces approches cherchent à modéliser la perte d'intérêt ou le regain d'intérêt d'un utilisateur pour

des items en fonction du temps. Certains TARS possèdent également un fonctionnement similaire aux approches basées sur le contexte, à la différence qu'ils ne s'intéressent qu'à la dimension temporelle là où cette dernière ne constitue que l'une des dimensions étudiées dans les approches contextuelles. Bien que constituant un aspect intéressant du point de vue de la recommandation, ces approches ne sont pas directement liées à notre problématique dans laquelle nous nous intéressons à la diversité ainsi qu'à l'ensemble des dimensions contextuelles.

Dans cette thèse, nous proposons donc de nous servir de l'évolution de la diversité au cours du temps afin de présenter une nouvelle approche de recommandation contextuelle. N'ayant pas trouvé dans la littérature d'approche permettant de modéliser l'évolution temporelle de la diversité au cours du temps, mettre au point le premier modèle permettant d'y parvenir constitue la première contribution que nous présentons dans cette thèse. Avant de présenter ce modèle, nous allons exposer dans le chapitre suivant les deux corpus de données que nous utilisons dans ce manuscrit.

Constats : Afin de proposer des recommandations adaptées aux besoins des utilisateurs, les SR se doivent de prendre en compte les facteurs humains inhérents à la prise de décision dans leur processus de recommandation. Parmi ces facteurs humains, nous nous intéressons en particulier à la diversité et au contexte. Alors que le contexte possède certaines limites concernant le respect de la vie privée des utilisateurs, la diversité est quant à elle uniquement considérée la plupart du temps dans la littérature comme un critère de qualité d'un SR. De plus, ces deux dimensions sont dans la majorité des cas traitées comme deux dimensions distinctes. A l'issue de l'état de l'art que nous avons élaboré, nous défendons dans ce manuscrit la thèse suivante : la diversité et plus particulièrement son évolution dans le temps peut être utilisée afin de proposer une nouvelle manière de prendre en compte le contexte ainsi que le besoin en diversité des utilisateurs.

Nos objectifs :

- **1^{er} objectif** : proposer et formaliser une approche de modélisation temporelle de la diversité d'un parcours de consultations ;
- **2^{ème} objectif** : définir une nouvelle famille de contextes basée sur la diversité ainsi que le modèle permettant d'isoler ces contextes ;
- **3^{ème} objectif** : présenter des approches de recommandation basées sur ces contextes.

Chapitre 2

Jeux de données utilisés

Sommaire

2.1	Corpus Lastfm	48
2.2	Corpus Deezer	49
2.3	Attributs collectés	49
2.4	Limites	52

Le contexte est une dimension dynamique qui évolue dans le temps. Comme nous entendons déterminer le contexte des utilisateurs sur la base de l'évolution de la diversité au cours du temps, certaines contraintes doivent être respectées. Les items consultés doivent former des séquences d'items contiguës et les items doivent posséder un temps de consultation relativement faible. De ce fait, nous n'avons pas pu exploiter certains jeux de données réputés dans le domaine de la recommandation tels que le MillionSongDataset⁴⁰ et Yahoo!⁴¹, ou encore les jeux de données de MovieLens⁴² qui ne contiennent pas les informations concernant les moments où ont été consultés les items par les utilisateurs. De plus, les films possèdent des temps de consultation bien trop longs pour espérer isoler le contexte de l'utilisateur sur la base de l'évolution de la diversité. Cependant, notre modèle pourrait être utilisé dans le scénario où l'utilisateur cherche ce qu'il peut regarder et qu'il passe de film en film (de recommandation en recommandation par exemple) afin de choisir son programme. Parmi les domaines applicatifs possibles pour notre modèle, nous pouvons notamment citer : la musique qui est un domaine dans lequel le contexte de l'utilisateur est prépondérant ; le e-commerce dans lequel l'analyse de l'évolution de la diversité nous permettrait de savoir si l'utilisateur cherche quelque chose de précis, cherche à se faire une idée d'une gamme de produits ou encore souhaite trouver un cadeau ; les sites d'actualités ; les sites d'offres d'emploi ; l'éducation ; les items à faible temps de consultation tels que les blagues⁴³ . . .

Nous avons choisi d'utiliser des corpus issus d'applications musicales, et ce pour plusieurs raisons. Premièrement, les consultations de musiques possèdent les prérequis mentionnés précédemment et offrent en plus d'autres avantages. Par exemple, une musique possède un temps de consultation qui lui est propre, c'est-à-dire qu'un utilisateur ne peut pas consulter plus ou moins rapidement une musique qu'un autre utilisateur (contrairement à une ressource textuelle par exemple qui peut être lue plus rapidement ou moins rapidement). Un utilisateur qui passe à la

40. <https://labrosa.ee.columbia.edu/millionsong/>

41. <https://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

42. <https://grouplens.org/datasets/movielens/>

43. <http://eigentaste.berkeley.edu/dataset/>

musique suivante nous permet de savoir que cette musique n'est pas souhaitée, quelle qu'en soit la raison [Jiang and He, 2016]. Deuxièmement, écouter de la musique est une activité fortement soumise à l'influence du contexte [Schedl et al., 2015]. Étant donné que nous souhaitons développer une approche permettant de détecter le contexte sur la base de l'évolution de la diversité, se situer dans un cadre où le contexte est fortement présent est indispensable. Troisièmement, sachant que notre modèle exploite les métadonnées des ressources consultées pour le calcul de la diversité, trouver des corpus de données pour lesquels ces dernières étaient facilement accessibles était primordial. Or, il existe certains sites spécialisés dans l'extraction des métadonnées musicales qui proposent un accès à leur base de données via leurs API^{44 45}, nous permettant ainsi d'obtenir tout le matériel nécessaire afin de tester nos hypothèses.

Nous avons utilisé deux corpus que nous allons présenter dans les sections suivantes : un corpus obtenu via le site Last.fm et un corpus obtenu suite à une collaboration avec Deezer.

2.1 Corpus Lastfm

Le premier corpus que nous avons utilisé dans nos travaux est issu de la plateforme de *scrobbling* de Lastfm⁴⁶. Le principe du *scrobbling* est d'envoyer de manière automatique à un serveur les musiques écoutées sur l'ordinateur d'un client via un logiciel installé à cet effet⁴⁷. Cela permet d'une part, d'alimenter la base de données du site et d'autre part, d'exploiter les données obtenues afin de calculer des recommandations pour les utilisateurs qu'ils pourront retrouver sur le site en se connectant à leurs comptes.

Les données que nous avons récoltées contiennent 100 utilisateurs pour lesquels nous possédons l'ensemble des consultations effectuées entre juin 2005 et décembre 2013 en utilisant la méthode `user.getRecentTracks()` de l'API. Les statistiques concernant le corpus sont présentées dans les Tableaux 2.1 et 2.2 ci-dessous :

TABLE 2.1 – Statistiques du corpus Lastfm

# Utilisateurs (U)	99
# Consultations (C)	1 980 587
# Artistes	17 803
# Musiques	168 084
C/U	20 006

TABLE 2.2 – Statistiques d'écoutes pour les utilisateurs du corpus Lastfm⁴⁸

	Min	Max	\bar{x}	σ
# Consultations (C)	10 002	71 289	20 005,92	12 919,34
# Artistes (A)	205	2 105	836,08	368,29
# Musiques (M)	708	11 549	3 347,37	1 888,18
C/A	7,81	95,71	26,38	15,07
C/M	1,93	21,136	6,91	3,95

44. <http://the.echonest.com/>

45. <https://developer.spotify.com/web-api/console/>

46. <https://www.last.fm/>

47. <https://www.last.fm/fr/about/trackmymusic>

Ce corpus contient un faible nombre d'utilisateurs pour lesquels nous possédons cependant de nombreuses écoutes. Les ratios C/M ou C/A nous donnent une indication quant à la diversification des utilisateurs en termes d'artistes ou de musiques. Plus ce rapport est grand, plus un utilisateur a tendance à écouter des musiques ou des artistes qu'il a déjà écoutés auparavant. Par exemple, un ratio C/M de 1,88 signifie qu'un utilisateur a écouté en moyenne 1,88 fois la même musique. En moyenne, les utilisateurs ont écouté 20 006 musiques. De plus, nous pouvons constater que ces utilisateurs ont des comportements relativement différents en termes d'habitudes d'écoutes si l'on regarde les ratios C/A et C/M .

2.2 Corpus Deezer

Au cours de l'année 2017, la société Deezer⁴⁹ avec laquelle nous avons établi une collaboration nous a fourni des données anonymisées issues de leur plateforme. Le corpus que nous avons obtenu contient plus de 2 000 utilisateurs qui ont consulté des musiques entre le 4 décembre 2016 et le 5 janvier 2017. Les statistiques concernant ce corpus sont présentées dans les Tableaux 2.3 et 2.4 ci-dessous :

TABLE 2.3 – Statistiques du corpus Deezer

#Utilisateurs (U)	2 903
#Consultations (C)	1 871 919
#Artistes	35 326
#Musiques	204 434
C/U	644

TABLE 2.4 – Statistiques d'écoutes pour les utilisateurs du corpus Deezer⁵⁰

	Min	Max	\bar{x}	σ
#Consultations (C)	50	10 420	639,52	792,89
#Artistes(A)	1	2 119	140,75	145,34
#Musiques (M)	4	4 999	276,61	280,76
C/A	1,05	599	7,31	21,44
C/M	1,0	79,62	2,53	2,72

2.3 Attributs collectés

Les deux corpus ne contenaient à l'origine pas d'information sur les musiques et les artistes. Les seules informations dont nous disposions étaient pour chaque consultation des utilisateurs, le pseudo de l'utilisateur, le nom de la musique, le nom de l'artiste ainsi que le *timestamp*⁵¹. Afin d'obtenir les attributs indispensables à notre approche, nous avons utilisé à l'origine de nos travaux les services du site theEchoNest⁵². Le site proposait à l'origine une API permettant de récupérer des métadonnées sur des musiques et des artistes. Grâce à ce dernier, nous avons pu collecter 13 attributs (7 pour les musiques et 6 pour les artistes) que nous avons utilisés

49. <https://www.deezer.com/>

51. Le timestamp représente le nombre de secondes écoulées depuis le 1er janvier 1970 (début de l'heure UNIX) (<http://glossaire.infowebmaster.fr/timestamp/>)

52. <http://the.echonest.com/>

dans l'ensemble des travaux publiés hors du cadre de ce manuscrit. Suite au rachat de theEchonest par Spotify en 2015, nous avons dû mettre à jour les données du corpus de Lastfm via l'API de Spotify⁵³. Les métadonnées retournées par Spotify, bien que basées sur les données de theEchonest, contiennent cependant quelques différences que nous avons décidé de supprimer en utilisant le même service pour les deux corpus. De ce fait, il existe quelques différences entre les résultats présentés dans ce manuscrit et les résultats présentés dans les publications que nous avons réalisées dans le cadre de cette thèse. Ces différences sont dues au fait que les attributs collectés entre les deux services ne sont pas les mêmes, bien qu'il existe des attributs communs et nous avons également constaté que certaines valeurs des attributs ont été mises à jour (pour des musiques et des attributs identiques, les valeurs récupérées par Spotify peuvent varier par rapport aux valeurs obtenues via theEchonest). Nous précisons que ces différences ne sont pas significatives pour les attributs communs entre les deux services et que l'utilisation de l'un ou l'autre des services ne change en rien les conclusions de nos travaux.

En utilisant l'API de Spotify⁵⁴, nous avons donc pu récupérer 16 attributs au total⁵⁵. En plus de définir ces attributs, nous précisons également quel est le type de chacun des attributs parmi les types suivants : valeur numérique (VN), valeur qualitative⁵⁶ (VQ) et liste de mots (LM). Les 13 premiers attributs portent sur les musiques :

- **Acousticness (VN)** : mesure du niveau “acoustique” d’une musique. Cette mesure est comprise entre 0 et 1, et indique à quel point une musique est acoustique, c’est-à-dire si elle emploie ou non une ou des sonorités électroniques (plus cette mesure est élevée, moins la musique contient de sonorités électroniques). Par exemple, *Oxygène* de JEAN-MICHEL JARRE possède une *acousticness* de 0,0234 et *Le vol du bourdon* de RIMSKY-KORSAKOV possède une *acousticness* de 0,966.
- **Durée (VN)** : durée de la musique, en millisecondes.
- **Danceability (VN)** : mesure du niveau indiquant si une musique se prête ou non à la pratique de la danse. Plus cette mesure est élevée, plus il est possible de danser “dessus”. Par exemple, l'*Adagio en sol Mineur* de TOMASO ALBINONI possède une *danceability* de 0,0885 alors que la chanson *Rasputin* de BONEY.M possède une *danceability* de 0,691.
- **Énergie (VN)** : mesure du niveau d’intensité et d’activité d’une musique. Cette mesure est comprise entre 0 et 1. Plus cette mesure est élevée, plus cette musique est jugée “énergique”, c’est-à-dire “intense”, “rapide” et “bruyante”. Par exemple, *Please don't touch* de MOTÖRHEAD possède une énergie de 0,998 alors que *Adagio en sol Mineur* de TOMASO ALBINONI possède une énergie de 0,124.
- **Instrumentalness (VN)** : mesure indiquant la présence ou non de parties vocales dans la musique. Plus cette mesure est proche de 1 et moins il est probable qu’il y ait des parties vocales dans la musique. Par exemple, *Adagio en sol Mineur* de TOMASO ALBINONI a une *instrumentalness* de 0,84 alors que *Bohemian Rhapsody* de QUEEN possède une *instrumentalness* de 0.
- **Clé (VQ)** : représente la tonalité d’une musique en notation internationale. Par exemple B, C, C_b/C_#, D... Une musique ne possède qu’une seule clé(*key*) parmi les 12 existantes sur l’API.

53. <https://github.com/beetbox/beets/issues/1920>

54. <https://developer.spotify.com/web-api/get-audio-features/>

55. Certains attributs ne peuvent pas être traduits littéralement en français, nous avons donc gardé dans ce cas les noms originaux en anglais.

56. Les attributs de type binaire appartiennent à cette catégorie d’attribut.

- **Liveness (VN)** : mesure indiquant la présence ou non d'un public lors de l'enregistrement, ce qui revient à préciser si une musique est enregistrée en studio ou en concert. Cette valeur est comprise entre 0 et 1. Plus la valeur est élevée, plus il est probable que cette musique ait été enregistrée en concert.
- **Loudness (VN)** : mesure la valeur moyenne de l'intensité des basses dans la musique. Cette mesure est exprimée en décibels (dB) et la plupart des musiques possèdent une mesure entre -60dB et 0db. Plus la valeur absolue de cette mesure est faible, plus il y a de basses dans la musique. Par exemple, *Piano Sonata in A major, D. 959 Fourth Movement (Rondo)* de FRANCK SCHUBERT possède une *loudness* de -44,481 alors que *Get on Top* des RED HOT CHILI PEPPERS possède une *loudness* de -0,407.
- **Mode (VQ)** : Indice binaire qui indique si le mode d'une musique est mineur (0) ou majeur(1). Par exemple, *Adagio en sol Mineur* de TOMASO ALBINONI possède un mode de 0 et *The Scientist* de COLDPLAY possède un mode de 1. Le mode ne doit pas être confondu avec la valence de la musique.
- **Speechiness (VN)** : mesure représentant l'importance des paroles dans la musique. Cette mesure est comprise entre 0 et 1. Plus la *speechiness* est grande est plus les paroles sont importantes. Par exemple, les poèmes ont une *speechiness* proche de 1 alors que les musiques sans paroles ont une *speechiness* proche de 0. La musique *Oxygène* de JEAN-MICHEL JARRE possède une *speechiness* de 0,0365 et *Il n'y a pas d'amour heureux* de LOUIS ARAGON possède une *speechiness* de 0,95.
- **Tempo (VN)** : tempo de la musique, c'est-à-dire sa vitesse d'exécution exprimée en nombre de battements par minute (BPM). Le tempo est généralement compris entre 40 et 208 BPM. Par exemple, *Modern Love* de DAVID BOWIE possède un tempo de 181,329 alors que *What a wonderful world* de LOUIS ARMSTRONG possède un tempo de 77,591.
- **Signature rythmique (VQ)** : cette valeur représente le chiffrage de mesure de la partition musicale. Par exemple, le chiffrage de mesure peut être 4/4 ou 2/4 ce qui correspond dans le premier cas à 4 noires par mesure et à 2 noires par mesure dans le second cas.
- **Valence (VN)** : mesure représentant l'émotion de la musique. Cette valeur est comprise entre 0 et 1. Plus cette mesure est grande, plus l'émotion dégagée par la musique est positive. Par exemple, *The Scientist* de COLDPLAY a une valence de 0,212 et *I'm still standing* d' ELTON JOHN a une valence de 0,772.

Les 3 derniers attributs sont relatifs aux artistes :

- **Popularité (VN)** : mesure représentant la popularité d'un artiste. Cette mesure est comprise entre 0 et 100. Par exemple, THOMAS ALBINONI possède une popularité de 51 et ED SHEERAN possède une popularité de 95.
- **Artistes similaires (LM)** : liste des 20 artistes les plus similaires. Par exemple, les artistes similaires au groupe AC/DC sont KISS, Deep Purple, ZZ Top, Black Sabbath, Aerosmith, Van Halen, Alice Cooper. . .
- **Genres (LM)** : liste des genres associés aux artistes. Par exemple, les genres associés au groupe DAFT PUNK sont *electro*, *electronic* et *filter house*.

Les statistiques correspondant aux attributs que nous avons pu récupérer sont résumées dans le Tableau 2.5 ci-dessous. Le Tableau 2.6 représente quant à lui la couverture en termes d'attributs récupérés.

TABLE 2.5 – Statistiques des valeurs attributs pour les deux corpus

Corpus		Lastfm				Deezer			
Attributs		Min	Max	\bar{x}	σ	Min	Max	\bar{x}	σ
Musiques	Acousticness	0,0	0,996	0,19206	0,2836	0,0	0,99599	0,3431	0,3379
	Danceability	0,0	0,985	0,4990	0,1787	0,050	0,9900	0,5729	0,1799
	Durée	1 486	4 143 930	231 552	99 291	15 000	5 947 530	240 771	104 491
	Énergie	0,0	1,0	0,71736	0,2385	0,00002	1,0	0,5977	0,2529
	Instrumen,	0,0	0,9980	0,15566	0,2938	0,0	1,0	0,1597	0,3062
	Liveness	0,0	1,0	0,2288	0,1911	0,00922	1,0	0,19828	0,17584
	Loudness	-44,6580	4,9229	-7,5704	4,1560	-47,8260	5,51399	-9,0607	5,0741
	Speechiness	0,0	0,9660	0,1092	0,1310	0,02160	0,9649	0,08743	0,0999
	Tempo	0,0	238,895	122,8547	30,8620	30,01	239,91	119,21	28,885
	Valence	0,0	0,99800	0,4607	0,2449	0,0	1,0	0,5017	0,2659
Artistes	Art. sim	0,0	20	19,97	0,6183	0,0	20	19,940	0,8316
	Genres	0,0	34,0	4,543	3,85	0,0	41,0	4,298	4,153
	Pop.	0,0	100	36,6528	20,6267	0,0	100	36,1498	19,7589

TABLE 2.6 – Statistiques de couverture des attributs pour les deux corpus

Couverture	Lastfm	Deezer
Musiques retrouvées	67,95 %	87,58 %
Artistes retrouvés	72,75 %	82,53 %
Couverture consultations	81,75 %	89,48 %
Couverture brute	71,63 %	88,13 %

La couverture brute correspond à la quantité d'informations que nous possédons, c'est-à-dire au nombre d'attributs par rapport au nombre d'attributs maximum que nous pourrions obtenir. En effet, ce n'est pas parce qu'une correspondance avec une musique a pu être établie sur l'API de Spotify que l'ensemble des attributs a été retourné (seule une partie des attributs peut être retournée).

2.4 Limites

Même si nous possédons de nombreuses informations sur les musiques de ces corpus, ces dernières peuvent souffrir de certaines limites. En ce qui concerne le corpus de Lastfm, son principal défaut provient du fait que les musiques sont issues du *Scrobbling*. Le principe consistant à récupérer les noms des musiques écoutées par les utilisateurs sur leur ordinateur ou au travers du lecteur Youtube⁵⁷ via un *plugin*, la plateforme Lastfm ne possède pas de contrôle sur les noms d'artistes et de musiques qui lui sont retournés. De fait, l'orthographe des musiques et artistes que nous possédons ainsi que leur authenticité n'est pas toujours fiable, et ce, même si la plateforme possède un service de correction automatique des titres des musiques. Si un utilisateur a écouté une musique possédant le nom d'une autre musique, voire même d'un artiste différent, aucune correction n'est possible. Certaines données aberrantes ont été trouvées dans les données

57. <https://www.youtube.com/>

récupérées via la méthode `getRecentTrack()` que nous avons utilisée pour obtenir le corpus. Par exemple, nous avons trouvé des noms de chansons tels que :

- PisteAudio 05
- 017-Audioslave - Be Yourself
- Ny Batteri (Hyper Line To Streaming Audio)
- Sign Language For Jizz Everywhere (This bit not ideal for audio)

Outre ces exemples, un manque d'homogénéité dans l'annotation des musiques peut également bruyé le corpus. Nous avons par exemple constaté que certaines musiques qui n'étaient pas retrouvées par l'API de Spotify contenaient des informations supplémentaires. Voici quelques-unes des expressions indésirables que nous avons pu trouver (la liste n'est pas exhaustive) :

- | | |
|----------------------------|------------------------------|
| — -Album Version (Edited) | — - Amended Album Version |
| — (Album Version (Edited)) | — (Album Version) |
| — Album Version (Edited) | — Album Version |
| — Album Version Edited | — [Album Version] |
| — (Explicit Album Version) | — Remastered Album Version |
| — - Explicit Album Version | — (Album Version (Explicit)) |
| — Explicit Album Version | — - Non-Album Track |
| — (explicit album version) | |

Même si certaines d'entre elles peuvent être retrouvées et supprimées à l'aide par exemple d'expressions régulières, le fait qu'elles ne soient pas toujours présentes au même endroit dans le titre de la musique rend le processus de nettoyage complexe. D'autres expressions indésirables peuvent également bruyé le corpus et rendre impossible la reconnaissance par l'API de Spotify. Nous pouvons notamment noter :

- (Live - Acoustic)
- (iTunes Bonus Track)
- (VIDEO EDIT)
- (Official Audio)
- (High Quality)

La seconde limite concerne la gestion automatique des noms des artistes, et ce, même s'ils sont correctement orthographiés.

- B-Boy Document 99 (Featuring Mos Def & Mad Skillz) *The High & Mighty*
- Eklypse (tEho & Tran Remix) *Deepfunk*
- Blood Pt 2 *Buck 65 Remix (featuring Sufjan Stevens and Serengeti)*

Les trois exemples ci-dessus permettent d'illustrer cette complexité. Le titre de la musique est en gras et le nom de l'artiste associé est en italique. Ces informations sont telles que nous les obtenons en lisant les données brutes obtenues par Lastfm. Dans la première musique, il s'agit d'une coopération de 3 artistes qui sont "Mos Def", "Mad Skillz" et l'artiste principal de cette musique, The High & Mighty. Alors qu'il faudrait normalement séparer les deux artistes qui figurent en tant que coauteurs, il nous est impossible de le faire de manière automatique. En effet, The High & Mighty forment un seul et même artiste, tout comme tEho & Tran dans la

seconde musique. De ce fait, si l'on sépare à chaque fois les artistes selon le symbole "&", nous nous exposons dans certains cas à fausser les données en créant des artistes qui n'existent pas.

Conclusion : il existe et il existera toujours des données erronées, incomplètes, voire frauduleuses dans certains cas, dans les corpus de données que nous sommes amenés (en tant que chercheurs) à constituer/récupérer et utiliser. Nous avons souhaité présenter les limites de nos corpus, car nous pensons qu'il est important de porter un regard critique et objectif sur les données que nous utilisons. Nous n'entendons néanmoins pas critiquer les données à chaque résultat indésirable. Nous pensons qu'il est important de savoir quels sont les facteurs qui peuvent influencer sur les résultats afin de pouvoir proposer les conclusions les plus objectives possible. Actuellement, il existe de nombreuses études qui s'intéressent à la reproductibilité des résultats [Jannach and Adomavicius, 2016, Çoba and Zanker, 2017, Said and Bellogín, 2015] et une présentation claire du corpus utilisé représente alors un pas en avant vis-à-vis de la transparence des résultats obtenus.

Nous avons décidé d'utiliser deux corpus afin de confirmer l'hypothèse selon laquelle notre approche peut être appliquée indépendamment du corpus de données. De plus, nous pourrions également comparer par la suite les performances obtenues étant donné que ces deux corpus sont différents en termes de caractéristiques. Le corpus de Lastfm comporte des consultations de peu d'utilisateurs sur une longue période temporelle alors que le corpus de Deezer comporte des consultations de nombreux utilisateurs sur une période temporelle plus faible.

Pour résumer : Les corpus que nous possédons sont différents tant par leurs origines que leurs caractéristiques. Le corpus de Lastfm contient un faible nombre d'utilisateurs ayant effectué de nombreuses consultations sur plusieurs années et sur différentes plateformes. Le corpus de Deezer contient de nombreux utilisateurs pour lesquels nous connaissons l'ensemble des consultations effectuées sur une période d'un mois et sur une unique plateforme. Nous possédons également 16 attributs de différentes natures que nous avons collectés à l'aide d'un service spécialisé, et ce, pour la majorité des musiques et des artistes.

Chapitre 3

Modéliser l'évolution temporelle de la diversité

Sommaire

3.1 Principe	55
3.2 Formalisme	57
3.3 Résultats de la modélisation temporelle de la diversité	62
3.3.1 Observations sur l'évolution temporelle de la diversité	62
3.3.2 Influences des paramètres sur la modélisation	65
3.3.3 Adaptation aux items de natures différentes	71
3.4 Discussion	71

Comme nous l'avons vu dans l'état de l'art présenté dans le Chapitre 1, et notamment à travers les travaux de [Castagnos et al., 2010], le besoin en diversité des utilisateurs n'est pas une dimension statique. Cette dimension évolue en fonction du contexte des utilisateurs, en fonction de l'intention d'achat. Par exemple, proposer un niveau de diversité adapté dans les recommandations améliore l'utilité de ces dernières. Malgré les conclusions apportées par ces travaux concernant la prise en compte du besoin en diversité, nous n'avons pas trouvé d'approche dans la littérature proposant de modéliser l'évolution en diversité au cours du temps à l'échelle des consultations des utilisateurs.

Dans ce chapitre, nous présentons le modèle que nous avons développé afin de répondre à cette problématique. Nous illustrons tout d'abord le principe de notre modèle avant de présenter son formalisme. Les résultats de modélisation obtenus sur les différents corpus que nous avons introduits dans le Chapitre 2 sont ensuite explicités. Enfin, nous clôturons cette partie par une discussion à propos des atouts et des limites de notre modèle, mais surtout des perspectives offertes que nous mettrons en application dans les chapitres suivants.

3.1 Principe

Calcul de la diversité

La diversité telle que nous l'utilisons dans la suite de ce manuscrit correspond à la définition de [Smyth and McClave, 2001] que nous avons présenté dans la Section 1.3.1 p. 37. Cette diversité est définie comme étant l'inverse de la similarité. Ainsi, connaissant une valeur de similarité, nous pouvons calculer la diversité correspondante. Nous entendons calculer cette similarité sur la base de l'ensemble des attributs des items indépendamment, mais également calculer la moyenne

des valeurs de similarité. Ne pas nous contenter de conserver uniquement la moyenne de la similarité nous offre l'avantage d'obtenir une mesure de similarité possédant autant de dimensions que les items comportent d'attributs. Ainsi, si les items possèdent x attributs, la diversité entre deux items pourra être représentée sur au moins x dimensions. Alors que la moyenne de l'ensemble des valeurs de diversité pourra servir d'indicateur général de la variation de diversité, les diversités de chaque attribut permettront une analyse détaillée de l'évolution de la diversité.

Évolution temporelle

Le modèle que nous avons conçu s'inspire de certains algorithmes de traitement du langage naturel [Bonnin et al., 2012] dans lesquels il est question de représenter un problème sous la forme d'un N-gramme ou encore selon une représentation Markovienne d'ordre n . Cependant, dans notre cas, nous n'entendons pas calculer des probabilités d'apparition du prochain item mais utiliser le principe d'une fenêtre glissante afin de calculer la diversité apportée par un item par rapport à un historique de petite taille constitué par les items consultés précédemment. La Figure 3.1 illustre le principe de notre modèle.

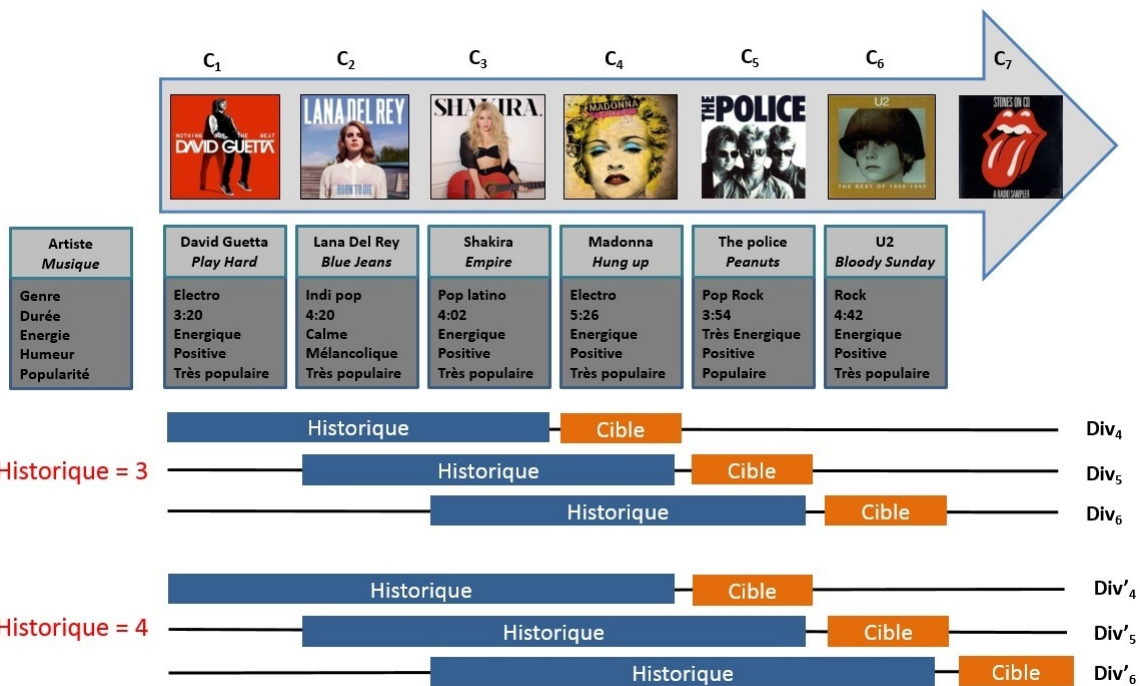


FIGURE 3.1 – Illustration du fonctionnement de notre modèle

Sur la Figure 3.1 nous appelons l'item courant **cible** et les n consultations précédentes impliquées dans le calcul de la diversité **historique**. Ainsi, à chaque nouvelle consultation d'un utilisateur, la cible et l'historique changent et une nouvelle valeur de diversité est alors calculée. Dans cet exemple, la diversité est calculée sur la base de 5 attributs qui sont le genre, la durée, l'énergie, l'ambiance et la popularité des musiques. Nous avons également représenté deux situations qui correspondent à des historiques de tailles différentes, un historique de taille 3 et un historique de taille 4.

3.2 Formalisme

Afin de calculer l'évolution de la diversité au cours du temps, nous avons défini un formalisme générique qui peut être utilisé sur n'importe quel corpus de données. La seule contrainte à respecter étant que les items doivent posséder des attributs à partir desquels une mesure de similarité (et donc de diversité) peut être calculée et qu'ils soient consultés en séquence.

Les notations que nous avons utilisées sont les suivantes :

$I = \{i_1, i_2, \dots, i_m\}$ est l'ensemble des items qui sont consultés par les utilisateurs.

$U = \{u_1, u_2, \dots, u_n\}$ est la liste des utilisateurs et u représente l'utilisateur courant.

$T = \{t_{-n}, \dots, t_{-1}, t_0\}$ est à la liste des pas de temps. t_0 représente le temps courant, t_{-1} le pas de temps précédent, *etc.*

$C = \{c_1, c_2, \dots, c_z\}$ est l'ensemble des consultations. Une consultation correspond à un item $i \in I$ ayant été consulté à un pas de temps donné $t \in T$ par un utilisateur $u \in U$.

$C_{k,t}^u$ représente l'historique récent de taille k au pas de temps $t \in T$ de l'utilisateur $u \in U$. Cet historique peut également être représenté sous la forme d'une séquence de consultations $\langle c_{t-k}^u, \dots, c_{t-2}^u, c_{t-1}^u, c_t^u \rangle$.

$A_i = \{a_1, a_2, \dots, a_h\}$ représente la liste des attributs d'un item i .

$i_1.a_2$ représente donc l'attribut 2 de l'item 1.

Étant donné que l'on souhaite connaître la diversité apportée par un item (la cible) relativement à un ensemble d'items (l'historique), nous avons adapté la formule de diversité relative de [Smyth and McClave, 2001] présentée dans le Chapitre 1 dont nous rappelons l'Équation 3.1. À l'origine, la diversité relative était calculée par rapport à la taille de la classe d'items, correspondant dans notre cas à l'historique. Cependant, étant donné qu'il est possible que certaines valeurs de diversité ne puissent pas être calculées à cause de données manquantes, nous avons adapté la formule. Nous avons ajouté dans l'Équation 3.2 un paramètre s qui correspond au nombre de fois où il est possible de déterminer une valeur de diversité entre la cible et les items de l'historique. Par exemple, dans le cas où l'on calcule la $DR(c_4^u, C_{2,3}^u)$, il est possible qu'aucune information ne soit connue sur l'item de la consultation c_3 . Dans ce cas, seules les valeurs de diversité entre c_4 & c_1 et c_4 & c_2 interviendront dans le calcul de la DR et s aura une valeur de 2. Nous appliquons ce même principe à l'échelle de chacun des attributs si une ou plusieurs valeurs d'attributs ne sont pas connues. Cette formule de calcul de la similarité basée sur les attributs prenant en compte les attributs manquants est représentée par l'Équation 3.3.

$$DR(item, C) = \begin{cases} 0 & \text{si } C = \emptyset, \\ \frac{\sum_{j=1..z} (1 - \text{similarité}(i, c_j))}{z} & \text{sinon} \end{cases} \quad (3.1)$$

$$DR(c_t^u, C_{k,t}^u) = \begin{cases} \text{NaN} & \text{si } C_{k,t}^u = \emptyset \text{ ou si } s = 0, \\ \frac{\sum_{j=1..k} (1 - \text{similarité}(c_t^u, c_{t-j}^u))}{s} & \text{sinon} \end{cases} \quad (3.2)$$

$$\text{Similarité}(c_t^u, c_{t-j}^u) = \begin{cases} \text{NaN} & \text{si } (A_{c_t^u} \cap A_{c_{t-j}^u}) \text{ ou } A_{c_t^u} \text{ ou } A_{c_{t-j}^u} = \emptyset, \\ \frac{\sum_{a \in A_{c_t^u} \cap A_{c_{t-j}^u}} (\beta_a * \text{sim}_a(c_t^u, c_{t-j}^u))}{\sum_{a \in A_{c_t^u} \cap A_{c_{t-j}^u}} \beta_a} & \text{sinon} \end{cases} \quad (3.3)$$

Dans l'Équation 3.2, nous avons utilisé le terme de NaN (acronyme de *Not a Number*) lorsque :

- les items ne contiennent pas d'attribut en commun ;

— la valeur de l'attribut n'est pas connue.

L'utilisation de NaN est justifiée par le fait que nous ne souhaitons pas attribuer une valeur numérique à la similarité lorsque cette dernière ne peut pas être calculée. Si nous avons choisi de mettre 0 comme valeur de similarité lorsqu'il n'est pas possible de calculer la similarité, cela se traduirait alors par une diversité de 1. Or, une diversité de 1 signifie que les items sont complètement divers. Attribuer la valeur NaN permet de ne pas faire intervenir cette valeur dans le calcul de la DR.

Lorsque nous avons présenté les différents jeux de données que nous utilisons dans nos travaux (cf. Chapitre 2), nous avons vu que les attributs des items peuvent être de natures différentes. Nous allons à présent introduire les formules que nous avons utilisées pour chacun des types d'attributs que nous possédons. Nous rappelons que les attributs des corpus que nous utilisons dans ce manuscrit⁵⁸ peuvent être : des valeurs numériques, des valeurs qualitatives et des listes de mots.

Listes

Les attributs que nous possédons sous forme de liste sont : les genres et les artistes similaires associés à des artistes. La formule associée aux attributs sous forme de listes est la suivante :

$$sim_a(i_1, i_2) = \frac{card(i_1.a \cap i_2.a)}{min(card(i_1.a), card(i_2.a))} \quad (3.4)$$

Cette formule permet de calculer la similarité qui existe entre deux listes (représentées par l'attribut a). Cette formule est une adaptation de l'indice de Jaccard qui se calcule comme suit :

$$sim_a(i_1, i_2) = \frac{card(i_1.a \cap i_2.a)}{card(i_1.a \cup i_2.a)} \quad (3.5)$$

Nous avons choisi d'adapter la formule de l'indice de Jaccard afin d'être plus permissifs sur les similarités qui sont calculées à partir de listes de mots. Les attributs sous forme de listes de nos corpus sont le genre d'un artiste et les artistes similaires. L'API que nous avons utilisé nous a retourné un total de 1 172 genres différents. De ce fait, il existe de nombreuses variantes d'un même genre. Par exemple le terme rock contient plus de 123 déclinaisons (alternative rock, modern rock, rock, blues-rock, classic rock, garage rock, indie rock, canadian rock, pop rock. . .). En ce qui concerne les artistes similaires, un artiste possède en moyenne une liste de 20 artistes qui lui sont similaires. En utilisant alors le minimum du cardinal des deux listes de genres des items qui interviennent dans la formule 3.4 plutôt que l'union, la similarité retournée sera plus grande pour une même intersection qu'en utilisant l'indice de Jaccard. Nous avons proposé cette adaptation en raison des caractéristiques des corpus que nous avons utilisés. Il est évidemment possible d'utiliser l'indice de Jaccard ou d'autres formules (utilisant par exemple les relations lexicales et sémantiques pour les genres ou en regroupant les genres proches sous un même terme [Porter, 1997]) suivant le domaine applicatif ou encore les jeux de données utilisés. Le choix de certaines formules peut être motivé par l'idée d'optimiser la répartition des valeurs de diversité entre les différents attributs. Et ce, afin d'éviter par exemple que les formules de certains attributs soient très "strictes" alors que les formules d'autres attributs sont plus "souples".

Valeurs qualitatives

58. Il existe d'autres types d'attributs tels que les coordonnées géographiques et les intervalles que nous avons utilisés pour d'autres corpus (cf. Section 3.2, p. 60).

Les attributs de type qualitatif sont que nous possédons : la clé, le mode et la signature rythmique. La formule associée aux attributs qualitatifs est la suivante :

$$sim_a(i_1, i_2) = \begin{cases} 1 & \text{si } i_2.a = i_1.a, \\ 0 & \text{sinon.} \end{cases} \quad (3.6)$$

La similarité entre deux attributs qualitatifs est dans notre cas identique à la similarité de deux attributs binaires, c'est-à-dire que deux attributs sont soit exactement similaires soit pas du tout similaires. Dans notre cas, les attributs concernés sont le mode, la tonalité et la signature rythmique et il semble effectivement très complexe, voire impossible, de définir des niveaux intermédiaires de similarité.

Valeurs numériques Les attributs de type numérique que nous possédons sont : l'*acousticness*, la durée, la *danceability*, l'énergie, l'*instrumentalness* la *liveness*, la *loudness*, la *speechiness*, le tempo, la valence et la popularité de l'artiste.

$$sim_a(i_1, i_2) = e^{-\alpha * \left(\frac{i_1.a - i_2.a}{max_a - min_a} \right)^2} \quad (3.7)$$

Le choix de cette formule de similarité est motivé par le fait que nous voulions obtenir une fonction nous permettant de gérer l'impact en termes de similarité pour une même différence de valeurs des attributs. Autrement dit, le coefficient α intervenant dans cette formule détermine si un même écart entre deux valeurs d'attribut sera plus ou moins discriminant dans le calcul de la similarité. Les graphiques de la Figure 3.2 ci-dessous représentent l'impact du coefficient α dans le calcul de la similarité entre deux attributs. Dans ces graphiques, l'axe des abscisses représente la différence entre deux valeurs numériques et l'ordonnée représente la similarité correspondante. Nous pouvons constater que plus le coefficient α est important, plus des écarts identiques entre les deux attributs diminueront la valeur de la similarité calculée. **L'utilisation d'un coefficient élevé permet alors de s'assurer que deux attributs sont jugés similaires s'ils ont des valeurs très proches.**

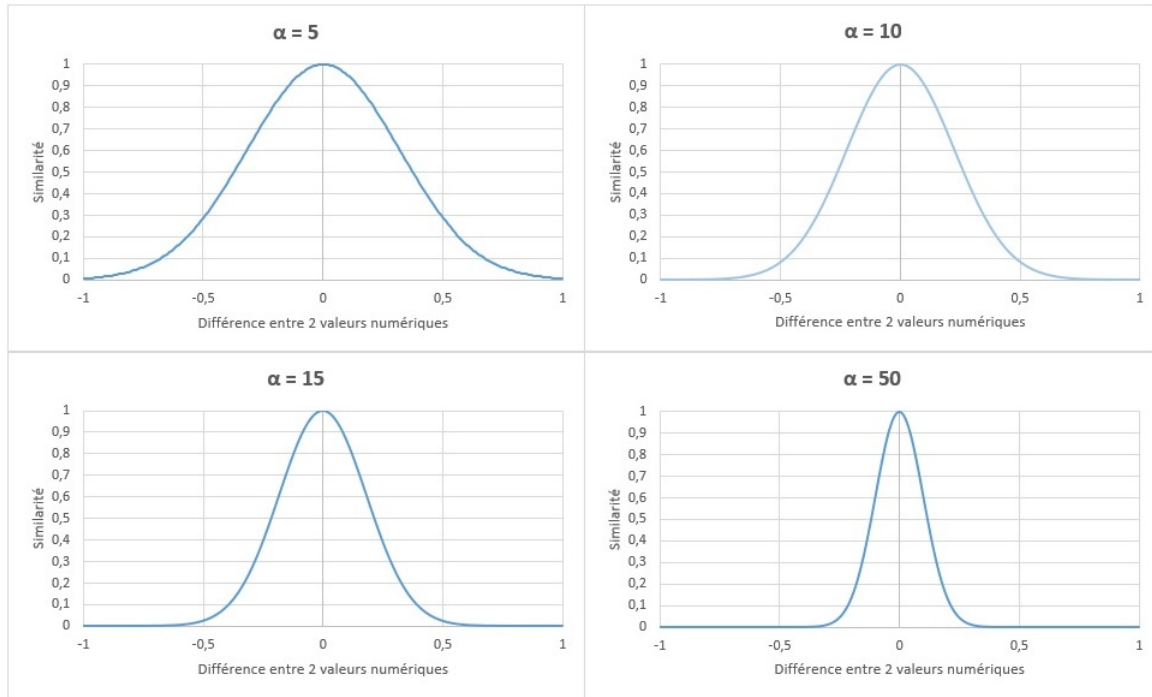


FIGURE 3.2 – Impact du coefficient sur la formule de similarité

Outre la représentation graphique ci-dessus, nous pouvons également observer dans le Tableau 3.1 ci-dessous l'impact que possède le coefficient α sur la valeur numérique de la similarité.

TABLE 3.1 – Similarité en fonction du coefficient α et de l'écart des valeurs des attributs

$\alpha \backslash$ Écart	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
5	1	0,951	0,819	0,638	0,449	0,287	0,165	0,086	0,041	0,017	0,000
10	1	0,905	0,670	0,407	0,202	0,082	0,027	0,007	0,002	0	0
15	1	0,861	0,549	0,259	0,091	0,024	0,005	0,001	0	0	0
50	1	0,607	0,135	0,011	0	0	0	0	0	0	0

Nous pouvons constater que la diversité reste supérieure à 0,5 pour un coefficient α de 5 lorsque la différence entre deux valeurs du même attribut est inférieure à 0,4. Inversement, avec un coefficient α de 50, la similarité est inférieure à 0,5 dès lors que la différence est supérieure à 0,1.

Si nous ne considérons que les attributs numériques afin de représenter un item sous la forme d'un vecteur dont chaque attribut est une dimension, nous aurions pu utiliser des mesures de l'état de l'art telles que la distance Euclidienne, de Manhattan, de Minkowski ou encore la similarité Cosinus. Cependant, en procédant de la sorte, nous ne connaissons que la diversité "globale" entre deux items mais nous ne connaissons pas la diversité existante pour chacun des attributs.

Coordonnées

Bien que nous ne l'utilisions pas dans le cadre de ce manuscrit, nous avons développé cette formule de similarité, car les coordonnées géographiques des artistes faisaient partie des attributs fournis par le site TheEchoNest. Les coordonnées étaient exprimées sous forme de deux valeurs :

la latitude et la longitude de la ville dont est originaire un artiste. Connaissant les coordonnées géographiques des artistes, nous avons pu calculer à l'aide d'une formule de géodésique (qui permet d'obtenir la distance minimale entre deux points sur une surface, *i.e.* la terre dans notre cas) la distance existante entre les coordonnées des villes de deux artistes⁵⁹. Cette distance est calculée selon l'Équation 3.8 ci-dessous où lat_a , $long_a$ et lat_b , $long_b$ représentent respectivement la latitude et la longitude d'un artiste a et d'un autre artiste b :

$$distance_{a,b} = (\arccos(\sin(lat_a) \times \sin(lat_b) + \cos(lat_a) \times \cos(lat_b) \times \cos(long_b - long_a))) \times 6378 \quad (3.8)$$

Le résultat obtenu par l'Équation 3.8 ci-dessus nous donne alors la distance en kilomètres séparant deux artistes. Dans un second temps, connaissant la distance géographique séparant deux artistes, nous pouvons calculer la similarité correspondante selon l'Équation 3.9 ci-dessous.

$$sim_a(i_1, i_2) = 1 - \frac{distance(i_1, i_2)}{max_{distance}} \quad (3.9)$$

Bien que nous ne possédions plus cet attribut dans les données récoltées par Spotify, nous soutenons que cet attribut constitue une information intéressante puisqu'il permet d'enrichir la compréhension que nous avons des habitudes d'écoute des utilisateurs (impact des facteurs culturels). Il pourrait par ailleurs être utilisé en tant que critère permettant de diversifier les musiques recommandées. De plus, l'utilisation de telles données pourrait être améliorée en se basant sur des services permettant de retrouver l'adresse correspondante à des coordonnées GPS, et donc d'obtenir les villes, régions, pays concernés afin ne plus se baser uniquement sur des distances⁶⁰. Par exemple, les États-Unis étant très vastes, deux groupes américains peuvent être séparés par une grande distance là où en Europe par exemple, cette même distance nécessiterait de traverser plusieurs pays. Dans le premier cas, les artistes appartiennent tous les deux au même pays, ce qui n'est pas le cas dans le second cas et cette information doit être prise en compte pour les recommandations. Prendre en compte uniquement la distance apparaît donc comme étant une information limitée puisque cette dernière est relative.

Intervalles

$$sim_a(i_1, i_2) = \frac{card(i_1.a \cap i_2.a)}{max(card(i_1.a), card(i_2.a))} \quad (3.10)$$

Cette formule de similarité pour des attributs contenant des intervalles n'est pas utilisée dans ce manuscrit puisqu'aucun attribut sous forme d'intervalle n'est donné par l'API de Spotify. Nous l'avons développée à l'origine afin de comparer les années d'activité des artistes fournies par le site TheEchonest. Contrairement aux coordonnées, cet attribut peut être exploité directement. Pour connaître la similarité entre deux groupes de musique par rapport à leurs années d'activité, nous comparons le nombre d'années que ces groupes ont en commun sur la plus grande période d'activité de ces deux groupes. Nous utilisons le maximum du cardinal dans ce cas, contrairement aux attributs sous forme de listes, car les périodes d'activité correspondent à des périodes très caractéristiques et qu'un groupe qui ne s'est produit qu'une seule année ne devrait pas avoir une similarité de 1 avec un groupe s'étant produit sur plusieurs dizaines d'années (dont l'année d'activité du premier groupe).

59. https://geodesie.ign.fr/contenu/fichiers/Distance_longitude_latitude.pdf

60. La principe consistant à retrouver une adresse à partir de coordonnées GPS se nomme *Reverse Geocoding* et peut être réalisé à l'aide de services tels que <https://wiki.openstreetmap.org/wiki/Nominatim>

3.3 Résultats de la modélisation temporelle de la diversité

Dans cette section, nous présentons les résultats de modélisation temporelle de la diversité obtenus à partir de notre modèle sur les 2 jeux de données musicaux. Pour des raisons évidentes de place et de lisibilité, nous ne pouvons pas présenter toutes les représentations graphiques qu'il est possible de générer. Nous avons choisi de présenter les situations les plus représentatives et les plus pertinentes pour illustrer nos propos. Non pas que les résultats que nous présentons soient des cas isolés, mais ils permettent de mettre en avant les différents phénomènes que nous avons relevés. De plus, les figures que nous présentons sont issues des deux corpus que nous avons présentés précédemment. Nous verrons également que notre modèle possède de nombreux paramètres influençant la modélisation de la diversité. Nous présentons donc ces différents paramètres, leurs caractéristiques ainsi que leurs utilités possibles.

Bien que l'objectif de cette thèse ne concerne pas directement les temps d'exécution de nos modèles, nous tenons à préciser que le calcul de la diversité pour l'ensemble des utilisateurs de chacun des deux corpus qui contiennent environ 2 000 000 de consultations s'effectue en moins d'une minute (59,89 sec.)⁶¹ dont quasiment la moitié de cette durée est nécessaire au chargement des données en mémoire. Ces temps d'exécutions montrent que la modélisation de la diversité peut être aisément réalisée en temps réel puisque cela ne représente à chaque pas de temps, qu'un seul calcul de diversité.

3.3.1 Observations sur l'évolution temporelle de la diversité

Étant donné que plusieurs paramètres interviennent dans le calcul de la diversité, nous avons décidé de fixer certaines valeurs de référence à chaque paramètre. L'impact et le rôle de chacun des paramètres seront par ailleurs présentés dans la section suivante. Les valeurs des paramètres que nous utilisons les suivantes :

- Coefficient α : 10 (équation 3.7)
- Taille d'historique : 3 (équation 3.2)
- Coefficient d'éloignement : aucun (équation 3.12)
- Poids des attributs : identiques pour tous (équation 3.3)

De manière générale, lors de chaque résultat que nous présentons, nous mentionnons les paramètres qui ont été utilisés.

La Figure 3.3 ci-dessous représente l'évolution de la diversité pour un seul attribut qui est la "Danceability" pour 50 consultations de l'utilisateur 1 du corpus de Deezer.

Sur le graphique de la Figure 3.3, nous pouvons constater qu'il existe des cas pour lesquels la DR ne possède pas de valeur. C'est le cas notamment pour les 3 premières consultations de l'utilisateur. Cela est dû au fait que l'historique de ces 3 premières consultations est inférieur à la taille de l'historique souhaité, c'est-à-dire un historique de taille 3. Comme nous l'avons vu sur la figure 3.1, la taille de la fenêtre est un paramètre qui peut être modifié. Bien qu'il soit possible de calculer des valeurs de diversité même si la taille de l'historique désiré n'est pas atteinte, nous avons fait le choix de ne calculer une valeur de DR que si l'historique contient le nombre d'items fixé par les paramètres initiaux. Nous rappelons également que, même s'il est possible d'adapter la taille de l'historique pour les premières consultations d'un utilisateur, la première consultation ne pourra quant à elle jamais posséder de valeur de diversité.

61. La configuration logicielle et matérielle est la suivante : Windows 7 64 bit, Java 8 (JDK et JRE version "1.8.0.171"), MariaDB version "5.6.36-82.2", connecteur Java Mysql : "mysql-connector-java-5.1.23-bin", processeur Intel 7700, 8 Go de mémoire dédiés à la JVM.

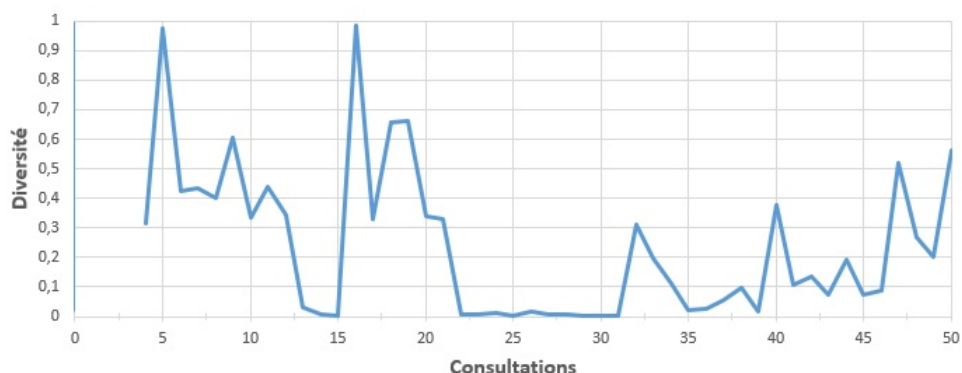


FIGURE 3.3 – Évolution de la diversité au cours du temps de l'attribut “Danceability”. Corpus de Deezer, utilisateur 1, historique de taille 3, coefficient α fixé à 10

La seconde Figure 3.4 ci-dessous représente l'évolution de la diversité pour 3 attributs qui sont l'“énergie”, la *danceability* et l'*instrumentalness* ainsi que la moyenne de ces 3 attributs. Pour des raisons de lisibilité, nous n'avons présenté que l'évolution de 3 attributs sur les 16 attributs que nous possédons. Cependant, nous précisons que dans la suite de ce manuscrit, lorsque nous parlerons de diversité, nous ferons référence à la DR calculée sur l'ensemble des attributs dont nous disposons. Dans le cas où la DR ne tiendra compte que d'un attribut ou d'un sous-ensemble d'attributs, nous prendrons soin de le préciser.

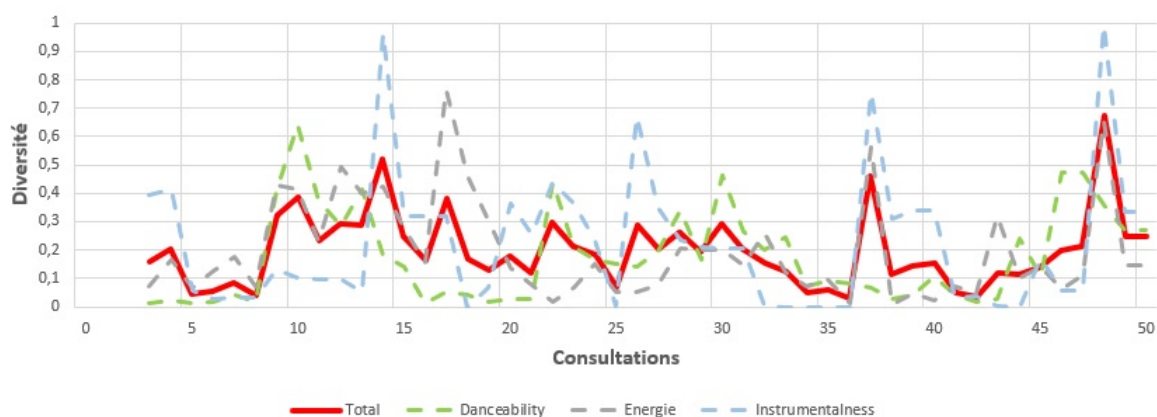


FIGURE 3.4 – Évolution de la DR pour 3 attributs et DR totale. Corpus de Lastfm, utilisateur 4, historique de taille 3, coefficient α fixé à 10

Nous pouvons constater sur les deux Figures 3.3 et 3.4 que la diversité est une valeur qui varie énormément au cours du temps. Cependant, bien que cette dernière puisse sembler erratique, nous pouvons tout de même dégager plusieurs comportements caractéristiques. Premièrement, nous pouvons observer des périodes pendant lesquelles la diversité bien que n'étant pas constante, reste relativement stable et n'évolue pas de manière importante. Deuxièmement, il existe des situations pendant lesquelles la diversité diminue fortement. Enfin, le troisième comportement concerne les fortes augmentations de diversité. Avant de présenter davantage ces trois comportements, nous avons choisi de les représenter graphiquement via la Figure 3.5. Cette dernière illustre les 3 comportements à l'échelle de 50 consultations de l'utilisateur 8 sur le corpus de

Lastfm. Les comportements que nous décrivons via ce graphique ont été observés chez la majorité des utilisateurs. Nous avons choisi cet utilisateur et ses consultations en particulier car les 3 comportements sont très démarqués ⁶².

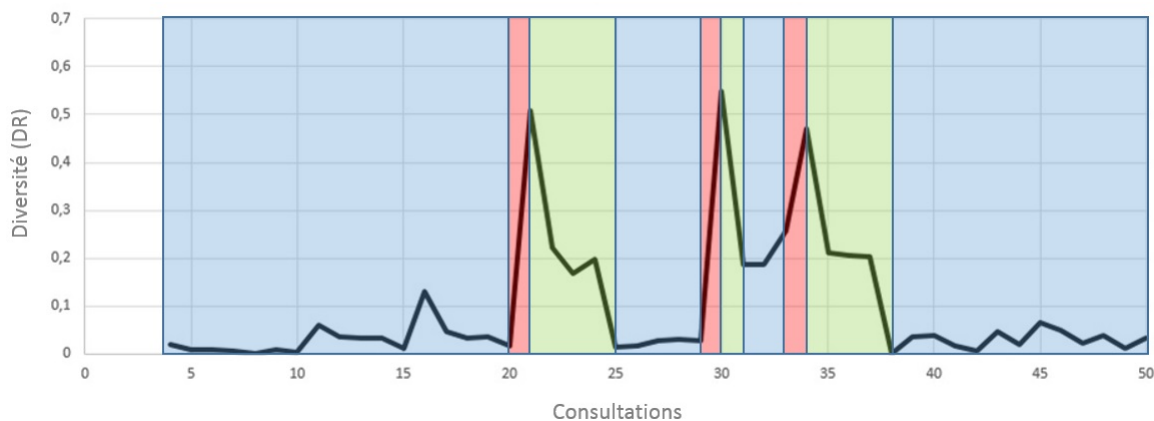


FIGURE 3.5 – Comportements remarquables de l'évolution de la diversité. Corpus de Lastfm, utilisateur 8, historique de taille 3, coefficient α fixé à 10

Les moments de stabilité (en bleu) correspondent aux périodes pendant lesquelles les items consultés possèdent des caractéristiques qui sont relativement similaires (*i.e.* les variations de diversité sont inférieures à un seuil donné). Nous employons le terme "relativement" car nous n'observons pas de période où la valeur de la diversité est constante. Une explication de ce comportement serait de dire que l'utilisateur écoute des musiques qui correspondent à une activité, une envie par exemple et que cette dernière couvre plusieurs musiques. Par exemple, l'utilisateur prend son petit-déjeuner et souhaite écouter des musiques calmes appartenant à un certain genre possédant un tempo lent.

Les pics de diversité (en rouge) correspondent aux moments où l'utilisateur consulte un item possédant des attributs différents de ceux ayant précédemment été consultés. Si nous reprenons l'exemple précédent, ayant fini son petit-déjeuner, l'utilisateur choisit d'écouter des musiques plus énergiques que les musiques antérieures. Les attributs de la nouvelle musique étant alors très différents de toutes les musiques précédentes, une forte valeur de diversité est calculée par notre modèle.

Suite à un pic de diversité, il est possible d'observer deux comportements. Le cas où l'utilisateur spécialise ses consultations (ce qui est le cas sur ce graphique) et le cas où l'utilisateur continue de diversifier ses consultations (cas qui n'est pas présent sur ce graphique). Dans le premier cas, si l'utilisateur continue de consulter des ressources similaires à celle ayant provoqué un pic de diversité, cela aura pour conséquence de diminuer la quantité de diversité. En effet, les items de l'historique et de la cible partageant davantage de caractéristiques similaires, la diversité calculée sera alors plus faible que la diversité calculée au pas de temps précédent. Toujours dans le cadre de l'exemple précédent, l'utilisateur continue d'écouter des musiques énergiques et l'on observe alors une diminution de la diversité de cet attribut. Dans le second cas, les items consultés sont tous différents en termes de caractéristiques et de ce fait, nous observons un niveau élevé de diversité. Ce comportement peut par exemple être observé lorsqu'un utilisateur lance une liste de lecture qui contient l'ensemble de ses musiques favorites. Cette liste de lecture n'étant pas créée sur la base de caractéristiques communes telles que pourrait l'être une liste de lecture pour

62. Ce qui est également le cas pour de nombreux utilisateurs également.

"se concentrer", "faire du sport" ou "danser" par exemple, un fort niveau de diversité pourra être observé. Cela peut également correspondre au cas où l'utilisateur ne sait pas encore ce qu'il souhaite écouter puisqu'il se situe entre deux contextes par exemple. Dans ce cas, il peut écouter différents morceaux qui sont très divers entre eux jusqu'à orienter son choix vers certaines caractéristiques lui convenant. Un contexte explicite aura une influence sur ce que va consulter un utilisateur. Cela ne veut pas dire pour autant qu'un même contexte donnera nécessairement lieu aux mêmes consultations. C'est pourquoi chercher à s'adapter uniquement au contexte explicite de l'utilisateur peut ne pas être suffisant.

Outre la visualisation graphique de la diversité, il est également possible de savoir comment la diversité évolue au cours du temps en calculant la valeur de la différence finie arrière⁶³ de la diversité à chaque pas de temps. Nous nommons cette valeur DR' et nous la calculons à l'aide de l'Équation 3.11.

$$DR'(c_t^u) = DR(c_t^u) - DR(c_{t-1}^u) \quad (3.11)$$

La valeur de DR' est donc une valeur comprise entre -1 et 1 et nous informe de la tendance de l'évolution de la diversité. Autrement dit, cette valeur nous indique si la diversité à un instant donné t est plus ou moins importante que la diversité à l'instant $t - 1$. Contrairement à la valeur de DR et comme nous pouvons le constater sur la Figure 3.6, les valeurs DR' peuvent être négatives. Cette valeur nous permet donc de savoir si un utilisateur diversifie ou spécialise ses consultations. Une valeur négative de DR' à un instant t nous informe que cette consultation apporte moins de diversité que la consultation à $t - 1$ et donc que l'utilisateur consulte des items qui partagent des attributs communs. Inversement, une valeur de DR' positive à l'instant t nous informe que l'utilisateur consulte des items qui possèdent des caractéristiques différentes par rapport à l'historique considéré.

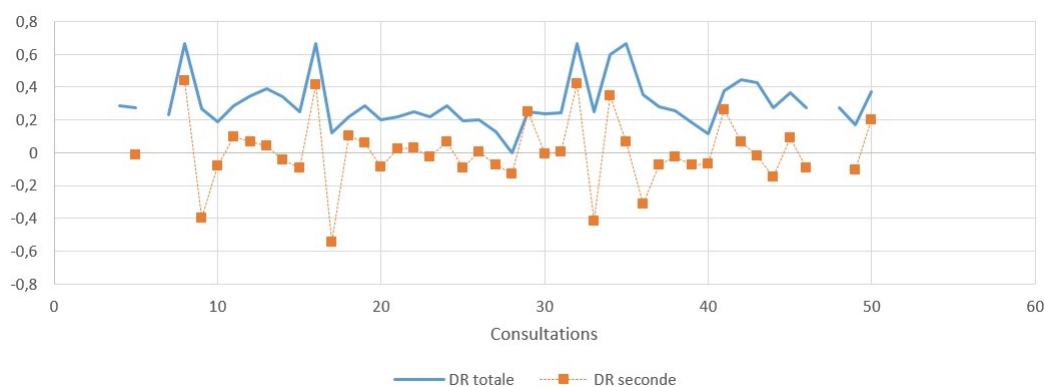


FIGURE 3.6 – Évolution de la DR et de la dérivée de DR (tous attributs compris). Corpus de Deezer, utilisateur 200, historique de taille 3

3.3.2 Influences des paramètres sur la modélisation

Jusqu'à présent, nous avons présenté le fonctionnement de notre modèle dans sa configuration la plus simple. Même si nous définissons la diversité de manière objective en nous intéressant aux valeurs des attributs, nous devons garder à l'esprit que cette modélisation de la diversité a pour but de comprendre le comportement des utilisateurs qui possèdent un regard qui leur est

63. La différence finie arrière est approximativement égale à la valeur de la dérivée discrète, modulo une faible erreur d'approximation déterminée par la formule de Taylor. Pour cette raison, nous utiliserons indifféremment les termes "dérivée de DR" et "différence finie arrière" dans la suite de ce manuscrit.

propre sur la diversité. Par exemple, un utilisateur qui n'écoute que du Rock trouvera diverse une musique du genre Hard rock alors qu'un utilisateur aux goûts beaucoup plus éclectiques ne sera pas forcément sensible à cette différence. Dans cette section, nous allons présenter les différents paramètres intervenant dans notre modèle. Ces différents paramètres peuvent être utilisés soit pour adapter le modèle aux profils des utilisateurs, soit pour rendre sensible notre modèle à certains événements particuliers.

Gestion des informations manquantes

Comme nous l'avons précisé précédemment, lorsqu'aucun attribut de la cible n'est connu, il n'est pas possible de calculer une valeur de diversité et la diversité correspondante (la diversité apportée par la consultation qui constitue la cible) doit être NaN. Sur le graphique 3.7, nous présentons 3 courbes correspondantes aux mêmes données. Le premier graphique en bleu représente le cas où l'on possède toutes les valeurs de diversité. Le deuxième graphique en vert et le troisième graphique en rouge représentent respectivement le cas où l'on attribue la valeur NaN et le cas où l'on attribue la valeur 0 à la DR lorsque cette dernière n'est pas calculable. Nous pouvons constater que le fait de définir une valeur égale à 0 a eu pour conséquence lors des 30^{ème} et 36^{ème} consultations de provoquer une diminution brutale de la diversité alors que cette dernière est restée stable ou a fortement augmenté en réalité. Ne pas définir de valeur de diversité ou encore traiter les valeurs inconnues de manière différente apparaît alors comme étant une meilleure solution que de définir une valeur de diversité nulle.

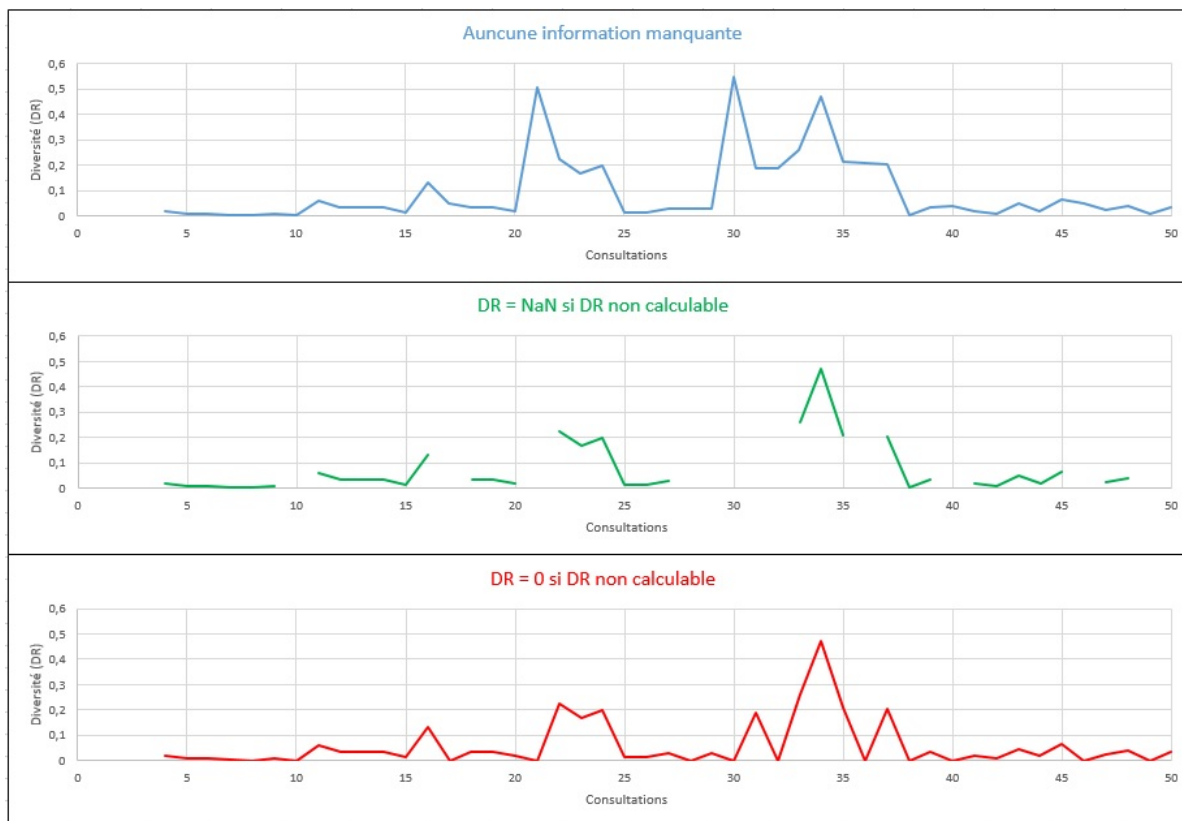


FIGURE 3.7 – Impact des données manquantes. Corpus de Lastfm, utilisateur 8, historique de taille 3, coefficient α fixé à 10

Sensibilité du modèle

Le coefficient α utilisé dans la Formule 3.7 possède également un impact sur l'évolution de la diversité au cours du temps. Nous pouvons constater sur la Figure 3.9 ci-dessous que plus ce dernier est important, plus le modèle sera sensible. Par sensibilité, nous entendons que les moindres écarts de valeurs des attributs numériques provoquent une mesure de diversité accrue. De plus, il est important de rappeler que la valeur de DR observée correspond à la valeur moyenne des valeurs de DR de chacun des attributs et que certaines de ces valeurs ne sont pas impactées par le coefficient α . Si nous ne tenions compte que des attributs numériques dans le calcul de la DR, le coefficient α posséderait davantage d'impact. Ce point nous permet également d'illustrer la complexité de travailler avec de nombreux attributs et des attributs de natures différentes. Même si l'utilisation de l'ensemble des attributs permet une modélisation plus précise du comportement des utilisateurs, il faut prendre en considération la complexification inhérente à l'utilisation de nombreux paramètres. À ce propos, nous verrons notamment dans le chapitre suivant de quelle manière il est possible de paramétrer notre modèle.

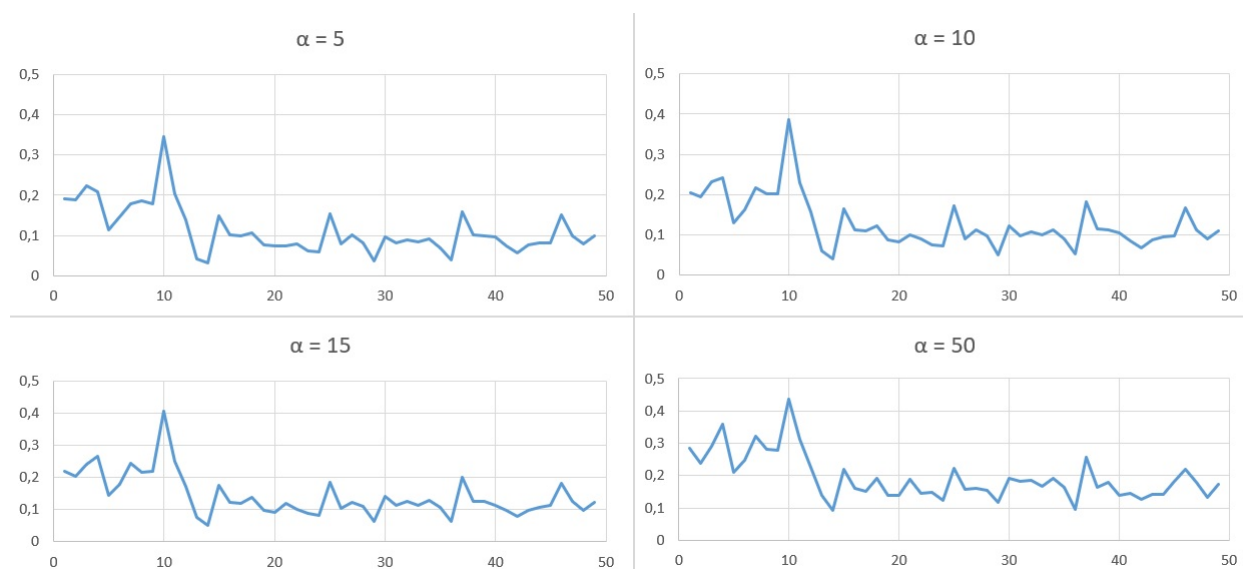


FIGURE 3.8 – Impact du coefficient α sur l'évolution de la DR. Corpus de Lastfm, utilisateur 25, historique de taille 3

Taille de l'historique

Dans le calcul de la DR utilisée par notre modèle, la taille de l'historique constitue également un paramètre pouvant être modifié. Ce dernier permet de définir si la diversité sera calculée sur un historique très restreint ou si ce dernier tient compte d'un nombre plus important de consultations. Les graphiques de la Figure 3.9 ci-dessous nous montrent également que la taille de l'historique a un impact sur la valeur de diversité calculée. Plus l'historique est grand, plus la courbe de l'évolution de la diversité a tendance à se lisser. Cela s'explique par le fait que la diversité est calculée sur la moyenne de l'ensemble des diversités calculées entre la cible et les éléments de l'historique. Un historique de grande taille a donc pour conséquence de supprimer certaines variations de diversité. Comme nous pouvons le voir, certaines variations importantes de diversité, présentes avec un historique de taille 1, sont très atténuées lorsque l'on considère un historique de taille 6.

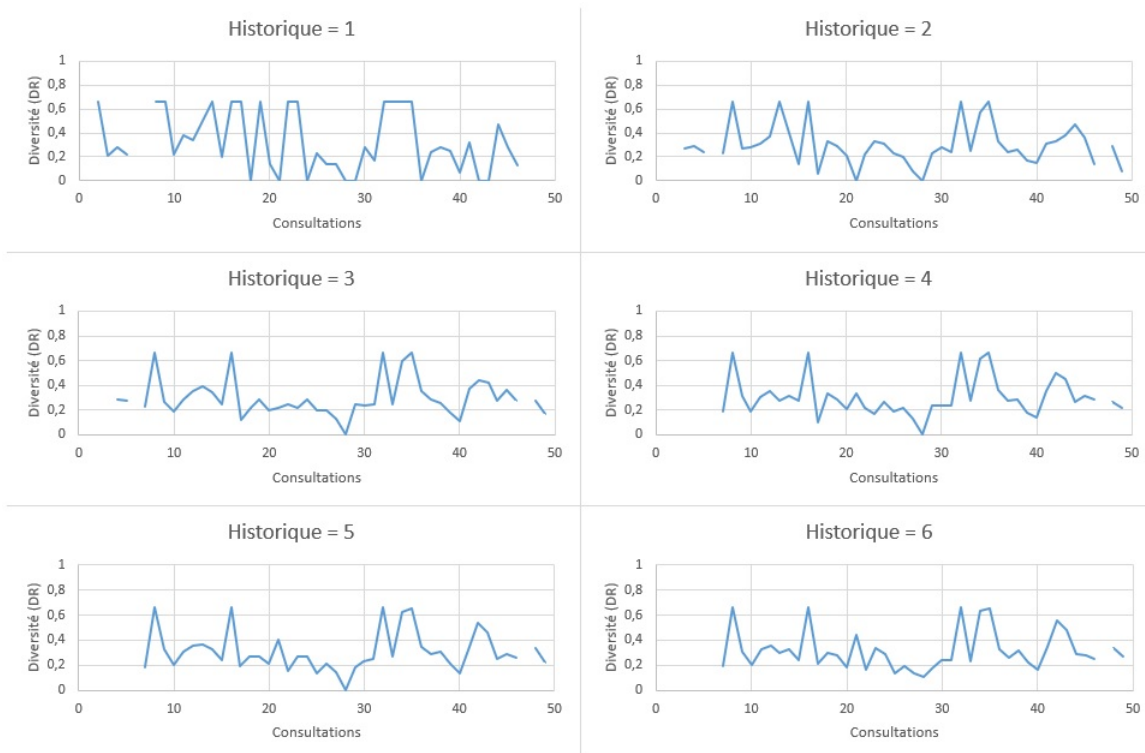


FIGURE 3.9 – Impact de la taille de l'historique. Corpus de Deezer, utilisateur 50, historique de taille 3, coefficient α fixé à 10

Coefficient d'éloignement

Jusqu'à présent, les items de l'historique avaient tous un poids identique dans le calcul de la diversité. Un autre paramètre de notre modèle consiste à accorder des poids différents aux diversités impliquées dans le calcul de DR en fonction de la distance séparant les items de l'historique et la cible. Pour ce faire, nous proposons deux cas opposés. Dans le premier cas (que nous appellerons DR_{croiss}), plus l'item de l'historique est proche de la cible et plus sa valeur de diversité impactera la diversité globale. Inversement, dans le second cas (que nous appellerons $DR_{decroiss}$), le poids de la diversité entre la cible et un item de l'historique aura d'autant moins d'impact sur la diversité globale que l'item sera proche. La Figure 3.10 ci-dessous représente ces deux cas :

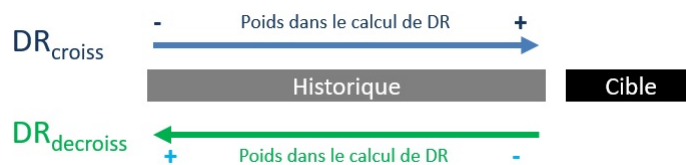


FIGURE 3.10 – Schéma représentant les deux coefficients d'éloignement

La formule associée au calcul de DR avec ces coefficients d'éloignement croissant ou décroissant est reportée dans l'Équation 3.12 ci-dessous :

$$DR_{avec_eloignement}(i, C) = \frac{\sum_{t=1..m} \delta(t)(1 - similarité(i, c_t))}{\sum_{t=1..m} \delta(t)} \quad (3.12)$$

Pour chaque coefficient (croissant ou décroissant), il est alors possible de choisir la fonction que l'on souhaite utiliser (linéaire, quadratique. . .). Dans le cas où aucun coefficient n'est utilisé, cela revient à utiliser un coefficient constant. Les deux coefficients peuvent être représentés comme suit :

- DR_{croiss} : $\delta(t) = f(t)$
- $DR_{decroiss}$: $\delta(t) = f(card(C) - t)$

Dans la Figure 3.11 ci-dessous, nous avons utilisé un coefficient linéaire selon la fonction $f(x) = x$.

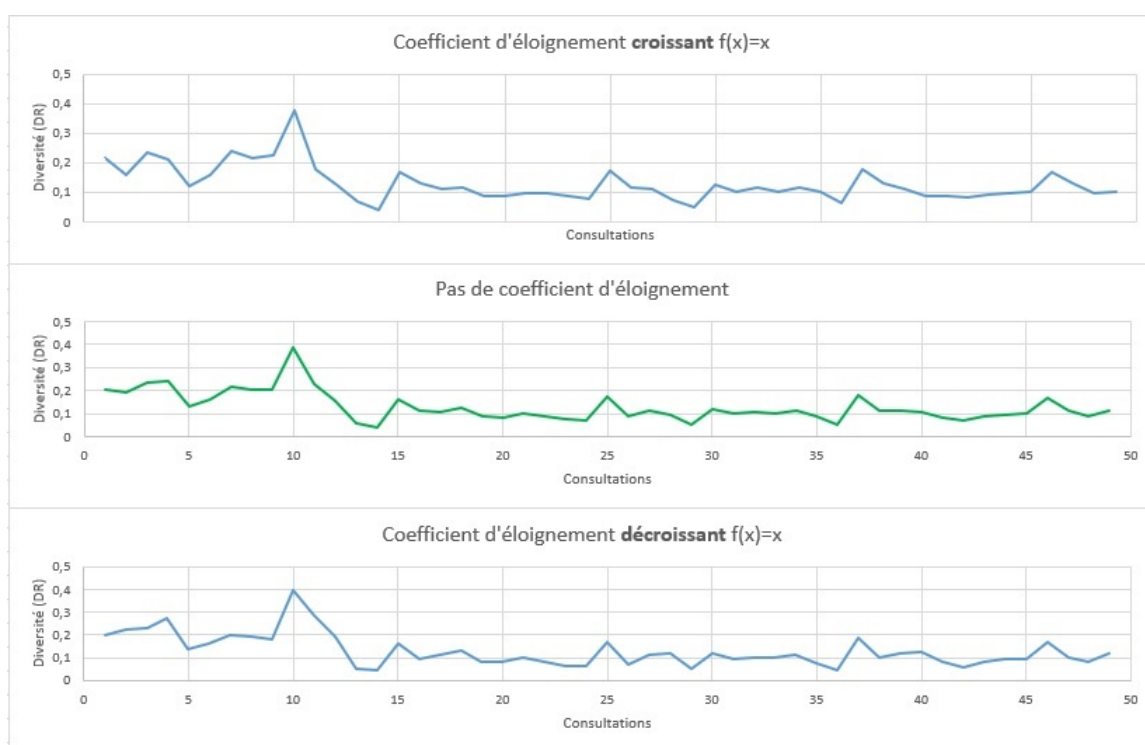


FIGURE 3.11 – Impact des coefficients d'éloignement. Corpus de Lastfm, utilisateur 2, historique de taille 3, coefficient α fixé à 10

Bien que possédant tout de même un impact sur l'évolution de la diversité, le coefficient d'éloignement n'impacte pas significativement cette dernière. Malgré cela, ce paramètre peut être utile s'il est adapté à chaque utilisateur. Par exemple, dans le cadre de la musique, un utilisateur peut accorder beaucoup d'importance aux dernières musiques qu'il vient d'écouter et il est alors primordial pour lui de ne pas lui proposer de transitions trop importantes. Inversement, un autre utilisateur peut être focalisé sur un historique un peu moins récent. Nous pouvons alors imaginer que, pour cet utilisateur, son intérêt est surtout porté sur le lien entre la consultation courante et les quelques consultations précédentes. Cependant, ces exemples ne constituent que des hypothétiques cas d'usages. Une étude utilisateur doit être conduite afin de savoir si les différences induites par ces deux coefficients font sens pour les utilisateurs et de quelle manière ils pourraient être exploités. Ce point constitue par ailleurs l'une nos perspectives de recherche.

Poids des attributs

La diversité est calculée à partir d'un ensemble d'attributs dont l'impact de chacun d'entre eux

peut être pondéré dans le calcul de la DR (totale). En effet, il est possible d'accorder plus d'importance à la diversité apportée par un attribut ou un ensemble d'attributs dans la valeur de DR par rapport aux autres attributs. Dans l'Équation 3.3, le coefficient β est propre à chaque attribut et permet d'attribuer des poids différents à chaque attribut. La configuration de poids utilisée, c'est-à-dire l'ensemble des poids accordés à chaque attribut peut être propre à chaque utilisateur ou être commune à l'ensemble des utilisateurs. La Figure 3.12 ci-dessous représente l'impact du poids des attributs dans la modélisation de la diversité. Afin de mettre en évidence l'impact de différents poids, nous avons calculé la DR totale uniquement avec 3 attributs qui sont la *danceability*, la *acousticness* ainsi que l'*instrumentalness*⁶⁴. Sur chacun des graphiques présentés dans la Figure 3.12, les poids de chacun des attributs sont précisés.

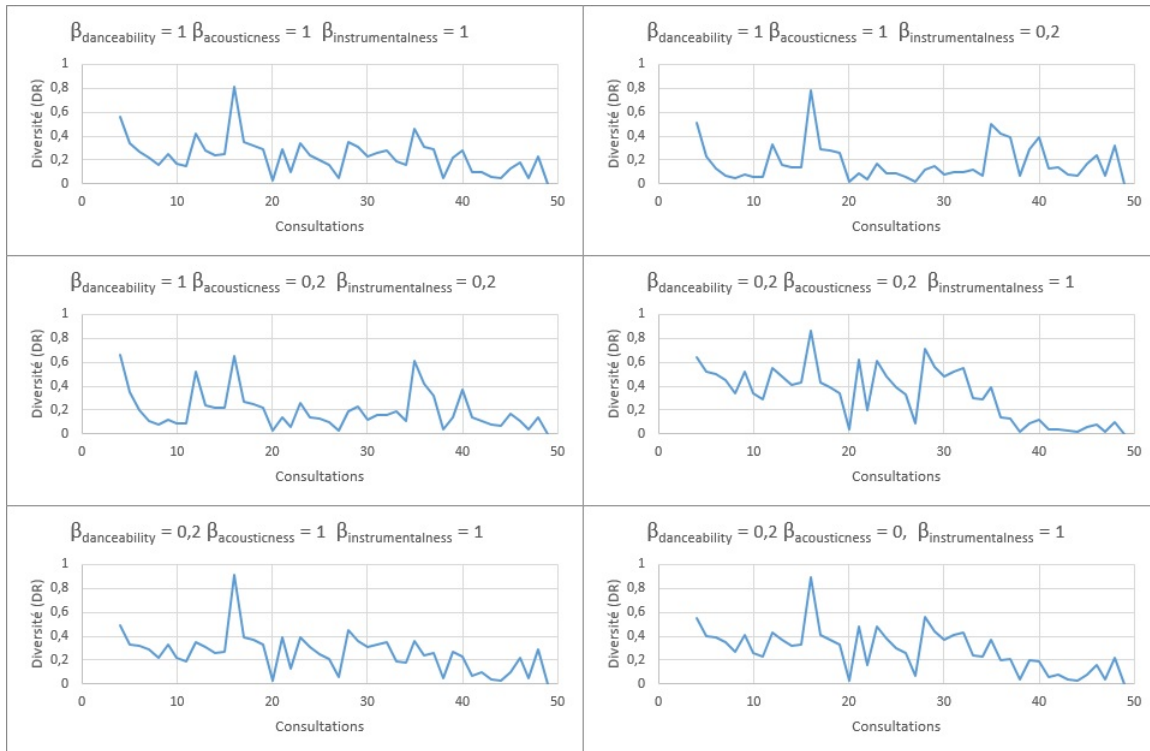


FIGURE 3.12 – Impact du poids des attributs. Corpus de Lastfm, utilisateur 2, historique de taille 3, coefficient α fixé à 10

Nous pouvons constater, d'après ces différents graphiques, que les différentes combinaisons des poids des attributs impactent les valeurs de DR totale et donc la modélisation de la diversité. Par exemple, lors que le poids de l'attribut *instrumentalness* est beaucoup plus important que les autres poids des attributs (graphique du milieu sur la droite), le nombre de fois où la diversité évolue fortement est plus important que pour les autres configurations. Inversement, lorsque les poids des attributs *danceability* et *acousticness* sont forts par rapport à l'*instrumentalness* (graphique du haut sur la droite), les variations de diversité sont beaucoup plus faibles, notamment entre la 20^{ème} et la 30^{ème} consultation. Ainsi, utiliser des poids différents pour les attributs permettra de personnaliser la modélisation de la diversité en fonction des préférences des utilisateurs ou encore de paramétrer notre modèle afin de détecter certains comportements. Alors

64. Nous avons choisi ces trois attributs en particulier dans le cadre cet exemple, mais il est possible de modéliser l'évolution temporelle de la diversité à partir de toutes les combinaisons d'attributs et de poids possibles

que le coefficient α dans le cas des attributs numériques intervient dans la sensibilité du modèle, la configuration des poids intervient pour sa part afin de mettre en avant certains attributs en particulier.

3.3.3 Adaptation aux items de natures différentes

Dans les jeux de données que nous avons présentés, l'ensemble des items qui sont consultés sont des musiques possédant les mêmes caractéristiques. De ce fait, il est toujours possible de calculer une valeur de diversité sur l'ensemble de ces attributs (hormis le cas dans lequel certaines données sont manquantes). Cependant, dans certains domaines applicatifs, les items consultés ne sont pas forcément des items de même type. Par exemple, sur un réseau social professionnel, un utilisateur peut consulter une offre d'emploi, le profil d'un utilisateur et ensuite le profil d'une entreprise. Ou encore, sur un site d'e-commerce, une séquence de consultations peut contenir des items de natures différentes tels qu'un livre, un DVD, une figurine et de nouveau un DVD. La diversité telle que nous l'avons définie ne peut être calculée que sur la base des attributs qui sont comparables entre eux. Or les livres et les DVD possèdent des attributs qui leur sont propres, dont certains qu'ils partagent avec d'autres items et d'autres qu'ils sont les seuls à posséder. Calculer une valeur de diversité ne pourra donc se faire que sur les attributs communs entre deux items de natures différentes. Par exemple, dans le cas où l'on calcule la diversité existante entre un livre et un DVD, il va de soi qu'il ne sera pas possible de calculer une valeur de diversité sur l'attribut "acteurs" avec un livre étant donné que la liste des acteurs est un attribut spécifique aux films. Cependant, il est tout à fait possible de calculer la diversité portant sur leurs genres, car un film et un livre possèdent tous les deux cet attribut. Ainsi, si nous reprenons la formule de similarité présentée dans l'Équation 3.3 présentée précédemment, si aucun attribut n'est commun entre deux items alors la similarité sera de NaN, sinon la similarité sera calculée uniquement sur les attributs communs.

Nous présenterons dans la suite de ce manuscrit dans la section 4.2.3 les résultats que nous avons obtenus en simulant un corpus de données contenant des items possédant des types différents (et donc des attributs différents).

3.4 Discussion

Dans ce chapitre, nous avons présenté la première approche de l'état de l'art à notre connaissance permettant de modéliser l'évolution temporelle de la diversité. Le modèle que nous proposons possède de nombreux avantages. Premièrement, il dispose d'une complexité algorithmique constante ($\mathcal{O}(k)$ où k correspond à la taille de l'historique utilisé) et notre modèle peut donc aisément être déployé en ligne pour modéliser en temps réel l'évolution de la diversité. De plus, le temps nécessaire pour effectuer les calculs de diversité sur 2 000 000 d'écoutes (inférieur à 1 min.) confirme cette idée. Deuxièmement, cette évolution nous permet de comprendre et de caractériser rapidement le comportement des utilisateurs sur de nombreuses dimensions que sont les caractéristiques des items. Par exemple, il est possible de savoir si l'utilisateur spécialise ou diversifie ses consultations sur certains attributs. Cette information peut alors être utilisée afin d'adapter les recommandations comme nous le verrons dans le Chapitre 6 de ce manuscrit. Enfin, notre modèle peut être paramétré de nombreuses manières. Il est par exemple possible d'accorder des poids plus importants à certains attributs dans le calcul de la diversité totale ou encore d'adapter la sensibilité aux écarts de valeurs des attributs numériques. La taille de l'historique peut également être adaptée en fonction des situations et de la sensibilité désirée. À ce stade du

manuscrit, il ne nous est pas encore possible de savoir quel est le meilleur paramétrage à utiliser. Seuls des métriques de performances en termes de qualité des recommandations, la corrélation avec d'autres indicateurs du comportement des utilisateurs ou des retours utilisateurs sur la perception des comportements que nous détectons nous permettront de définir la configuration optimale des paramètres à utiliser.

L'analyse de l'évolution de la diversité dans le parcours d'un utilisateur (*cf.* Figure 3.5) nous a permis de montrer qu'il pouvait exister certains comportements particuliers dans l'évolution de la diversité. Nous avons notamment vu qu'il existait 3 comportements remarquables : des périodes où la diversité varie peu, des périodes où la diversité augmente ou décroît progressivement et enfin des moments où la diversité varie soudainement. Ces soudaines variations de diversité que nous avons nommées "**pics de diversité**" sont particulièrement intéressantes. Ces pics de diversité témoignent d'un changement brutal des caractéristiques et peuvent être représentatifs d'un changement de contexte. En effet, nous avons fait précédemment l'hypothèse selon laquelle les items consultés au sein d'un même contexte partagent des caractéristiques communes. Un changement de caractéristiques soudain pourrait alors être le témoin d'un changement de contexte, d'autant plus si la diversité diminue par la suite. Nous allons donc nous intéresser dans la section suivante, à ces pics de diversité, mais surtout aux correspondances pouvant exister entre ces derniers et certains événements initiés par les utilisateurs.

Nous avons atteint notre **premier** objectif : définir un modèle permettant de modéliser l'évolution temporelle de la diversité dans le parcours de consultations d'un utilisateur.

Objectif suivant : tester l'existence de liens entre les pics de diversité et certains événements relatifs au comportement des utilisateurs.

Publication :

- L'Huillier, A., Castagnos, S., and Boyer, A. (2014). Understanding Usages by Modeling Diversity over Time. volume 1181 of ACM UMAP 2014 Extended Proceedings, Aalborg, Denmark.

Chapitre 4

Diversité et contexte : DANCE

Sommaire

4.1	Contexte implicite	73
4.2	Détection des contextes implicites	75
4.2.1	Performances <i>a posteriori</i>	77
4.2.2	Performances en temps réel	86
4.2.3	Robustesse du modèle DANCE	92
4.2.4	Amélioration des performances	98
4.3	Discussion	103

Dans ce chapitre, nous exploitons le modèle de suivi de l'évolution temporelle de la diversité que nous avons présenté précédemment afin de proposer une nouvelle approche de détection du contexte. Contrairement aux travaux de l'état de l'art qui définissent le contexte à partir des dimensions et des facteurs contextuels intrinsèques à l'utilisateur et à son environnement, nous proposons une nouvelle définition du contexte basée sur les caractéristiques des items pour une meilleure préservation de la vie privée. La présentation de cette nouvelle définition du contexte dit implicite ainsi que des perspectives alors offertes constitueront la première partie de ce chapitre. Dans une seconde partie, nous présenterons le modèle DANCE (*Diversity And Natural Context Elicitation*) permettant d'isoler ces contextes sur la base de l'analyse de l'évolution de la diversité au cours du temps. Nous présenterons et discuterons des performances de ce modèle sur les corpus de Deezer et de Lastfm. Nous prouverons notamment la pertinence des contextes implicites que nous isolons en démontrant qu'ils sont liés à des contextes explicites. Par la suite, nous étudierons la robustesse de notre modèle face aux données manquantes et nous verrons que notre modèle peut être utilisé sur des items de natures différentes. Enfin, nous démontrerons qu'il est possible d'améliorer les performances de notre modèle à l'aide d'un algorithme génétique.

4.1 Contexte implicite

Dans l'état de l'art, nous avons pu voir que le contexte des utilisateurs était la plupart du temps défini par un ensemble de dimensions et de conditions contextuelles (cf. *supra* Chapitre 1, p.28) permettant de représenter explicitement le contexte des utilisateurs. Nous avons décidé de synthétiser ces travaux à travers une définition plus précise de ces contextes, que nous appellerons dans la suite de ce manuscrit "contextes explicites".

Contexte explicite : ensemble des critères permettant de caractériser la situation dans laquelle se trouve l'utilisateur courant. Il peut s'agir en particulier de critères temporels (date et heure, jour de la semaine, période de l'année...), spatiaux (localisation, destination...), de son environnement social, de son état mental, de son environnement technologique (*smartphone*, tablette, ordinateur)...

Comme nous avons également pu le voir dans l'état de l'art, malgré l'amélioration de la qualité des recommandations inhérente à la prise en compte de ces contextes, ces derniers possèdent certains inconvénients tels qu'une complexification de la représentation des préférences des utilisateurs, des problèmes de généralité mais surtout de contraintes liées au respect de la vie privée des utilisateurs. De plus, nous avons également formulé l'hypothèse selon laquelle l'étude et la prise en compte de l'évolution de la diversité au cours du temps pourraient permettre de déterminer le contexte des utilisateurs. Ainsi, à la notion de contexte explicite précédente, nous opposons⁶⁵ la notion de contexte implicite que nous définissons ci-dessous :

Contexte implicite : ensemble des caractéristiques communes aux items co-consultés pendant un certain laps de temps.

De la même manière qu'il existe des préférences explicites et implicites se différenciant sur la manière dont elles sont déterminées [Jones, 2010, Jawaheer et al., 2014], nous considérons qu'il existe des contextes explicites et des contextes implicites. La différence majeure qui existe entre le contexte implicite et le contexte explicite concerne les informations qui sont utilisées. Dans le cas du contexte explicite, les informations contextuelles portent sur l'utilisateur et son environnement lors des interactions entre l'utilisateur et les items. Le contexte explicite tel que nous le définissons ne fait pas seulement référence au cas où il est possible de définir le contexte de manière factuelle comme "être en soirée avec ses amis en semaine" ou encore "faire du sport à un rythme important". Nous considérons qu'un SR exploite des contextes explicites à partir du moment où au moins une information concernant l'utilisateur dans son interaction avec un item est exploitée. Cela est valable sur les informations élicitées de la même manière que lorsqu'il est question de demander à l'utilisateur de définir son activité (sport, fête...), ou les informations dites "brutes" telles que les informations récupérées à partir de capteurs (GPS, bruit environnant...). Le contexte implicite quant à lui n'est constitué que des informations décrivant les items. Nous n'entendons alors pas exploiter par la suite (et si cela était possible) des attributs donnant des indications sur le contexte explicite dans lesquels se trouve l'utilisateur⁶⁶. Autrement dit, le but des contextes implicites est double : définir le contexte d'une autre manière et non pas de nous servir du contexte implicite afin de déduire le contexte explicite de l'utilisateur (ne pas être intrusif) et ne pas solliciter l'utilisateur (en lui demandant de renseigner son contexte). Il s'agit à titre d'exemple de constater et d'exploiter le fait qu'un utilisateur a écouté d'affilée plusieurs chansons de Rock avec un tempo élevé pour adapter les recommandations à ces besoins, et non de déduire ou de prendre en compte le fait qu'il écoute ce genre de musiques car il fait la fête ou qu'il souhaite

65. Le terme d'opposition fait référence non pas à une opposition en termes d'objectif mais en termes de moyen. Alors que le contexte explicite repose sur des informations personnelles de l'utilisateur, le contexte implicite repose uniquement sur les attributs des items consultés.

66. Par exemple, l'attribut *danceability* est un attribut qui peut dans une certaine mesure être utilisé afin de déduire si l'utilisateur écoute des musiques pour danser. Cependant, nous n'entendons pas croiser les attributs entre eux ou utiliser quelque méthodologie qui soit afin de confirmer ou d'infirmer le contexte explicite de l'utilisateur.

danser.

4.2 Détection des contextes implicites

D’après la définition précédente, les items consultés au sein d’un contexte implicite partagent des caractéristiques communes. Dès lors que les caractéristiques d’une nouvelle consultation sont trop différentes des caractéristiques des consultations précédentes (par rapport à un seuil), cette consultation ne peut pas appartenir au même contexte. Autrement dit, lorsqu’un niveau important de diversité est apporté par une consultation, nous considérons qu’il s’agit d’un nouveau contexte implicite. De ce fait, une suite de consultations peut être divisée en plusieurs contextes implicites. Selon ce principe, nous avons mis au point le modèle **DANCE** (acronyme pour *Diversity and Natural Context Elicitation*). Le modèle **DANCE** s’appuie sur l’approche de modélisation temporelle de la diversité que nous avons présentée dans le chapitre précédent et isole les contextes implicites dans une séquence de consultations.

Si nous reprenons l’exemple que nous avons présenté dans le chapitre précédent, un exemple de découpages en contextes implicites est représenté par la Figure 4.1 si dessous :

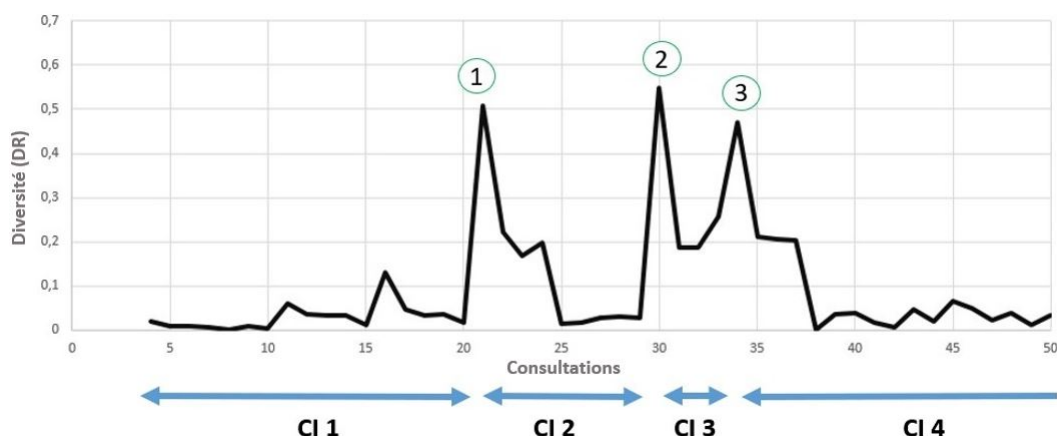


FIGURE 4.1 – Exemple de découpage en contextes implicites⁶⁷

Sur cette figure que nous avons présentée dans le chapitre précédent, nous avons observé qu’il existait 3 variations importantes (“pics”) de diversité ainsi que des périodes pendant lesquelles la diversité apportée par les consultations était relativement faible. Si la diversité observée pendant une certaine période est relativement faible, cela signifie que les items partagent des caractéristiques similaires et donc que cette période correspond à un contexte implicite. Ainsi, à partir de la Figure 4.7 nous pouvons constater l’existence de 4 contextes implicites (*CI 1*, *CI 2*, *CI 3* et *CI 4*) qu’il nous faut parvenir à isoler.

Bien qu’il existe des méthodes de l’état de l’art permettant de détecter des changements de tendance au cours du temps, nous avons choisi de mettre au point notre propre fonction de détection car dans un premier temps, nous nous intéressons davantage à des variations soudaines

67. Le graphique utilisé correspond à celui présenté dans sur la Figure 3.5. Il s’agit de 50 consultations de l’utilisateur 8 du corpus de Lastfm avec un historique de taille 3 et un coefficient α fixé à 10. Sauf indication contraire, les prochaines figures utilisant un exemple de modélisation de la diversité seront issues de ces mêmes observations.

plutôt qu'à des tendances. Les techniques de détection de changement de tendance (*Change Point Analysis*) telles que [Taylor, 2000],[Killick and Eckley, 2014] ne sont alors pas adaptées à notre problème. La Figure 4.2 tirée des travaux de [Killick and Eckley, 2014] représente un exemple de jeux de données pour lesquels ces méthodes sont utilisées ainsi que les résultats obtenus. Nous pouvons constater que les variations sont beaucoup plus fréquentes que les courbes d'évolution de la diversité que nous avons présentées et que les changements détectés ne correspondent pas à nos besoins. En effet, ces méthodes détectent des variations non pas à l'échelle d'un point, mais sur des fenêtres d'observation, et ce, en comparant des distributions. Or, dans notre cas, nous cherchons à détecter de variations de diversité remarquables d'une consultation à l'autre. De plus, étant donné que nous souhaitons utiliser par la suite notre modèle en temps réel, nous ne pouvons pas nous permettre de détecter en retard un changement de contexte. Bien que ces méthodes soient écartées dans le cadre de cette thèse, l'utilisation de ces méthodes de détection de changement de tendance de l'état de l'art sur les résultats de modélisation temporelle d'évolution de la diversité constitue des perspectives intéressantes qui seront discutées dans la suite de ce manuscrit.

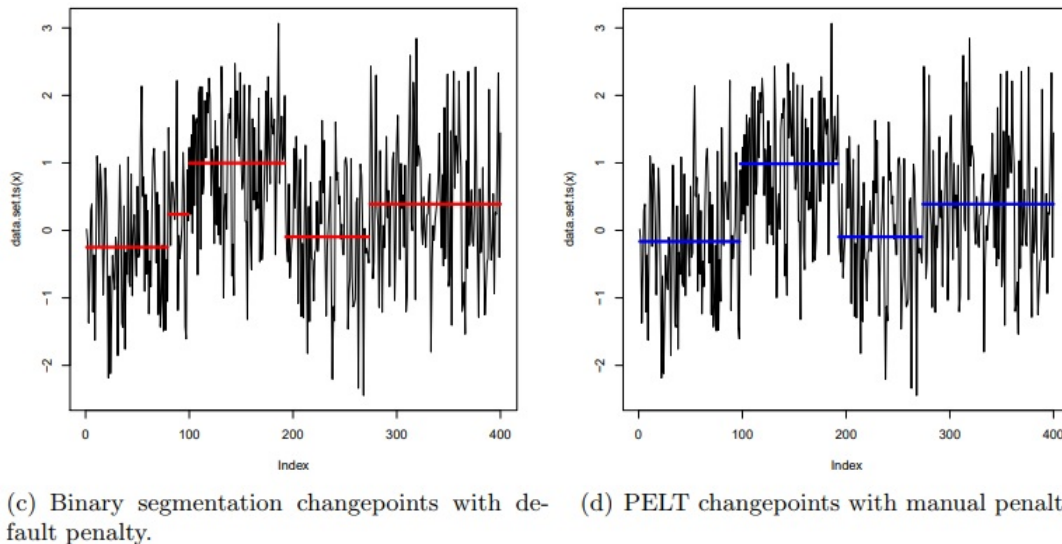


FIGURE 4.2 – Exemple d'utilisation de la méthode *change point analysis* [Killick and Eckley, 2014]

Le principe de notre méthode de détection de changements de contextes implicites consiste à comparer le niveau de diversité de chaque consultation par rapport à certaines conditions que nous aurons définies. À l'origine, les consultations passées, mais également les consultations futures (par rapport à la consultation testée) étaient impliquées dans notre méthode de détection. Pour cette raison, nous parlons de détection *a posteriori* étant donné que l'item consulté à $t + 1$ qui est une information qui n'est pas accessible en conditions réelles d'application intervient dans la méthode de détection. Cette première approche nous a permis d'obtenir une preuve de concept quant à l'efficacité du modèle DANCE et à la pertinence des contextes implicites. Nous avons ensuite modifié notre formule de détection afin de respecter le fonctionnement en temps réel et donc de ne plus être dépendant de l'item contenu dans la consultation suivante pour déterminer si un changement de contexte est intervenu. Ainsi, seules les données disponibles à l'instant t sont utilisées dans notre seconde méthode de détection de changements de contextes

implicites. Le schéma 4.3 ci-dessous illustre le cas *a posteriori* et en temps réel :

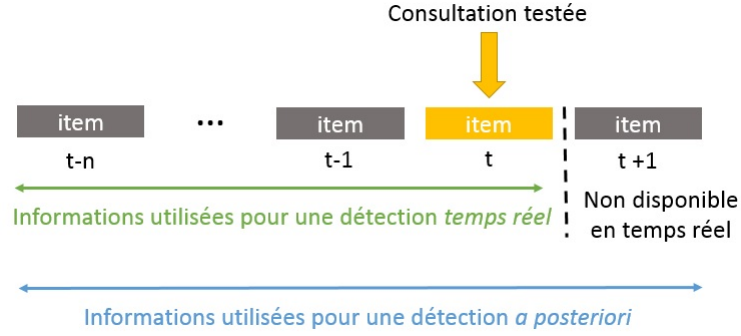


FIGURE 4.3 – Détection *a posteriori* et en temps réel

4.2.1 Performances *a posteriori*

Détecter un changement de contextes implicites comme nous l’avons expliqué revient à détecter ce que nous appelons des pics de diversité. Afin de détecter ces “pics” de diversité, nous proposons la formule ci-dessous qui peut être résumée comme suit : lorsque les valeurs de diversité de la consultation précédente, courante et suivante sont connues, nous regardons si la diversité apportée par la consultation courante est supérieure à la diversité de la consultation précédente et de la consultation suivante.

$$\begin{aligned}
 & DR(c_{t-1}^u, C_{k,t-1}^u), DR(c_t^u, C_{k,t}^u), DR(c_{t+1}^u, C_{k,t+1}^u) \neq \text{NaN} \quad (1) \\
 & DR(c_{t-1}^u, C_{k,t-1}^u) < DR(c_t^u, C_{k,t}^u) > DR(c_{t+1}^u, C_{k,t+1}^u) \text{ et } DR'(c_t^u, C_{k,t}^u) > \tau \quad (2)
 \end{aligned} \tag{4.1}$$

Ainsi, pour qu’un changement de contexte implicite ait lieu, deux conditions doivent être réunies :

- (1) Les valeurs de DR utilisées dans les formules doivent toutes être définies, c’est-à-dire différentes de *NaN*. Nous choisissons de ne pas statuer sur un changement de contexte si les valeurs de DR ne sont pas connues. Cette situation peut survenir si tous les attributs de la cible et/ou que tous les attributs de l’ensemble des items contenus dans l’historique ne sont pas connus. Étant donné que nous souhaitons par la suite adapter les recommandations aux contextes détectés, il est judicieux de prolonger le contexte courant lorsque nous ne savons pas si un changement de contexte est survenu plutôt que de nous adapter à un hypothétique nouveau contexte.
- (2) La valeur de DR de la consultation courante doit être supérieure à la valeur de DR de la consultation précédente et de la consultation suivante. De plus, la valeur de la dérivée (DR’) doit être supérieure à un seuil. Cette dernière condition a pour but de ne détecter que les variations importantes de diversité. Sans cette condition, une séquence de consultation serait divisée en de nombreux CI comme nous pouvons le voir dans la Figure 4.4 ci-dessous étant donné qu’il existe beaucoup de “pics de diversité”. Nous pouvons constater que, même si les CI détectés sont en adéquation avec notre définition, le modèle de détection s’avère beaucoup trop sensible.

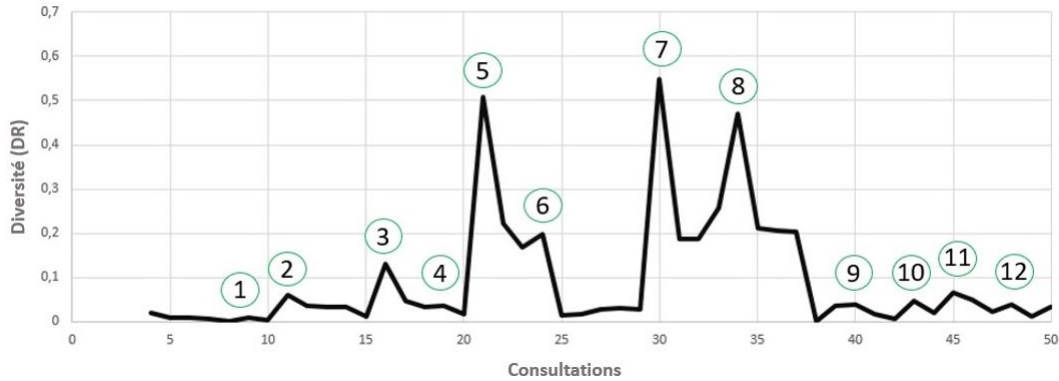


FIGURE 4.4 – Détections obtenues sans le paramètre τ

La présence du paramètre τ est donc indispensable afin que notre modèle ne soit pas sensible aux nombreuses variations de diversité pouvant exister. Même si deux consultations sont très proches en termes d'attributs, une diversité entre ces deux items sera tout de même présente étant donné que seules deux musiques identiques possèdent une diversité égale à 0. La Figure 4.5 ci-dessous illustre le rôle d'une valeur τ égale à 0,15. Alors que les consultations 1, 2, 3 et 4 remplissent la première condition de notre formule de détection, seules les consultations 1, 2 et 3 correspondent à de nouveaux contextes car la valeur de la dérivée pour cette consultation (représentée en vert) est suffisamment élevée.

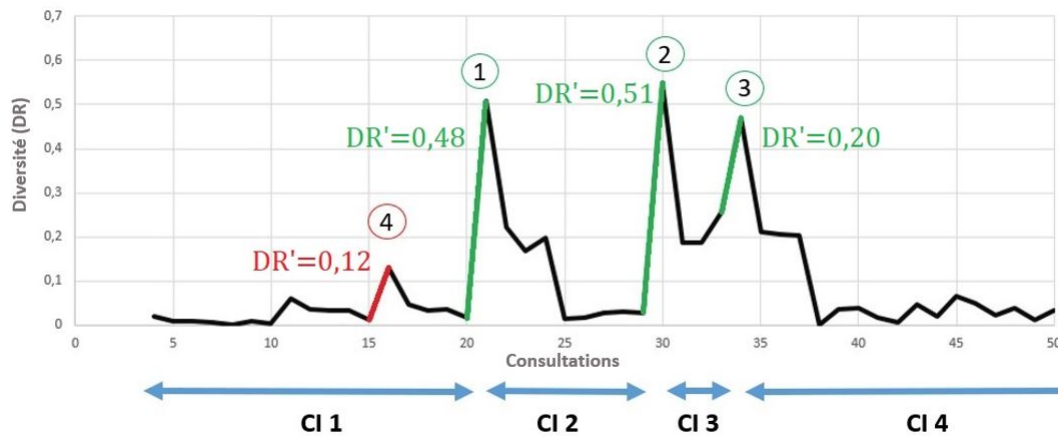


FIGURE 4.5 – Contextes implicites détectés si $\tau = 0,15$

Choisir une valeur de τ faible aura pour conséquence de détecter beaucoup de changements de contextes et une valeur élevée de τ aura pour conséquence de diminuer le nombre de changements de contextes. Il est important de distinguer à ce stade la légitimité des contextes implicites et leur utilité. En effet, quelle que soit la valeur de τ , tous les contextes détectés sont valides dans le sens où ils correspondent effectivement à des périodes pendant lesquelles les attributs partagent des caractéristiques similaires.

Toutefois, détecter trop souvent des changements de contextes implicites ou n'en détecter que très rarement selon la valeur de τ aura des conséquences sur l'utilité de ces contextes en termes de modélisation utilisateur ou encore de recommandation. Nous pouvons en conclure qu'il est

nécessaire de s'interroger sur la valeur de τ optimale et sur ce que nous considérons comme optimal. Dans un premier temps nous allons présenter les résultats de détection de changements de contextes implicites que nous avons obtenus en fixant une valeur de τ *a posteriori*, c'est-à-dire en utilisant des informations qui ne sont normalement pas accessibles en temps réel. Dans un second temps, nous présenterons dans la Section 4.2.2 comment nous pouvons adapter la formule de détection afin de pouvoir utiliser le modèle *DANCE* en temps réel et sans utiliser d'informations obtenues *a posteriori*.

Détection des contextes implicites

Le contexte explicite influe sur les caractéristiques des items que les utilisateurs vont préférer consommer/consulter. Par exemple, un utilisateur n'écouterait probablement pas de musique classique dans le contexte "sport" alors même qu'il écoute majoritairement de la musique classique dans les contextes précédents ou suivants. Connaissant les séquences de consultations d'items et les contextes explicites de ces consultations, il serait alors possible de s'intéresser aux variations de diversité existantes lorsque se produisent également des changements de conditions contextuelles. En utilisant des techniques d'apprentissage supervisé, il deviendrait possible de déterminer le paramètre τ optimal, c'est-à-dire le paramètre permettant de retrouver le plus de changements de contextes à partir de l'évolution de la diversité. Malheureusement, il n'existe pas dans la littérature de corpus de données (sans restriction au domaine de la musique) dans lesquels les données contextuelles ainsi que les séquences de consultations sont connues. Afin de pallier ce problème, nous faisons l'hypothèse suivante :

Hypothèse n° 1 : Sachant que le contexte est une dimension évoluant au cours du temps, le contexte d'un utilisateur n'est souvent pas le même d'une session d'écoute à une autre. Étant donné que les attributs des items consultés par les utilisateurs sont dépendants des contextes, un changement de session peut également correspondre à un changement de contexte implicite.

Une session d'écoute correspond à la période pendant laquelle un utilisateur consulte des items sans qu'il n'y ait d'interruption supérieure à une certaine durée. Ainsi, si la durée écoulée entre le *timestamp* d'une consultation, à laquelle on ajoute la durée de la musique correspondante, et le *timestamp* d'une nouvelle consultation est supérieure à un seuil, nous considérons que cette dernière appartient à une nouvelle session. Les raisons pour lesquelles nous faisons cette hypothèse sont illustrées par cet exemple : un utilisateur écoute de la musique de 8h jusqu'à 12h via son ordinateur à son travail. De 12h à 14h, il décide de partir prendre son repas puis revient à 14h et écoute de nouveau de la musique. Du point de vue du contexte explicite, nous avons bien à faire à un changement de contexte. La période avant le repas correspond à la condition "matin" de la dimension contextuelle "moment de la journée" et la période après le repas à la condition "après-midi" de cette même dimension. La dimension temporelle, bien qu'étant le seul indicateur de changement de contexte explicite que nous possédions, s'avère liée à d'autres dimensions contextuelles. Par exemple, si ce même utilisateur écoute de nouveau de la musique chez lui en rentrant du travail pour le déjeuner, son contexte physique aura également changé et il est probable qu'il possède alors d'autres préférences (dans ce contexte). Afin de calculer les sessions dans les deux corpus, nous avons fixé le seuil de changement de session à une durée de **15 minutes** [Castagnos, 2008, Gayo-Avello, 2009].

En plus des changements de contextes explicites que nous souhaitons détecter au travers des changements de sessions, nous avons également formulé une deuxième hypothèse :

Hypothèse n° 2 : Au sein d’un contexte explicite, les utilisateurs consultent des items partageant des caractéristiques communes permettant de définir également un contexte implicite. Un item possédant des caractéristiques non appropriées pour le contexte implicite ou explicite courant sera alors rejeté par l’utilisateur. Dans le cadre de la musique, ce cas peut être observé lorsqu’un utilisateur est passé à la musique suivante sans avoir écouté en entier la musique précédente.

En effet, selon [Jiang and He, 2016], le fait qu’une musique soit passée peut être interprété comme un *feedback* implicite de l’utilisateur quant au fait qu’il n’a pas apprécié la musique. Nous souhaiterions alors voir si le rejet d’une musique peut s’expliquer par un trop grand apport en diversité. Par exemple, si un utilisateur se situe dans un contexte dans lequel il écoute des musiques pour courir et qu’un *slow* est recommandé, alors il est possible que la musique soit passée/rejetée même si l’utilisateur apprécie les deux musiques. Afin de tester cette hypothèse, nous avons dû déterminer au préalable si les musiques consultées par les utilisateurs avaient été écoutées en entier. Cette information n’étant pas présente à l’origine dans les données que nous avons récupérées, nous avons dû la déterminer en utilisant les *timestamp* des consultations des musiques. Nous considérons qu’une musique est passée si la durée d’écoute est inférieure à 30 % de sa durée. Bien que nous n’ayons pas trouvé dans la littérature d’études recommandant le pourcentage d’écoute maximal afin de considérer qu’une musique est “passée”, nous avons choisi ce seuil car nous souhaitons détecter les musiques ayant été rejetées rapidement, ce qui témoignerait alors d’une aversion importante pour cette musique.

Bien que notre modèle permette de calculer une valeur de diversité, même si les attributs des musiques sont partiellement absents, nous avons restreint nos corpus afin de limiter les biais dus aux données manquantes. À ce stade, notre objectif n’est pas de démontrer la robustesse de notre modèle mais de nous intéresser au recouvrement qui peut exister entre un changement de session et un changement de contexte implicite. Nous souhaitons donc nous assurer que certaines données manquantes n’impacteraient pas nos observations. De plus, la robustesse de notre modèle face aux données manquantes est présentée dans la Section 4.2.3 et nous discuterons alors des performances de DANCE sur les corpus entiers que nous avons présentés. Les statistiques des sessions et des musiques passées dans les deux corpus “restreints” sont résumées dans le Tableau 4.1 ci-dessous :

TABLE 4.1 – Statistiques des sessions et des musiques passées dans les corpus ne comprenant pas d’information manquante

	Lastfm	Deezer
# Utilisateurs	99	2 454
# Consultations	183 374	249 042
# Musiques	38 477	51 809
# Artistes	5 063	7 503
# Sessions	20 668	21 449
Moy. consul./Session	8,87	11,61
# Musiques passées	5 052	75 426

D’après le Tableau 4.1 ci-dessus, nous pouvons constater qu’il existe quelques différences entre les deux corpus. Les utilisateurs du corpus de Deezer ont des sessions plus longues que

les utilisateurs du corpus de Lastfm. Une explication de cette différence peut être le nombre de musiques passées. En effet, les utilisateurs de Lastfm passent en moyenne 1 musique toutes les 36 musiques écoutées alors que les utilisateurs de Deezer passent en moyenne 1 musique toutes les 3 musiques écoutées. Si l'on décidait de supprimer les musiques passées et de recalculer le nombre moyen de musiques par session, alors la différence observée serait bien plus faible. Le corpus de Deezer est issu de la plateforme de musique en ligne éponyme contenant des millions de titres et dans laquelle des algorithmes de recommandation sont utilisés pour proposer les musiques suivantes (mode radio) ainsi que des *playlists*. Le corpus de Lastfm est basé quant à lui sur le principe du *Scrobbling* (cf. p.48) et nous ne savons donc pas si les items sont lus à partir de la collection personnelle de l'utilisateur ou à partir de services tiers tels que Youtube qui possèdent également des moteurs de recommandation et des systèmes de *playlists*. Même si nous ne pouvons pas affirmer avec certitude les raisons pour lesquelles il existe une telle différence en ce qui concerne les musiques passées, une hypothèse serait de dire que les utilisateurs du corpus de Lastfm passent moins de musiques, car ils possèdent un catalogue restreint de musiques (leur bibliothèque personnelle) et sont à l'origine des musiques qu'ils écoutent. Inversement, les utilisateurs du corpus de Deezer passent plus de musiques. Nous pensons que cela peut être dû à certains comportements tels que l'exploration des musiques du catalogue : l'utilisateur exploite un algorithme de recommandation ou passe de musique en musique dans une *playlist* jusqu'à trouver quelque chose qui lui corresponde.

En comparant ces nouveaux corpus (Tableau 4.1) par rapport aux corpus intégraux (Tableaux 2.1 et 2.3), nous pouvons constater que le nombre d'utilisateurs pour le nouveau corpus de Deezer a diminué. Cette diminution est due au fait que certains utilisateurs ne possèdent pas de sessions ne contenant que des items pour lesquels tous les attributs sont connus. Étant donné que les utilisateurs du corpus de Lastfm ont effectué de nombreuses consultations, le nombre d'utilisateurs reste inchangé. En revanche, de nombreux utilisateurs ont été retirés dans le nouveau corpus de Deezer.

À partir de ces nouveaux corpus de données, nous avons calculé la valeur moyenne de DR pour les débuts de sessions et la valeur moyenne pour les autres consultations. Dans notre cas, nous considérons que la diversité d'un début de session correspond à la valeur de DR de la première consultation d'une session. Pour cet exemple ainsi que pour l'ensemble des résultats présentés dans ce chapitre, sauf indication contraire, la DR de chaque consultation est calculée avec un historique de taille 3, un coefficient $\alpha = 10$ et aucun coefficient d'éloignement.

TABLE 4.2 – Statistiques des valeurs de DR en fonction de la position dans la session

	Lastfm	Deezer
DR moyenne toutes consultations	0,2392	0,2371
DR moyenne des débuts de session (DR_{DS})	0,2942	0,2800
DR moyenne des autres consultations (DR_{AC})	0,2323	0,2334
Différence $DR_{DS} - DR_{AC}$	0,055	0,0429

Nous pouvons constater dans le Tableau 4.2 qu'il existe une différence entre la moyenne de diversité des consultations des débuts de session et la moyenne de diversité des autres consultations. Dans le but de savoir si ces différences sont significatives, nous avons effectué un test de *Student* permettant de comparer les deux distributions. L'hypothèse nulle selon laquelle il n'existe pas de différence entre ces deux distributions est rejetée pour le corpus de Lastfm et le corpus de Deezer avec une *p-value* inférieure 2.2×10^{-16} . Étant donné ces résultats, nous pouvons affirmer que les différences de diversité entre les consultations des débuts de sessions et les

autres consultations sont statistiquement significatives. Autrement dit, la diversité en début de session est plus importante que la diversité des autres consultations. Étant donnée l’hypothèse n° 1 que nous avons formulée précédemment, nous avons décidé d’utiliser cette différence pour définir la valeur de τ . Ainsi, pour le corpus de Lastfm, τ vaut 0,055 et pour le corpus de Deezer τ vaut 0,0429. Même si nous avons défini cette valeur de τ *a posteriori*, cela ne nous garantit pas de retrouver les fins de session étant donné que ce paramètre ne constitue qu’une partie de la formule de détection.

Les valeurs de τ étant définies pour les deux corpus, nous avons pu détecter les changements de contextes implicites sur l’ensemble des deux corpus dont les résultats sont résumés dans le Tableau 4.3 ci-dessous.

TABLE 4.3 – Détection des contextes implicites avec le paramètre τ

	Lastfm	Deezer
# Consultations	183 374	249 042
# Contextes implicites	32 365	44 226
Moy. consul./Contexte	5,66	5,6311

Nous pouvons constater dans le Tableau 4.3 que les résultats obtenus sur les deux corpus sont quasiment identiques. Notre modèle détecte des contextes implicites ayant en moyenne une taille de 5,6 consultations. Étant donné que le nombre moyen de musiques par session est au moins supérieur à 8, nous pouvons en déduire qu’il peut exister plusieurs contextes implicites au sein d’une même session. Cela signifie qu’un SR basé sur l’ensemble des musiques écoutées depuis le début d’une session utilise des items pouvant être divers par rapport à ce que l’utilisateur souhaite écouter au moment où les recommandations sont proposées. La Figure 4.6 représente la répartition des tailles des contextes détectés pour le corpus de Lastfm. La première chose que nous pouvons constater est qu’il n’existe pas de contexte implicite de taille 1. Cela est dû à la formule de détection exigeant que la DR au temps t soit supérieure à la DR aux temps $t - 1$ et $t + 1$. Nous pouvons également constater qu’il existe certains contextes de très grande taille (>20) qui correspondent sans doute à des comportements de consultation très particuliers.

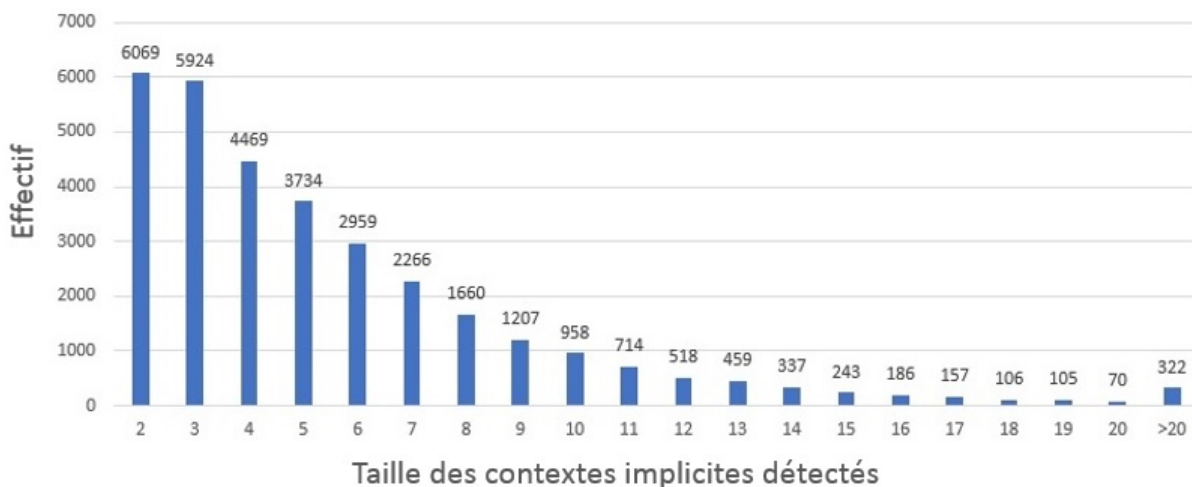


FIGURE 4.6 – Répartition des tailles de contextes sur le corpus de Lastfm

Le paramètre τ possède une grande influence sur les résultats de détection des contextes implicites. Si l'on ne change pas les paramètres intervenant dans le calcul de la DR, en modifiant la valeur de τ , nous rendons plus ou moins strict un changement de contexte implicite. Comme nous pouvons le constater dans le Tableau 4.4 ci-dessous, plus ce dernier est grand, moins nombreux seront les contextes implicites. En effet, une grande valeur de τ aura pour conséquence de ne définir un nouveau contexte que lorsque le niveau de diversité apporté par une consultation est très important.

TABLE 4.4 – Impact de la valeur de τ sur la détection des contextes implicites

	Lastfm			Deezer		
# Consultations	183 374			249 042		
$\tau = x \times \tau$	$x = 1$	$x = 2$	$x = 3$	$x = 1$	$x = 2$	$x = 3$
# Contextes implicites	32 365	17 973	9 090	44 226	23 439	11 368
Moy. consul./Contexte	5,66	10,20	20,17	5,63	10,62	21,90

Dans le chapitre précédent et notamment dans la Section 4.2.2, nous avons présenté différents paramètres qu'il est possible d'appliquer lors du calcul de la diversité. Dans la mesure où ces derniers impactent directement les valeurs de DR, ils impacteront également la détection des contextes implicites. Afin de voir quelles en sont les conséquences sur les détections de contextes implicites, nous avons fait varier ces paramètres tout en conservant une même valeur de τ . La valeur de τ est définie sur la base de l'ensemble des valeurs de DR qui sont elles-mêmes calculées selon certains paramètres. Recalculer τ en changeant certains paramètres aurait pour conséquence d'ajouter un facteur supplémentaire dans les résultats obtenus. Par exemple, modifier la valeur du coefficient α aura pour conséquence la modification de la valeur de τ et donc nous ne saurons pas si les changements sont dus à la nouvelle valeur de τ ou bien de α . Pour des raisons de place et afin d'éviter une redondance dans les résultats, nous ne présenterons pas ici l'impact de chacun des paramètres qui seront discutés dans la Section 4.2.2. Cependant, à titre d'exemple, nous avons fait varier le coefficient α (cf. Équation 3.7, p. 59) utilisé dans le calcul de la diversité des attributs numériques. Les résultats sont présentés dans le Tableau 4.5 ci-dessous :

TABLE 4.5 – Impact du coefficient α sur la détection des CI

	Lastfm				Deezer			
# Consultations	183 374				249 042			
Coefficient α	5	10	15	50	5	10	15	50
# Contextes implicites	30 798	32 365	33 087	34 345	41 344	44 226	45 469	46961
Moy. consul./Contexte	5,95	5,66	5,54	5,33	6,023	5,63	5,477	5,303

Le coefficient α utilisé dans le calcul de la DR impacte effectivement la détection des contextes implicites comme le montre le Tableau 4.5 ci-dessus. Plus ce dernier est grand et plus deux items doivent posséder des valeurs d'attributs numériques proches afin d'être jugés similaires. De ce fait, une valeur de α importante tend à augmenter les différences entre deux items et cela a pour conséquence d'accentuer les variations de diversité au cours du temps. En termes de détection des contextes implicites, l'augmentation de la valeur du paramètre α se traduit par une augmentation du nombre de contextes implicites détectés, étant donné que les variations de diversité sont plus soutenues et donc plus susceptibles d'être supérieures à la valeur de τ qui, elle, reste fixée et n'est pas mis à jour dans notre cas avec les différentes valeurs de α .

Ainsi, de la même manière que la modélisation de la diversité est sensible à certains paramètres, notre approche permettant de détecter des changements de contextes implicites l'est également. Il s'avère donc nécessaire de trouver une manière optimale permettant de configurer ces différents paramètres afin de maximiser la correspondance entre les changements de contextes implicites et les contextes explicites.

Nous avons montré que le modèle DANCE permet d'isoler des contextes implicites sur la base de l'évolution de la diversité. Cependant, nous n'avons à ce stade aucune preuve que les contextes implicites alors isolés peuvent constituer des alternatives aux contextes explicites. Nous avons donc décidé de comparer les changements de contextes implicites détectés avec les changements de sessions. S'il existe une relation entre un changement de contexte implicite et un changement de session pour de nombreuses sessions, cela confirmerait l'hypothèse selon laquelle les contextes implicites peuvent se substituer aux contextes explicites étant donnée l'Hypothèse 1 formulée vis-à-vis du lien entre sessions et contextes.

Contextes implicites et sessions

Nous allons à présent voir s'il existe effectivement un recouvrement entre les changements de contextes implicites que nous détectons et les changements de sessions. Il convient de préciser que ce recouvrement ne peut pas être de 100 % étant donné qu'un changement de session n'implique pas nécessairement un changement de contexte explicite. Il se peut qu'un utilisateur se trouve dans le même contexte d'une session à une autre, que les contextes explicites n'impactent pas ses préférences ou encore qu'il décide de poursuivre sa lecture là où il s'était arrêté.

En plus du recouvrement existant entre débuts de contextes implicites et débuts de sessions, nous nous intéressons également aux musiques passées et plus particulièrement au lien entre changement de contexte implicite et musique passée. En effet, nous souhaitons savoir si l'interruption d'une musique et le passage à la musique suivante peuvent s'expliquer par le fait que la musique passée apporte trop de diversité par rapport au contexte implicite de l'utilisateur. L'Algorithme 1 que nous avons utilisé pour mettre en lumière les correspondances entre les différents événements est présenté ci-dessous :

Les résultats obtenus avec cet algorithme et une valeur de τ telle que calculée dans le Tableau 4.2 sont présentés dans le Tableau 4.6 ci-dessous. **Étant donné que la première session de chacun des utilisateurs ne peut pas être détectée, nous avons soustrait au nombre de sessions total de chaque corpus le nombre d'utilisateurs de ce corpus.** Ainsi, 99 sessions ont été "enlevées" sur le corpus de Lastfm et 2 454 sessions ont été "enlevées" sur le corpus de Deezer.

TABLE 4.6 – Recouvrement entre les changements de sessions et les changements de contextes implicites

	Lastfm	Deezer
#Consultations	183 374	249 042
#Sessions	20 569	18 995
#Contextes implicites	32 365	44 226
Sessions retrouvées	11 992 (58,30 %)	11 554 (60,82 %)
#Musiques passées	5 052	75 426
#Musiques passées retrouvées	854 (16,90 %)	13 621 (18,05 %)

Nous pouvons constater que pour les deux corpus, quasiment 60 % des changements de ses-

Algorithme 1 Pseudo-code simplifié du modèle DANCE

```

sessions_détectées, musiques_passées_détectées, contextes_détectés = 0
k = taille de l'historique utilisé dans le calcul de DR
pour tout utilisateur u faire
  pour chaque consultation c faire
    calculer  $DR(c_t^u, C_{k,t}^u)$ 
     $DR'(c_t^u) = DR(c_t^u) - DR'(c_{t-1}^u)$ 
    si  $DR(c_{t-1}^u) < DR(c_t^u) > DR(c_{t+1}^u)$  et  $DR'(c_t^u) > \tau$  alors
      contextes_détectés ++
      si  $c_t^u$  ou  $c_{t-1}^u$  ou  $c_{t-2}^u$  est un début de session alors
        sessions_détectées ++
      fin si
      si  $c_t^u$  a été passé alors
        musiques_passées_détectées ++
      fin si
    fin pour
  fin pour
retourner sessions_détectées, musiques_passées_détectées, contextes_détectés

```

sions sont effectivement liés à des changements de contextes. Dans l'Algorithme 1 que nous avons présenté, nous regardons lorsqu'un nouveau contexte est détecté si un changement de session intervient au même moment, mais également lors des deux consultations précédentes. Nous avons fait ce choix car nous pensons qu'il est possible que certains changements de sessions ne puissent pas être détectés avec la première consultation d'une nouvelle session. Par exemple, il se peut qu'après avoir arrêté d'écouter de la musique pendant un certain laps de temps, l'utilisateur relance son lecteur de musique et reprenne où il s'était arrêté. Cela aura pour conséquence de ne pas apporter de diversité et ce n'est peut-être qu'après cette consultation ou la suivante que l'utilisateur adaptera ses consultations par rapport à son nouveau contexte explicite. Dans le Tableau 4.7 ci-dessous, nous présentons le nombre de détections effectuées en fonction du moment où la détection d'une nouvelle session est effectuée.

TABLE 4.7 – Position des détections

Moment de la détection	t - 2	t - 1	t	total
Nombre de détections Lastfm	2 882	2 590	7 546	13 018
Nombre de détections Deezer	2 910	3 378	6 198	12 486

Nous pouvons constater que la majorité des liens entre les contextes implicites et les sessions ont lieu au moment même où le changement de contexte est détecté. Cependant, dans certains cas, détecter un changement de session nécessite un ou deux pas de temps supplémentaire. Nous précisons que le nombre total de changements de sessions est légèrement supérieur au nombre présent dans le Tableau 4.6 car certaines sessions ont été détectées par deux changements de contextes différents. Par exemple, le nombre total de sessions détectées au travers des changements de CI pour le corpus de Lastfm est de 11 992. Or si on additionne les sessions détectées en fonction des moments, le total est de 13 018. Il y a donc 1 026 sessions qui ont été détectées plusieurs fois.

En ce qui concerne les musiques passées, plus de 16 % d'entre elles sont liées à un changement de contexte. Autrement dit, dans plus de 16 % des cas où les musiques sont passées, il est possible qu'un trop grand apport en diversité en soit la cause. Cette valeur constitue un résultat intéressant car il nous montre qu'il est dans certains cas possible d'expliquer la raison ayant poussé l'utilisateur à passer une musique. Néanmoins, il existe de nombreuses autres raisons pour lesquelles un utilisateur peut choisir de passer une musique : il n'apprécie pas cette musique, il apprécie cette musique mais ne souhaite pas l'écouter maintenant, cette musique lui est proposée trop souvent, *etc.* Le fait que nous n'expliquions que 16 % des musiques passées via la DR ne veut pas dire que notre modèle se trompe pour les 84 % restants. Ces 84 % de musiques passées peuvent avoir d'autres causes qu'une trop grande diversité, telles que celles que nous avons citées et nous pensons que l'analyse de l'évolution de la diversité reposant sur d'autres paramètres que les attributs s'avère une perspective intéressante. Par exemple, le rejet d'une musique appréciée par un utilisateur au prétexte qu'il l'a trop entendue récemment pourrait être détecté par une métrique mesurant la fréquence d'apparition des musiques recommandées (cette fréquence peut constituer un attribut sur lequel la DR peut être calculée). Il s'agirait en quelque sorte d'une mesure de diversité agrégée propre à chaque utilisateur (cf p.1.3.1). Ainsi, l'Hypothèse n° 2 que nous avons formulée page 80 concernant le lien entre une diversité importante et le rejet d'un item se confirme tout en sachant qu'un fort niveau de diversité n'est pas la seule explication.

Les résultats que nous venons de présenter nous ont permis de montrer qu'il est effectivement possible de détecter des changements de contextes explicites en se basant sur l'analyse de l'évolution temporelle de la diversité. Bien que cela constitue une preuve de concept du modèle DANCE, nous ne pouvons pas utiliser notre modèle dans sa configuration actuelle en situation réelle étant donné que nous utilisons des données obtenues *a posteriori*. Premièrement, la valeur du seuil de τ a été déterminée en comparant la diversité globale et la diversité existante en début de session. Deuxièmement dans notre formule de détection de changement de contextes implicites, nous utilisons la valeur de $DR(c_{t+1}^u)$. Or, cette information n'est pas disponible lorsque l'on travaille en temps réel. Nous allons donc dans la section suivante présenter une nouvelle formule de détection de contexte implicite ainsi que les résultats que nous avons obtenus en utilisant uniquement des informations disponibles en temps réel.

4.2.2 Performances en temps réel

Pour les raisons citées précédemment, nous avons adapté notre formule de détection afin de pouvoir tester notre approche dans des conditions correspondant à un fonctionnement en temps réel et sans utiliser un seuil τ déterminé *a posteriori*. Nous présenterons dans un premier temps la nouvelle formule de détection de changement de contexte implicite, puis dans un second temps nous présenterons et discuterons des résultats alors obtenus.

Détection des contextes implicites

La formule que nous avons adaptée et que nous présentons dans l'Équation 4.2 ci-dessous n'exploite plus la valeur de la DR' mais uniquement la valeur de $DR(c_t^u)$ qui est comparée par rapport à un seuil ϵ . De plus, nous n'exploitons que **l'ensemble des informations disponibles au pas de temps t** et de fait nous n'utilisons plus la valeur de $DR(c_{t+1}^u, C_{k,t+1}^u)$. Nous avons décidé d'utiliser le paramètre ϵ à la place de τ afin de montrer que les contextes implicites peuvent

être détectés de différentes manières. Alors que le paramètre ϵ correspond à un seuil minimum de diversité à atteindre pour considérer qu'un nouveau CI débute, le paramètre τ correspondait quant à lui à un seuil de variation minimum devant être atteint. Cependant, tout comme nous n'avons pas pu déterminer la valeur de τ , nous ne pouvons pas non plus connaître la valeur optimale de ϵ .

$$\begin{aligned} DR(c_{t-1}^u, C_{k,t-1}^u) <> \text{NaN} \text{ et } DR(c_t^u, C_{k,t}^u) <> \text{NaN} \quad (1) \\ DR(c_{t-1}^u, C_{k,t-1}^u) < DR(c_t^u, C_{k,t}^u) \text{ et } DR(c_t^u, C_{k,t}^u) > \epsilon \quad (2) \end{aligned} \quad (4.2)$$

De la même manière que pour la première formule de détection, nous avons décidé de ne pas statuer en faveur d'un changement de contexte si les valeurs de diversité utilisées dans la formule ne sont pas définies. La Figure 4.7 ci-dessous permet d'illustrer le fonctionnement de cette nouvelle méthode de détection de contextes implicites. Dans cet exemple, 3 consultations correspondent effectivement à un changement de contextes implicites, étant donné qu'elles respectent toutes les conditions de l'équation de détection 4.2. La consultation portant le label 4 ne possède pas une valeur de DR suffisante (c'est-à-dire supérieure à ϵ) pour constituer un changement de contexte implicite malgré le respect des autres conditions.

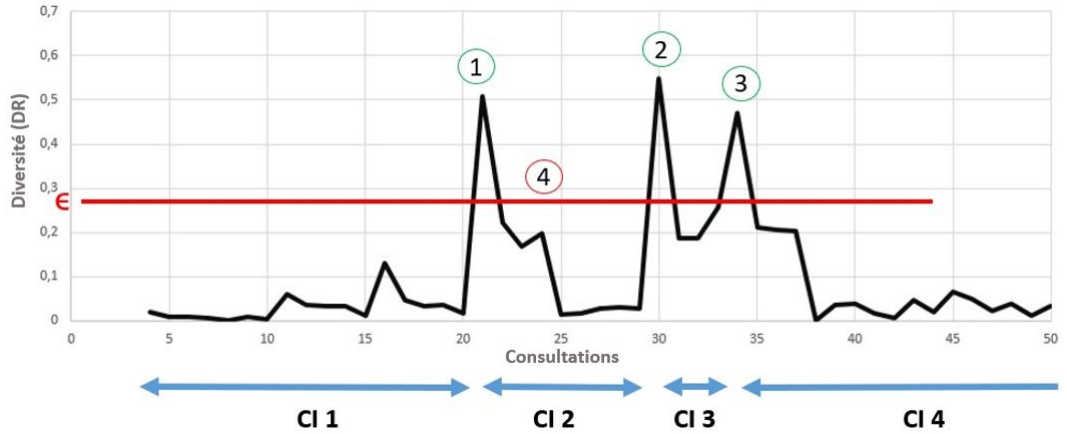


FIGURE 4.7 – Exemple de découpage

Afin de comparer les deux formules de détection, les résultats de détection des contextes implicites sont résumés dans le Tableau 4.8 ci-dessous. La valeur de ϵ est fixée pour chaque corpus à la valeur moyenne de la diversité observée sur l'ensemble des consultations. Nous rappelons ces valeurs présentes dans le Tableau 4.2 : $\epsilon = 0,2392$ pour le corpus de Lastfm et $\epsilon = 0,2371$ pour le corpus de Deezer.

TABLE 4.8 – Contextes implicites détectés avec la première et la seconde formule

	Lastfm		Deezer	
# Consultations	183 374		249 042	
Détection	τ	ϵ	τ	ϵ
# Contextes implicites	32 365	55 178	44 226	69 414
Moy. consul./Contexte	5,66	3,32	5,63	3.587
# Détections communes	28 609		35 748	

Premièrement, nous pouvons constater que le nombre de contextes détectés avec la seconde formule est plus important qu’avec la première formule. Étant donné que le nombre de contextes implicites augmente, le nombre de consultations moyen par contexte diminue également. Pour les raisons que nous avons déjà citées précédemment, il ne nous est pas possible de savoir si l’une des formules de détection est préférable à l’autre, sur la base d’un corpus hors-ligne tout du moins. Il serait toujours possible de mener une étude utilisateur sur ce point spécifique afin de déterminer la meilleure formule, voire même d’en dégager de nouvelles en fonction des retours des utilisateurs. De plus, chacune des formules possède un coefficient τ ou ϵ ayant un impact considérable sur le nombre de contextes implicites détectés. Ces coefficients n’impactent cependant que la sensibilité de chacune des méthodes de détection. En effet, la partie de la formule de détection faisant intervenir l’un ou l’autre de ces coefficients ne constitue pas l’unique condition selon laquelle sont détectés les changements de contextes implicites. Même si cette nouvelle méthode détecte plus des changements de contextes que la méthode précédente, les contextes détectés comprennent également les contextes détectés par la méthode précédente. En effet, 88,39 % des changements de contextes implicites du corpus du Lastfm et 80,83 % des changements de contextes implicites du corpus de Deezer sont détectés par les deux méthodes.

Si l’on regarde la répartition des tailles des contextes implicites présentée dans le Tableau 4.8 ci-dessous, nous pouvons constater qu’il existe une différence majeure avec la répartition obtenue avec la première méthode. Cette nouvelle formule de détection isole des contextes de taille 1 contrairement à la formule précédente. Ces contextes de taille 1 ne peuvent être obtenus que lorsque deux consultations successives respectent les conditions d’un changement de contexte implicite. Ces contextes témoignent du fait que l’utilisateur diversifie ses consultations et donc qu’il n’est pas possible que ces deux consultations appartiennent au même contexte. Autrement dit, ce sont des périodes pendant lesquelles les items ne partagent pas beaucoup d’attributs en commun. En termes de recommandation, cette information est très utile, car elle permet de savoir que l’utilisateur ne cherche pas quelque chose de similaire à ce qu’il a consulté récemment et donc qu’il va falloir prendre en compte ce besoin en diversité. Une analyse approfondie de la diversité apportée par les attributs pourrait alors être un moyen de comprendre ce que recherche l’utilisateur. Nous verrons notamment dans le Chapitre 6 quelles sont les stratégies de recommandation que nous pouvons adopter sur la base des contextes implicites.

Une autre explication serait de dire, comme nous l’avons mentionné précédemment dans la Section 3.3, qu’il existe des phases de transition pendant lesquelles un utilisateur ne sait pas encore vers quoi s’orienter et donc procède par essai-erreur. Ce comportement a d’ailleurs été mentionné dans les travaux de [Häubl and Murray, 2003] dans le cadre de recherches de produits sur un site de e-commerce.

En conclusion, il est effectivement possible d’isoler des contextes implicites dans des conditions identiques à des conditions réelles d’application. De la même manière que nous avons confirmé la pertinence des contextes implicites détectés *a posteriori*, nous allons maintenant voir s’il existe un recouvrement entre les changements de sessions et les contextes implicites détectés via cette nouvelle méthode.

Contextes implicites et sessions

Lorsque nous avons utilisé la première formule de détection, à chaque changement de contexte implicite nous avons également vérifié s’il existait un changement de session à t , $t - 1$ et $t - 2$. Étant donné que nous voulons éprouver notre modèle face à des conditions réelles d’application, les correspondances entre les changements de contextes implicites et les changements de sessions

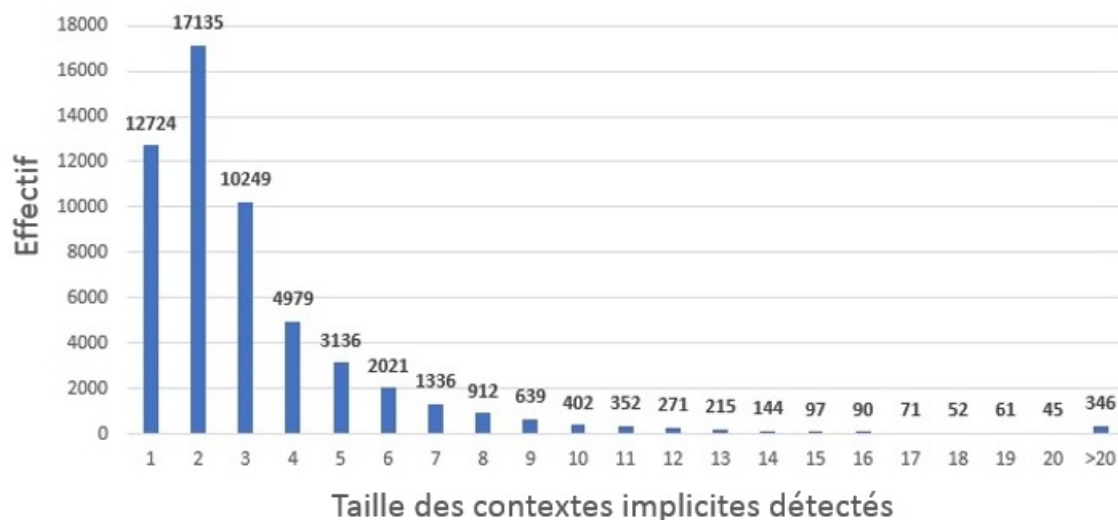


FIGURE 4.8 – Répartition des tailles de contextes implicites obtenue sur le corpus de Lastfm

sont à présent observées **uniquement à l’instant t** .

TABLE 4.9 – Recouvrement entre les changements de sessions et les changements de contextes implicites

	Lastfm	Deezer
# Sessions	20 569	18 995
# Contextes implicites	55 178	69 414
Sessions retrouvées	11 697(56,87 %)	9 530 (50,17 %)
Musiques passées	5 052	75 423
Musiques passées retrouvées	1 035(20,48 %)	20 238 (26,83 %)

Nous pouvons constater qu’une fois encore les changements de sessions sont effectivement liés à des changements de contextes implicites. Bien qu’étant inférieures aux performances obtenues avec la formule de détection 4.1, nous rappelons que nous n’utilisons plus la valeur de DR de la consultation suivante et les détections sont uniquement effectuées à l’instant t . Étant donné le nombre important de contextes détectés, nous avons voulu savoir si les sessions retrouvées à l’aide des changements de contextes étaient uniquement dues au hasard ou non. Pour ce faire, nous avons décidé de définir un changement de contexte implicite non plus par rapport à la diversité, mais selon une probabilité d’apparition afin de comparer les résultats alors obtenus par rapport aux vraies données. Cette probabilité est choisie afin d’obtenir un nombre sensiblement identique de contextes implicites. Ainsi, nous avons défini pour chacune des consultations, un changement de contexte selon la probabilité de $1/\delta$ où δ représente le nombre moyen de consultations par contexte présent dans le Tableau 4.9. Afin de limiter les biais dus à l’utilisation d’un processus aléatoire, nous avons effectué la moyenne des résultats obtenus suite à 20 exécutions de l’algorithme de détection (voir l’Algorithme 1) en redéfinissant à chaque exécution les contextes implicites aléatoirement. Les valeurs moyennes de ces 20 exécutions sont présentées dans le Tableau 4.10 ci-dessous.

TABLE 4.10 – Détection des sessions selon le hasard

	Lastfm	Deezer
δ	3,32	3,58
# Sessions	20 569	18 995
Moyenne #Contextes implicites	55 102,15	69 581,1
Écart-type #Contextes implicites	214,37	179,22
Moyenne Sessions retrouvées	30,11 %	27,79 %
Écart-type Sessions retrouvées	0,26 %	0,34 %

À partir des résultats présentés dans le Tableau 4.10, nous pouvons constater que le nombre de sessions retrouvées est largement inférieur lorsque nous définissons les changements de contextes aléatoirement. Quel que soit le corpus de données, l'utilisation de la diversité permet de retrouver quasiment 2 fois plus de changements de sessions. De plus, la valeur de δ utilisée n'est pas totalement aléatoire puisqu'elle est égale à la valeur moyenne des tailles des contextes implicites, ce qui explique pourquoi il est tout de même possible de retrouver presque 30 % des changements de sessions "aléatoirement". Ces résultats confirment que l'utilisation des contextes implicites définis à partir de la diversité s'avère une solution viable afin de déterminer des changements de contextes explicites.

Étant donné que nous n'avons pas pu valider notre approche de détection de changement de contextes implicites en temps réel, nous souhaitons à présent discuter de l'impact des différents paramètres de notre modèle. Nous rappelons que le modèle *DANCE* peut être paramétré à deux niveaux : dans le calcul de la DR ainsi qu'au niveau de la formule de détection de changement de contexte.

Concernant la formule de détection, le paramètre qu'il est possible de modifier est la valeur du coefficient ϵ . Comme nous pouvons le constater dans le Tableau 4.11, plus ce dernier est important et plus le nombre de contextes implicites détectés sera faible et réciproquement. Cela aura également comme conséquence de diminuer le nombre de sessions retrouvées. Il est important de rappeler que cette diminution ne signifie pas que notre approche est moins performante mais que les changements de sessions détectés sont marqués par une évolution plus importante de diversité. Autrement dit, il existe dans ce cas une différence de préférence accrue en termes de caractéristiques musicales et cette différence doit être considérée dans le processus de recommandation. Un SR possédant l'information selon laquelle l'utilisateur commence une nouvelle session pourra ne considérer que les musiques de cette session afin de fournir les recommandations. Cependant, nous avons montré qu'il existait plusieurs contextes implicites au sein d'une même session et prendre en compte ces changements de contextes implicites est tout aussi important que prendre en compte un changement de session.

TABLE 4.11 – Impact de la valeur de ϵ

# Consultations	Lastfm			Deezer		
	20 668			333		
$\epsilon = x * \epsilon$	$x = 1, 1$	$x = 1, 2$	$x = 1, 3$	$x = 1, 1$	$x = 1, 2$	$x = 1, 3$
# Contextes implicites	48 861	41 374	33 678	60 149	49 624	39 686
# Sessions retrouvées	51,84 %	44,86 %	37,10 %	46,44 %	41,06 %	34,75 %

En ce qui concerne le calcul de la diversité, nous avons présenté dans le chapitre précédent 3 paramètres sur lesquels il est possible d’agir. Le premier paramètre que nous avons présenté est le paramètre α qui intervient dans le calcul de la diversité des attributs numériques. Dans le Tableau 4.12 ci-dessous, nous présentons les résultats obtenus pour différentes valeurs de ce paramètre. Nous rappelons que le paramètre ϵ est, quant à lui, fixé à partir des valeurs présentées dans le Tableau 4.2 page 81. Nous pouvons constater que plus le coefficient α est important, et plus nombreux seront les contextes implicites détectés. De plus, le nombre de sessions retrouvées augmentera également avec ce paramètre. Cela est dû au fait qu’avec une valeur de *alpha* élevée, la formule de calcul de diversité est beaucoup plus sensible aux différences (même faibles) de valeurs d’attributs de deux items. Il en résulte alors de plus grandes variations de diversité, mais aussi une augmentation de la diversité moyenne. Le coefficient ϵ n’étant pas mis à jour, nous observons une augmentation des contextes implicites. Comme expliqué précédemment, nous ne pouvons pas déterminer quelle est la combinaison optimale des paramètres à partir des corpus hors ligne que nous possédons. Nous pouvons simplement discuter des conséquences de chacun des paramètres. Seuls des tests utilisateurs ou des métriques d’utilité de ces contextes nous permettront de définir ces paramètres.

TABLE 4.12 – Impact du coefficient exponentiel (α)

	Lastfm			Deezer		
# Sessions	20 569			18 995		
α	5	15	50	5	15	50
# Contextes implicites	46 412	59 851	71 050	53 490	79 669	99 636
Sessions retrouvées	50,79 %	59,57 %	63,87 %	44,06 %	52,84 %	57,84 %

Les deux autres paramètres qu’il est possible de faire varier sont le coefficient d’éloignement, ainsi que la taille de la fenêtre (cf. *supra*, Section 3.3.2, p. 65). Le seuil ϵ est également fixé selon la même condition que l’étude de l’impact du coefficient α (cf. 4.2.1, p. 83). Les résultats concernant leurs variations sont présentés respectivement dans les Tableaux 4.13 et 4.14 ci-dessous. Les différents coefficients d’éloignement ne semblent pas impacter de manière significative les résultats de détection de changements de sessions. Cependant, le nombre de contextes implicites détectés est plus important lorsque ce coefficient est décroissant, et ce, sur les deux corpus et pour les deux fonctions utilisées dans le coefficient d’éloignement (linéaire et quadratique). Par exemple, sur le corpus de Lastfm, l’utilisation d’un coefficient décroissant mène à la détection de 6 546 (fonction linéaire) et 10 060 (fonction quadratique) contextes implicites supplémentaires par rapport à l’utilisation d’un coefficient croissant. Même si un impact sur les sessions retrouvées n’est pas constaté, ce paramètre pourra tout de même être exploité lors du processus de recommandation. Par exemple, dans certains domaines applicatifs particuliers ou encore pour certains utilisateurs, la qualité de la modélisation du comportement utilisateur peut reposer sur le fait qu’elle tienne davantage compte du dernier item consulté ou à l’inverse des items consultés précédemment.

TABLE 4.13 – Impact du coefficient d'éloignement. LC = Linéaire Croissant, LD = Linéaire Décroissant, QC = Quadratique Croissant, QD = Quadratique Décroissant

# Sessions	Lastfm				Deezer			
	20 569				18 995			
Coeff. Éloignem.	LC	LD	QC	QD	LC	LD	QC	QD
# CI	52 982	59 528	52 885	62 945	69 359	73 778	69 608	77 389
Sess. retrou.	56,30 %	56,95 %	55,91 %	56,37 %	49,93 %	50,29 %	49,91 %	50,56 %

Comme nous pouvons le constater dans le Tableau 4.14, la taille de l'historique possède un impact mineur sur les performances de détection. Le fait que nous n'observions pas de différence notable s'explique par le fait que nous calculons la diversité en faisant la moyenne des diversités entre la cible et chacun des items de l'historique. Nous pensons cependant que le fait de faire varier la taille de l'historique possède sans doute un impact à l'échelle des utilisateurs étant donné que chaque utilisateur adopte son propre comportement. Un utilisateur très éclectique dans ses choix de musiques devra se voir attribuer par le système une taille d'historique faible afin de s'adapter au mieux à des changements fréquents d'attributs. Inversement, un utilisateur qui se diversifie très peu devra posséder un historique plus important afin que le système ne détecte pas des changements de contextes implicites qui ne seraient pas utiles pour l'utilisateur. En effet, lorsque la taille de l'historique est plus grande que 1, la diversité se calcule en procédant à la moyenne des diversités entre les items de l'historique et la cible. Ainsi, en utilisant un historique de grande taille, les variations de diversité sont moins prononcées par rapport au cas où l'historique est de petite taille.

TABLE 4.14 – Impact de la taille de l'historique

#Sessions	Lastfm			Deezer		
	20 569			18 995		
taille historique	2	4	5	2	4	5
# Contextes implicites	54 462	55 921	57 030	71 972	75 426	73 625
Sessions retrouvées	56,26 %	56,79 %	56,36 %	50,36 %	48,78 %	49,56 %

En résumé, les résultats que nous avons obtenus dans cette sous-section nous ont permis de montrer que les contextes implicites détectés par *DANCE* sont liés aux changements de sessions et qu'ils peuvent constituer une sérieuse alternative à l'utilisation des contextes explicites. Nous avons également montré que certains paramètres du modèle *DANCE* pouvaient avoir un impact sur ces détections de contextes implicites. Comme nous l'avons précisé au début de ce chapitre, nous n'avons utilisé que les sessions contenant l'intégralité des informations sur les attributs. Étant donné qu'à terme, notre modèle doit pouvoir fonctionner dans des conditions réelles, il est impératif de connaître sa robustesse face aux données manquantes. Nous allons donc à présent tester ses performances face à des données manquantes, ainsi que dans des conditions où les items consultés ne possèdent pas tous les mêmes attributs.

4.2.3 Robustesse du modèle DANCE

Avant de tester notre modèle sur le corpus initial possédant effectivement des données manquantes, que nous avons présenté dans le chapitre 2, nous avons souhaité quantifier l'impact que peuvent avoir les données manquantes sur les performances de détection de sessions. Nous

rappelons que, même si le but de notre modèle n’est pas à terme de retrouver des sessions, parvenir à retrouver des changements de sessions (que nous voyons comme des indices probables de changements de contextes) est la seule métrique de performance que nous possédons à l’heure actuelle.

Les corpus “restreints” (cf. *supra*, Section 4.1, p.80) possèdent l’avantage de ne contenir aucune information manquante. Nous avons choisi de dégrader ces corpus de manière contrôlée afin de connaître le pourcentage de données manquantes et donc d’en mesurer l’impact sur les performances. Par pourcentage d’informations manquantes, nous faisons référence au nombre d’attributs dont nous disposons à un moment donné par rapport au nombre d’attributs qui sont connus s’il ne manque aucune information. Si par exemple le corpus utilisé contient 50 musiques et que nous possédons les 16 attributs de chacune de ces musiques, nous possédons en tout $50 * 16 = 800$ informations. Avoir 1 % d’informations manquantes consiste à retirer aléatoirement les valeurs de 80 attributs, 2 % d’informations manquantes 160 attributs... Étant donné que certains attributs des musiques sont issus d’un même artiste, supprimer la valeur d’un attribut d’un artiste revient alors à supprimer autant d’informations que cet artiste possède de musiques. Par exemple, si nous supprimons la *popularity* de l’artiste **The Beatles** et que ce dernier possède 10 musiques dans notre corpus, nous considérons que nous avons supprimé 10 informations. Les performances de détection de sessions obtenues en fonction du pourcentage de données manquantes sont présentées dans la Figure 4.9 ci-dessous. Les paramètres utilisés dans le calcul de la diversité ainsi que ceux utilisés pour détecter les CI sont les mêmes que ceux ayant servis à produire les résultats du Tableau 4.9 présent à la page 89. Afin de limiter les biais dus à la suppression des informations, nous avons effectué 20 exécutions de l’algorithme consistant à dégrader les corpus et les résultats présentés correspondent à la moyenne des performances obtenues à chaque exécution.

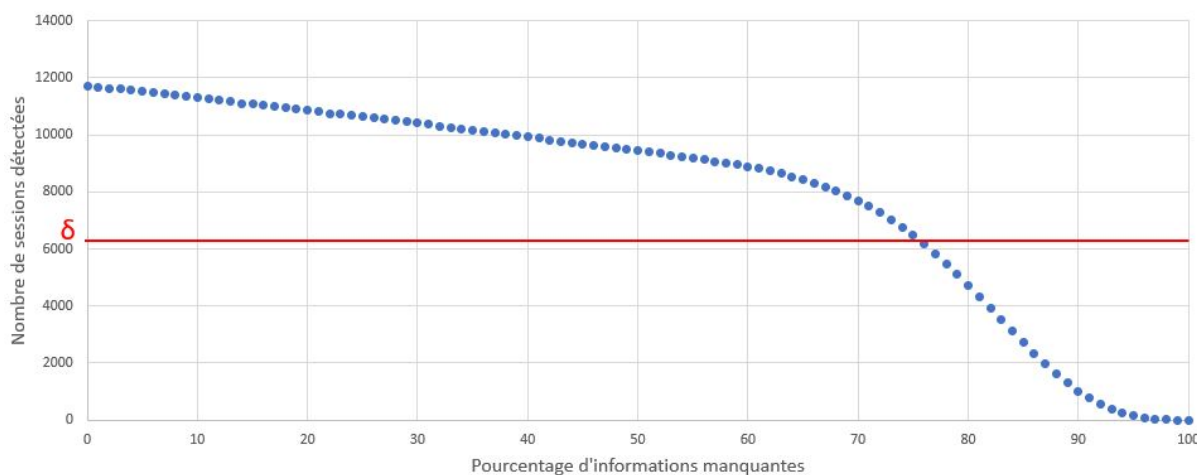


FIGURE 4.9 – Performance de détection en fonction du nombre de données manquantes sur le corpus de Lastfm

Les pourcentages en abscisse représentent le pourcentage d’informations enlevées par rapport à la quantité d’informations totale que nous possédons et les valeurs en ordonnée représentent le nombre de sessions détectées via les changements de contextes implicites. Nous pouvons constater que l’impact des informations manquantes reste modéré et linéaire tant que la quantité d’information manquante est inférieure à 60 %. Au-delà de ce seuil, les performances chutent beaucoup plus rapidement à chaque pourcentage supplémentaire d’information supprimée. Sur

ce graphique, nous avons également représenté par un trait horizontal le nombre de sessions retrouvées si l'on détermine les contextes à partir du hasard comme nous l'avons fait dans la Section 4.2.2. Nous constatons que les deux courbes se croisent lorsque le taux de données manquantes est proche de 75 % ce qui conforte l'idée que notre modèle est robuste face aux données manquantes. Des résultats similaires sont également obtenus sur le corpus de Deezer comme nous pouvons le constater sur la Figure 4.10 ci-dessous.

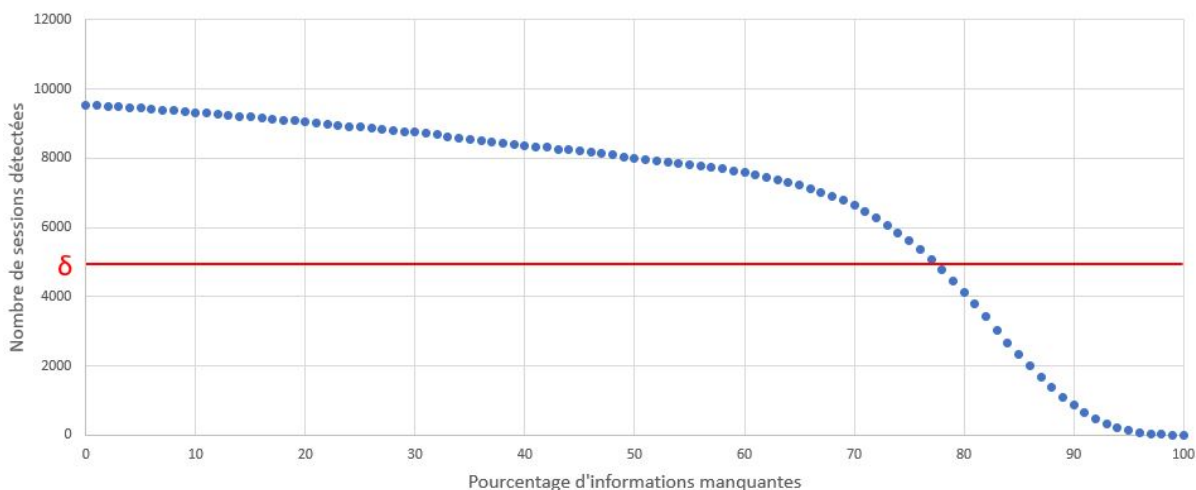


FIGURE 4.10 – Performance de détection en fonction du nombre de données manquantes sur le corpus de Lastfm

Étant donné les résultats encourageants que nous avons obtenus sur la dégradation contrôlée du corpus, nous avons décidé de tester les performances de DANCE sur le corpus initial. Dans le Tableau 2.6 présenté dans la Section 2.3 page 52, nous avons montré que les corpus de données obtenus possédaient des données manquantes. Nous avons également soustrait à chaque corpus un nombre identique de sessions au nombre d'utilisateurs pour prendre en compte le fait que la première session de chaque utilisateur ne peut pas être détectée étant donné qu'elle commence par la première musique écoutée par l'utilisateur (il n'y a aucun historique possible). Par exemple, si un utilisateur possède 3 sessions, nous ne pouvons tenter de détecter que le passage de la première session vers la deuxième, et le passage de la deuxième vers la troisième. Ce qui fait donc 2 sessions pouvant être détectées sur les 3 existantes.

TABLE 4.15 – Sessions et musiques passées retrouvées sur les jeux de données “bruts”

	Lastfm	Deezer
# Consultations	1 980 587	1 856 552
# Sessions	145 183	103 979
# Contextes implicites	547 423	574 550
# Sessions retrouvées	63 215(43,54 %)	38 739(37,25 %)
# Musiques passées	55 489	721 971
# Musiques passées retrouvées	11 304(20,37 %)	224 927 (31,15 %)

Lorsque nous avons présenté les corpus de données, nous avons présenté une métrique de couverture dite brute (*cf.* Tableau 2.6 page 52). Cette métrique correspond au pourcentage

d'informations que nous possédons réellement, car nous rappelons que le fait qu'une musique soit retrouvée sur l'API de Spotify ne nous assure pas d'obtenir l'ensemble de ses attributs. Ci-dessous, nous rappelons dans le Tableau 4.16 la couverture des attributs sur nos corpus. Le pourcentage d'informations réelles est supérieur au pourcentage de musiques retrouvées via l'API étant donné qu'un artiste possède plusieurs musiques et qu'il est alors possible de récupérer les informations sur l'artiste même si la musique n'a pas été retrouvée.

TABLE 4.16 – Statistiques de couverture des attributs en termes d'informations

	Lastfm	Deezer
#Musiques	168 084	204 434
#Informations théoriques	2 689 344	3 270 944
Musiques retrouvées via l'API	67,95 %	87,58 %
#Informations réelles	1 926 558	2 882 809
Informations réelles	71,63 %	88,13 %

Dans le Tableau 4.16, la ligne “#Informations théoriques” correspond à la quantité d'informations normalement obtenues si tous les attributs de toutes les musiques sont connus (ce qui est obtenu en calculant $\#Musiques \times \#Attributs$). Nous pouvons constater d'après ce tableau que le corpus de Lastfm possède 28,37 % d'informations manquantes et le corpus de Deezer 11,87 %. Si nous reportons sur les graphiques présents sur les Figures 4.9 et 4.10 la valeur de 28,37 % et 11,87 % d'informations manquantes en abscisse, le nombre de sessions retrouvées correspondant est d'environ 10 496 pour le corpus de Lastfm et d'environ 9 311 pour le corpus de Deezer. Cela correspond à une perte de 10,27 % de performance sachant que le nombre de sessions retrouvées sans information manquante est de 11 697 pour le corpus de Lastfm et une perte de 2,71 % de performance sachant que le nombre de sessions retrouvées sans information manquante est de 9 530 pour le corpus de Deezer. En suivant le même raisonnement pour tenter d'estimer les performances de détection sur les corpus complets de Lastfm et de Deezer en supposant que l'on peut retrouver les données manquantes, si nous ajoutons 10,27 % de sessions aux résultats obtenus dans le Tableau 4.15 pour le corpus de Lastfm, cela représente 70 436 sessions détectées soit en pourcentage 48,51 % et pour le corpus de Deezer cela représente 39 816 sessions manquantes soit en pourcentage 38,29 %. Nous pouvons constater que ce résultat de détection de sessions est en deçà de ce que nous avons obtenu avec les corpus ne contenant pas d'information manquante : 56,59 % pour le corpus de Lastfm et 50,17 % pour le corpus de Deezer. Cette différence de performances peut s'expliquer par le fait que les données réellement manquantes dans les corpus (contrairement au cas où nous avons simulé une dégradation) sont moins réparties sur l'ensemble des musiques. Lorsque nous avons dégradé nos corpus de données, nous avons supprimé des informations à l'échelle des attributs. Or, comme nous avons pu le voir sur le corpus de Lastfm, plus de 30 % des musiques n'ont pas été retrouvées par l'API de Spotify. En termes d'informations, cela signifie que nous possédons alors au mieux 3 attributs pour ces musiques si l'artiste a été retrouvé et aucun attribut le cas échéant. Cependant, le fait que, dans certains cas, l'ensemble des attributs ne soit pas connu n'est pas représentatif des autres domaines applicatifs et est spécifique à notre corpus. Dans les domaines applicatifs potentiels de notre modèle (e-éducation, e-commerce... cf. *supra* Section 2, p. 47), même si certains attributs peuvent être manquants, il est peu probable que tous le soient en même temps. Par exemple dans le domaine du e-commerce, chaque article possède au moins un prix et appartient au moins à une catégorie. Dans le secteur de l'e-éducation, chaque ressource en ligne possède au moins un domaine, une taille ainsi qu'un type (exercice, leçon...). De ce fait, il sera quasiment toujours

possible de calculer une valeur de diversité.

Dans le chapitre précédent, nous avons également expliqué que notre modèle peut être déployé sur un corpus contenant des items de natures différentes. Afin de tester les performances de notre modèle dans ce cas de figure, nous avons modifié notre corpus afin de simuler la présence d'items de différentes natures. Nous avons, à cet effet, utilisé deux paramètres qui sont $nbAttr$ et $nbCom$ afin de transformer notre corpus ne contenant qu'un seul type d'items (les musiques) en d'autres corpus contenant plusieurs types d'items. Le paramètre $nbAttr$ correspond au nombre d'attributs que possède l'ensemble des items des corpus créés et le paramètre $nbCom$ correspond au nombre d'attributs communs minimum entre 2 items. Par exemple, si les items du corpus initial contiennent 7 attributs ($A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$) et que nous souhaitons créer un corpus avec 3 types différents d'items selon les paramètres $nbAttr = 4$ et $nbCom = 2$, nous pouvons obtenir : $A_{type1} = \{a_1, a_4, a_6, a_7\}$, $A_{type2} = \{a_1, a_2, a_3, a_4\}$, et $A_{type3} = \{a_2, a_3, a_4, a_6\}$. Dans ce cas, les attributs communs entre chaque paire d'items sont : $A_{type1} \cap A_{type2} = \{a_1, a_4\}$, $A_{type1} \cap A_{type3} = \{a_4, a_6\}$, et $A_{type2} \cap A_{type3} = \{a_2, a_3, a_4\}$. Nous avons décidé de fixer à 4 le nombre d'items différents dans les nouveaux corpus de données que nous allons créer. Ce choix est motivé par le fait que, hormis les sites de e-commerce pour lesquels les items différents sont presque aussi nombreux que le nombre de produits, nous avons constaté qu'il existe généralement 4 items principaux dans les catalogues des services en ligne. Par exemple, les principaux items qu'il est possible de consulter sur un site de musique en ligne sont une musique, un artiste, un album ainsi qu'un label/producteur. Sur les réseaux sociaux professionnels, il est possible de consulter la page d'un utilisateur, la page d'une entreprise, la page d'un établissement de formation et une offre d'emploi. Bien que nous ayons fixé le nombre d'items à 4, il est possible de tester les performances de notre modèle sur le nombre d'items désiré.

Afin de limiter les biais dus à l'initialisation de certaines valeurs, nous avons choisi de sélectionner au hasard les attributs utilisés dans $nbAttr$ et $nbCom$ et d'exécuter 20 fois nos algorithmes.

Nous précisons que la valeur de $nbCom$ doit toujours être inférieure à la valeur de $nbAttr$. Si $nbCom = nbAttr$, alors tous les types d'items partageront les mêmes attributs. Or, nous souhaitons utiliser des items ayant des attributs différents et non pas des items identiques partageant un sous-ensemble des attributs des items initiaux. Nous pouvons constater d'après le Tableau 4.17 ci-dessus que, même si les paramètres $nbAttr$ et $nbCom$ sont petits, le nombre de sessions retrouvées est supérieur à ce qu'il est possible de trouver en déterminant les contextes au hasard (qui est 30,71 % de sessions retrouvées pour Lastfm). Au fur et à mesure que ces deux paramètres augmentent, le pourcentage de sessions retrouvées augmente également. Le pourcentage de détection augmentent jusqu'à atteindre son maximum lorsque les items contiennent l'ensemble des attributs. Ainsi, même si nous avons réalisé une simulation, à défaut de disposer d'un corpus contenant différents types d'items pour lesquels nous connaissons les attributs, ces résultats constituent une preuve de concept de la capacité de notre modèle à pouvoir s'adapter à différents types d'items.

Nous pouvons également constater que le pourcentage de sessions détectées n'est pas uniquement dépendant du nombre d'attributs. Pour une même configuration des paramètres $nbCom$ et $nbAttr$, nous constatons que les écarts-types des sessions détectées sont relativement importants. Cela signifie que certaines combinaisons d'attributs apportent plus de diversité que les autres. Autrement dit, il est possible que certains attributs soient plus pertinents que d'autres afin de détecter des changements de contextes implicites. Nous étudierons notamment ce cas dans la section suivante.

Paramètres		% Sessions retrouvées		Contextes implicites	
<i>nbAtt</i>	<i>nbCom</i>	Moy.	σ	Moy.	σ
3	2	36,68	14,91	47971,4	16480,92
4	2	37,16	14,99	51527,00	19328,10
4	3	44,49	16,52	52960,40	15954,07
5	2	44,34	12,57	57655,20	14747,31
5	3	39,86	12,85	52195,50	15779,24
5	4	36,17	16,44	45671,10	19901,01
6	2	46,19	7,66	58929,65	9929,91
6	3	45,40	10,36	57712,65	11623,69
6	4	38,40	15,84	48665,55	18392,33
6	5	43,22	14,58	48382,20	15775,03
7	2	49,40	7,42	57594,95	8395,18
7	3	46,10	10,08	55416,95	11388,80
7	4	46,79	10,12	54804,40	12250,08
7	5	51,31	10,34	60002,25	10209,51
...
10	2	50,61	5,25	57620,95	4823,01
10	3	49,64	5,49	58012,10	7475,16
10	4	49,20	7,56	54157,25	7943,30
10	5	46,36	9,15	51594,95	9157,97
10	6	46,06	8,97	51251,40	9203,95
10	7	52,74	8,79	57267,70	8716,89
...
15	2	56,14	2,42	55817,35	1740,56
15	3	55,25	2,33	55484,30	1913,82
15	4	55,60	2,63	55484,30	2409,13
15	5	54,67	3,28	55363,50	2333,070
...
16	-	56,87	0	55178,00	0

TABLE 4.17 – Performances de détections pour différents types d'items sur le corpus de Lastfm

4.2.4 Amélioration des performances

Jusqu'à présent, chaque attribut possédait le même poids dans le calcul de la DR qui, nous le rappelons, correspond à la DR moyenne de l'ensemble des attributs. Dans le calcul de cette moyenne de l'Équation 3.3, les coefficients β_a pondérant chacun des attributs étaient fixés jusqu'à présent à 1. Compte tenu des résultats obtenus dans la section précédente et du caractère spécifique de chaque attribut, il est possible que certains attributs soient plus déterminants que d'autres afin de révéler un changement de contexte implicite qui soit "utile". La notion d'utilité fait ici référence à l'utilisation des contextes implicites exploitables pour la recommandation. Par exemple, à valeurs égales, la diversité apportée par l'attribut "énergie" est peut-être plus révélatrice d'un réel changement de besoin de l'utilisateur que la diversité apportée par la durée. Si tel est le cas, le poids de l'attribut "énergie" devra être plus important que le poids de l'attribut "durée" dans l'Équation 3.2.

Si l'on considère que les poids des attributs sont des nombres à deux décimales pris dans l'intervalle $[0; 1]$ et que chaque attribut possède un poids lui étant propre, tester l'ensemble des possibilités d'attributions des poids reviendrait à tester 101^{16} combinaisons (101 valeurs possibles de l'intervalle attribuées à 16 attributs). Étant donné cet ordre de grandeur, il n'est évidemment pas concevable de tester toutes les configurations possibles. Ce problème est un problème d'exploration d'un espace de très grande taille et l'utilisation d'un algorithme génétique semble être une solution adéquate. En effet, les algorithmes génétiques ont été mis au point dans les années soixante et ont trouvé leur intérêt dans la résolution de problèmes possédant de très nombreuses combinaisons parmi lesquelles les meilleures d'entre elles doivent être identifiées. Un problème ayant par exemple largement bénéficié des algorithmes génétiques est celui du voyageur de commerce (ou du commis voyageur) [Shi and Zhang, 2017]. En utilisant un algorithme génétique, nous ne cherchons pas à voir si nous parvenons à trouver la meilleure configuration possible des poids. L'utilisation d'un algorithme génétique nous permettra de balayer un large spectre de combinaisons et nous pourrons alors savoir si certaines configurations sont meilleures que d'autres, *i.e.* si certains attributs sont plus déterminants que d'autres.

Les algorithmes génétiques sont utilisés dans de nombreux problèmes et constituent également un champ de recherche à part entière. Il existe de nombreuses variantes en termes de mécanismes utilisés [Mallipeddi et al., 2011], bien que tous les algorithmes génétiques partagent un principe évolutionniste commun : initialisation, évaluation, reproduction. Dans notre cas, nous avons décidé d'utiliser les mécanismes suivants :

- **Initialisation de la population** : la population initiale comporte 500 chromosomes dont les seize gènes ont été choisis aléatoirement entre 0 et 1 (ce qui correspond aux poids des attributs pour le calcul de la DR) ;
- **Évaluation des chromosomes (fonction de *fitness*)** : le score d'un chromosome est déterminé par le nombre de sessions retrouvées. Un chromosome est alors plus performant qu'un autre s'il parvient à retrouver plus de sessions ;
- **Formation des couples** : les couples sont formés aléatoirement parmi la population de chromosomes et chaque couple possède un score correspondant à la somme des scores des chromosomes le composant. Ensuite, en simulant une loi de Poisson avec une espérance⁶⁸ de 2,12, nous avons généré une liste du nombre de fils devant être créés par chaque couple.

68. Une valeur de l'espérance de 2,12 a été choisie selon la loi des grands nombres de manière à perpétuer la population sans pour autant qu'elle grandisse trop rapidement.

Plus le score du couple est important et plus le nombre de fils qu'il devra générer sera grand. Ce mécanisme permet de privilégier au sein de la population, les gènes permettant d'obtenir de bons scores sans pour autant tomber dans une forme d'eugénisme dont le principe peut s'avérer néfaste pour la convergence de l'algorithme ;

- **Reproduction** : lors du processus de reproduction, les deux chromosomes des couples formés précédemment vont générer des enfants en s'échangeant une partie de leurs gènes à hauteur de 70 %. Chaque nouveau chromosome contient 70 % des gènes de l'un des parents du couple et 30 % des gènes de l'autre parent. Les parents ne seront pas inclus dans la génération suivante ;
- **Mutations** : chaque nouveau chromosome peut muter avec une probabilité de $\frac{1}{16}$ (1/nombre de gènes). Si un chromosome mute, alors un de ses gènes sera généré aléatoirement.

Dans un premier temps, nous avons utilisé notre algorithme génétique sur l'ensemble des utilisateurs, c'est-à-dire que les chromosomes sont évalués selon les performances obtenues pour tous les utilisateurs confondus. Dans un second temps, notre algorithme génétique sera appliqué pour chacun des utilisateurs.

Poids des attributs identiques pour tous les utilisateurs

Les calculs ont été effectués sur le corpus de Lastfm avec les paramètres suivants : ϵ est fixé à la valeur moyenne présentée dans le Tableau 4.2, $\alpha = 10$, l'historique est de 3 et aucun coefficient d'alignement n'est utilisé. Ces paramètres sont identiques à ceux utilisés pour obtenir les résultats présentés dans le Tableau 4.9 p. 89.

Avant toute chose, nous avons dû définir le critère d'arrêt de notre algorithme génétique. Nous avons exécuté une seule fois notre algorithme génétique sur 100 générations afin de savoir si le critère d'arrêt peut être défini en fonction du nombre de générations. Les performances de détection obtenues en fonction du nombre de générations sont présentées dans la Figure 4.11 ci-dessous.

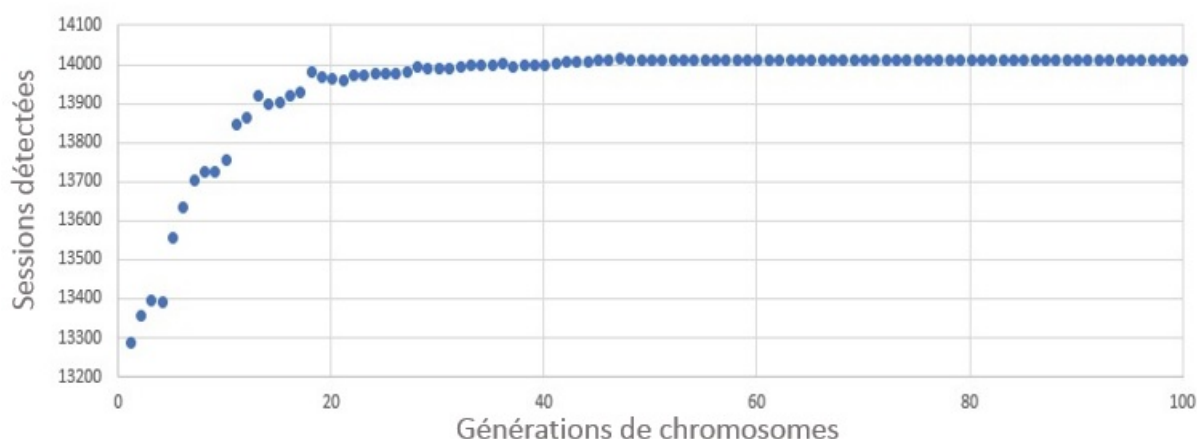


FIGURE 4.11 – Performances et nombre de générations de l'algorithme génétique

Nous pouvons constater sur ce graphique que le gain de performances se manifeste en grande partie pendant les 30 premières générations avant d'atteindre un point de convergence à partir de 50 générations. Pour cette raison, nous avons choisi de fixer notre condition d'arrêt à 50 générations par exécution.

Dans le but de limiter les biais dus à l’initialisation des chromosomes, nous avons effectué 20 exécutions de notre algorithme génétique dont nous reportons les performances dans le Tableau 4.18 ci-dessous.

TABLE 4.18 – Résultats de détection des sessions suite à l’utilisation d’un algorithme génétique pour tous les utilisateurs et après 20 exécutions

	Lastfm	Deezer
# Sessions	20 569	18 995
# Contextes implicites	74 318,6	103 083,25
Sessions retrouvées	13 973,70(67,93 %)	11 552,8 (60,82 %)

Nous pouvons constater que l’utilisation de poids d’attributs différents permet effectivement d’améliorer les performances de détection de notre modèle si nous nous comparons aux résultats (obtenus sans configuration de poids particulière) présentés dans le Tableau 4.9 p. 89. En effet, pour le corpus de Lastfm nous trouvons en moyenne 11,06 % de sessions en plus et pour le corpus de Deezer, nous trouvons en moyenne 10,65 % de sessions supplémentaires. Ces résultats confirment l’hypothèse selon laquelle certains attributs sont plus déterminants que d’autres afin de détecter des changements de contextes implicites correspondant également à des changements de contextes explicites. Il est également intéressant de noter qu’il ne semble pas possible d’augmenter le nombre de sessions retrouvées sans augmenter le nombre de contextes implicites. Cependant, les changements de sessions sont les seuls moyens que nous possédons pour évaluer notre approche. Être capable de quantifier l’utilité des contextes implicites ou encore de confronter ces derniers au jugement des utilisateurs constitue une des nombreuses perspectives que nous envisageons. En possédant d’autres critères d’évaluation tels que la compréhension des besoins utilisateurs ou encore la pertinence des recommandations générées, il serait possible d’utiliser ce processus d’optimisation afin de déterminer les poids les plus adéquats.

Nous avons voulu également vérifier la stabilité de ce paramétrage à la suite de ces multiples exécutions. Pour ce faire, nous avons regardé pour plusieurs exécutions de notre algorithme génétique, les rangs des attributs obtenus à chaque exécution. Les rangs correspondent à la position des poids dans un classement par ordre décroissant. Plus un poids est important, plus son rang est faible. Le poids ayant la valeur la plus importante est classé premier et le poids faible est classé dernier. Nous avons choisi de travailler sur les rangs car ils sont plus facilement comparables que les valeurs des poids des attributs. À l’issue de ces 20 exécutions, nous avons calculé la moyenne ainsi que l’écart-type de chacun des rangs des poids des attributs. Les résultats obtenus sont reportés dans le Tableau 4.19 ci-dessous.

TABLE 4.19 – Classement du poids des attributs obtenus avec l’utilisation d’un algorithme génétique sur l’ensemble des utilisateurs et après 20 exécutions

Ordre	Lastfm			Deezer		
	Nom	\bar{x}	σ	Nom	\bar{x}	σ
1	Clé	1,2	0,4	Clé	1,05	0,218
2	Genre	1,85	0,477	Genre	2,3	0,557
3	Popularité	3,3	0,714	Popularité	2,75	0,698
4	Valence	4,35	0,653	Valence	5,25	1,728
5	Liveness	5	1,816	Acousticness	6,7	3,018
6	Mode	6,5	1,532	Speechiness	7	2,387
7	Danceability	10,15	3,07	Énergie	8,15	3,825
8	Acousticness	10,7	3,18	Mode	9	3,912
9	Speechiness	10,95	2,77	Liveness	9,3	2,685
10	Énergie	11,3	2,81	Danceability	10,3	2,830
11	Sign. rythm.	11,45	2,62	Sign. rythm.	11,4	2,709
12	Artistes. Sim.	11,45	2,84	Instrumentalness	11,75	2,467
13	Instrumentalness	11,55	3,12	Durée	12,5	2,377
14	Durée	11,8	3,50	Artistes. Sim.	12,65	2,414
15	Loudness	12,15	3,24	Tempo	12,85	2,455
16	Tempo	12,3	2,39	Loudness	13,05	2,479

Dans le Tableau 4.19, nous pouvons constater que sur les 20 exécutions effectuées, les 4 attributs possédant les poids les plus forts, et donc les mieux classés sont les mêmes. De faibles écarts-types sont observés pour ces 4 premiers attributs, et des écarts-types plus importants sont observés sur les autres attributs. En effet, la “clé”, le “genre”, la “popularité” et la “valence” possèdent un écart-type inférieur à 0,7 sur le corpus de Lastfm et 1,8 sur le corpus de Deezer. Avant toute autre interprétation des résultats présents dans le Tableau 4.19, il convient de discuter de l’attribut “clé”. En effet nous ne savons pas à ce stade si l’attribut “clé” constitue un attribut pertinent dans la détection de changements de contextes implicites ou si ce dernier constitue un attribut qui apporte nécessairement une quantité importante de diversité. Lorsque nous avons présenté les attributs dans la Section 2.3, nous avons vu que cet attribut est de type qualitatif. La formule de diversité utilisée pour ce type d’attribut est binaire, soit deux musiques possèdent la même “clé” et ont de fait une diversité de 0, soit elles possèdent une “clé” différente et ont une diversité de 1. Or, contrairement au “mode” qui ne possède que deux modalités (mineure ou majeure), l’attribut “clé” possède 12 modalités différentes et nous ne savons pas si le passage d’une clé à l’autre est perçu par les utilisateurs. Autrement dit, bien que l’évolution en diversité de cet attribut constitue une information qu’il est possible d’exploiter afin de détecter des changements de contextes, nous ne pouvons pas affirmer que ces contextes implicites feront sens chez l’utilisateur (qui peut ne pas percevoir les variations de cet attribut). En ce qui concerne les autres attributs pour lesquels des niveaux de diversité intermédiaires (autres que 0 ou 1) peuvent être déterminés, les résultats montrent que certains attributs sont effectivement plus importants que d’autres. Par exemple, la valence permettant de quantifier la joie ou la tristesse dégagée par une musique (*cf. supra*, définition de la valence, page 51) est très déterminante dans le fait de retrouver des changements de sessions. Le fait que l’humeur dégagée par une musique soit parmi les attributs les plus déterminants sachant que l’état mental est une dimension très importante du contexte explicite d’un utilisateur confirme notre Hypothèse n° 1 sur le lien entre changement

TABLE 4.20 – Résultats des détections de sessions suite à l’utilisation d’un algorithme génétique pour chaque utilisateur et après 20 exécutions

	Lastfm	Deezer
# Sessions	20 569	18 995
# Contextes implicites	73 190,6	97 727,4
Sessions retrouvées	14 697,30 (71,14 %)	14 181,95 (74,66 %)

de contexte explicite et changement de session.

En conclusion, l’utilisation d’un algorithme génétique s’avère effectivement une solution pertinente afin de paramétrer notre modèle *DANCE*. Nous allons à présent nous intéresser aux performances de ce même algorithme génétique à l’échelle des utilisateurs.

Poids des attributs propres à chaque utilisateur

Contrairement à l’expérience précédente, nous avons appliqué notre algorithme génétique à l’échelle des utilisateurs, c’est-à-dire que les poids des attributs sont définis à partir des performances de détection des sessions pour chaque utilisateur individuellement. Nous avons également effectué 20 exécutions de l’algorithme génétique pour chacun des utilisateurs et les résultats sont reportés dans le Tableau 4.20 ci-dessous :

Nous pouvons constater que les performances de détection sont également améliorées puisque nous parvenons à détecter 71,14 % des sessions sur le corpus de Lastfm contre 67,93 % dans la version précédente et 74,66 % des sessions sur le corpus de Deezer contre 60,82 % dans la version précédente utilisant un algorithme génétique non spécifique à chaque utilisateur (*cf.* Tableau 4.18). Afin de comprendre cette amélioration, nous avons souhaité savoir si les rangs des poids des attributs étaient les mêmes que pour la configuration précédente. Pour ce faire, nous avons tiré au sort deux utilisateurs du corpus de Lastfm pour lesquels nous présentons, dans le Tableau 4.21 ci-dessous, les rangs des attributs obtenus.

Bien que l’attribut “clé” soit toujours l’attribut ayant le poids le plus important, nous pouvons constater qu’il existe effectivement une différence entre les rangs obtenus pour les deux utilisateurs. Par exemple, alors que pour l’utilisateur ayant l’id 87 le genre et l’*acousticness* sont classés respectivement 2^{ème} et 3^{ème}, ces mêmes attributs sont classés 4^{ème} et 8^{ème} pour l’autre utilisateur.

Les résultats que nous venons de présenter dans cette section nous ont permis de confirmer que certains attributs sont plus déterminants que d’autres pour détecter des changements de contextes implicites liés à des changements de contextes explicites (fins de sessions). Nous avons également pu constater que le paramétrage des poids des attributs est relativement stable après de nombreuses exécutions de l’algorithme génétique. En effet, les faibles écarts-types obtenus pour les attributs les plus importants confirment que leurs poids ne sont pas dus à l’initialisation des paramètres de l’algorithme génétique. Enfin nous avons pu constater que ce paramétrage était encore plus efficace à l’échelle des utilisateurs. Ce qui signifie que bien que le lien entre diversité et contexte soit partagé par l’ensemble des utilisateurs, la configuration à utiliser en termes de poids des attributs est quant à elle propre à chaque utilisateur.

TABLE 4.21 – Poids des attributs pour deux utilisateurs du corpus de Lastfm

Ordre	Utilisateur 1 (id 87)			Utilisateur 2 (id 29)		
	Attribut	\bar{x}	σ	Attribut	\bar{x}	σ
1	Clé	1,6	0,73	Clé	1,35	0,73
2	Genre	2,15	1,05	Valence	2,7	1,31
3	Acousticness	3,3	1,19	Danceability	4,85	1,74
4	Mode	4,45	1,91	Genre	5,05	1,69
5	Énergie	4,45	1,50	Mode	5,15	1,06
6	Danceability	7,5	2,99	Énergie	5,4	2,85
7	Sign. rythm	8,55	3,01	Instrumentalness	6,7	4,04
8	Tempo	10,65	3,05	Acousticness	8,45	2,40
9	Valence	10,8	3,44	Popularity	9,3	3,81
10	Speechiness	11,15	3,13	Sign. rythm.	11,9	1,95
11	Intrumentalness	11,20	2,45	Speechiness	12,1	2,49
12	Artistes sim.	11,45	2,73	Liveness	12,15	2,63
13	Popularity	12,00	3,29	Loudness	12,25	2,49
14	Durée	12,05	2,50	Durée	12,5	2,54
15	Loudness	12,05	3,01	Tempo	12,65	2,33
16	Liveness	12,65	2,46	Artistes sim.	13,5	2,5

4.3 Discussion

Dans ce chapitre, nous avons vu que l’approche de modélisation de la diversité au cours du temps introduite dans le premier chapitre peut être utilisée afin de détecter une nouvelle famille de contextes : les contextes implicites. Contrairement aux contextes explicites basés sur des informations concernant l’utilisateur (et son environnement), les contextes implicites sont uniquement déterminés sur la base de l’évolution des attributs des items consultés dans un historique récent de petite taille. Dans un premier temps, nous avons présenté le modèle DANCE permettant d’isoler ces contextes implicites dans une séquence de consultations. Le modèle DANCE exploite l’apport en diversité de l’item couramment consulté par rapport aux items précédents afin de déterminer si cet item appartient au contexte courant ou s’il constitue le premier item d’un nouveau contexte. Afin de démontrer la pertinence de notre approche, nous avons émis l’hypothèse qu’un changement de contexte explicite avait une forte probabilité de se produire entre deux sessions d’un utilisateur. Nous avons alors constaté dans un second temps que plus de 50 % des changements de sessions, et donc, par extension des changements de contextes explicites, coïncidaient avec les contextes implicites détectés. Nous avons également démontré que notre modèle peut être paramétré à de nombreux niveaux et que la variation de ces paramètres permet de modifier la manière dont sont détectés les contextes implicites. Parmi ces paramètres, nous avons notamment démontré que les poids des attributs intervenant dans le calcul de la diversité pouvaient être adaptés afin de spécialiser notre modèle. De plus, ces poids peuvent également permettre de personnaliser la détection des changements de contextes implicites en fonction du comportement propre à chaque utilisateur. En outre, nous avons montré que le modèle DANCE est robuste face à des données manquantes et qu’il peut être utilisé dans des domaines où les items sont différents. Nous avons montré qu’il était possible de détecter des changements de contextes implicites de différentes manières. Même si nous avons obtenu de bonnes performances, nous sommes conscients que les formules de détection présentées ne sont que les prémises de nombreuses perspectives.

Nous n'avons utilisé que la valeur de la diversité globale afin de déterminer un changement de contexte. Mettre au point des méthodes de détection exploitant indépendamment l'évolution en diversité de chacun des attributs constitue par exemple une première perspective. Une seconde perspective serait de développer des formules de détection probabiliste pour lesquelles un changement de contexte n'aurait lieu que si un ensemble de critères est rempli. Une troisième perspective serait d'étudier le comportement à adopter lorsqu'un contexte implicite est détecté. Jusqu'à présent, nous avons calculé la diversité sur l'ensemble des consultations des utilisateurs et de manière ininterrompue entre leur première et leur dernière consultation. Lorsqu'un changement de contexte était détecté, nous avons continué de faire "avancer" la cible et l'historique d'un pas de temps comme nous l'avons montré sur la Figure 3.1 p.56. Or, lorsqu'un contexte implicite est détecté, cela signifie que les caractéristiques de l'item courant sont différentes des caractéristiques des items contenus dans l'historique. Comme nous n'avons pas rompu l'historique de taille k , la diversité de l'item suivant est alors calculée sur la base d'un historique composé du premier item du contexte courant ainsi que des $k - 1$ items du contexte précédent. Il serait donc intéressant d'observer l'impact de cette fenêtre empiétant sur le contexte précédent et d'étudier de quelle manière il faudrait adapter la taille de la fenêtre après chaque début de contexte. Il serait par exemple, possible lors d'un nouveau contexte, de définir la taille de l'historique à 1 et de faire augmenter ce dernier jusqu'à la taille souhaitée si aucun changement de contexte implicite n'est observé entre temps et de prendre en compte cela dans la formule de détection. Enfin, une dernière perspective serait de s'intéresser au cas où la diversité est suffisamment instable sur plusieurs consultations consécutives au point de déclencher un changement de contexte implicite à chaque consultation. Nous pouvons imaginer que ces comportements se produisent lorsqu'un utilisateur recherche quelque chose en particulier. Ne sachant pas encore vers quoi il souhaite s'orienter, il sélectionne plusieurs items ayant des caractéristiques diverses afin de savoir ce qui correspond le mieux à ses besoins ou ses envies. Dans ce cas, il faudrait pouvoir détecter cette séquence particulière afin de la considérer comme un seul contexte implicite de transition. Ainsi, comme nous venons de le voir, la détection des changements de contextes offre de nombreuses perspectives.

L'ensemble du potentiel qu'offrent les contextes implicites ne peut pas être résumé par leur capacité à pouvoir déterminer les moments où interviennent des changements de sessions. Le concept des contextes implicites est né de plusieurs constats que nous avons fait dans l'état de l'art et pourrait permettre de proposer des éléments de réponses aux questions encore ouvertes. Premièrement, les contextes implicites sont basés sur le premier modèle d'analyse de l'évolution du niveau de diversité au cours du temps. Ainsi, sans même parler de contextes implicites, de nombreuses perspectives de recommandations sont offertes telles que l'adaptation au niveau de diversité souhaitable pour l'utilisateur [Castagnos et al., 2010], la détection d'erreurs de navigations ou encore la détection de sessions de navigation parallèles. Deuxièmement, nous avons présenté le concept de contexte implicite dans le but de proposer une alternative aux contextes explicites qui possèdent certains inconvénients. Comme nous l'avons vu dans l'état de l'art, la récupération d'informations relatives aux contextes explicites des utilisateurs pose des problèmes de vie privée mais également des contraintes liées à la définition et la récupération des dimensions qui composent le contexte explicite. Dans ce chapitre, nous avons montré que les changements de contextes explicites peuvent être retrouvés à l'aide des contextes implicites. Ne possédant pas de corpus de données pour lesquels les données contextuelles sont connues, nous avons formulé l'Hypothèse n° 1 afin de définir des changements de contextes explicites dans nos corpus. Forts des résultats que nous avons alors obtenus sur la base de cette hypothèse, nous souhaitons à présent mettre en lumière les liens existants entre les contextes explicites des utilisateurs et les caractéristiques des items que nous possédons. En effet, si des connexions entre les attributs sur

lesquels nous avons travaillé et les dimensions/conditions contextuelles sont mis en évidence, il deviendra alors possible d'affirmer que l'étude de l'évolution de ces attributs dans le temps (par le biais de la diversité) constitue une alternative aux contextes explicites.

Nous avons atteint le **second** objectif : proposer une nouvelle famille de contextes (appelés contextes implicites) ainsi qu'un modèle (DANCE) permettant de détecter ces derniers dans un parcours de consultations d'items. DANCE repose sur la détection des variations remarquables de diversité afin d'isoler les contextes implicites. Ce modèle possède l'avantage d'être générique et de posséder de nombreux paramètres tels que la taille de l'historique utilisé, la sensibilité du modèle, les conditions de détection ou encore la configuration des poids des attributs. Étant donné le fort taux de recouvrement existant entre les changements de contextes implicites et les changements de sessions, nous soutenons sous l'hypothèse d'un lien entre changements de sessions et changements de contextes explicites, que les contextes implicites peuvent substituer les contextes explicites.

Objectif suivant : mettre en évidence les liens existants entre les attributs des items sur lesquels reposent les contextes implicites et les différentes dimensions/conditions contextuelles.

Publications :

- Castagnos, S., L'Huillier, A., and Boyer, A. (2015). Toward a robust diversity-based model to detect changes of context. In 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2015), Vietri sul Mare, Italy.
- L'Huillier, A., Castagnos, S., and Boyer, A. (2016b). The new challenges when modeling context through diversity over time in recommender systems. In Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP '16, pages 341–344, New York, NY, USA. ACM.
- L'Huillier, A., Castagnos, S., and Boyer, A. (2016a). Modéliser la diversité au cours du temps pour détecter le contexte dans un service de musique en ligne. *Revue des Sciences et Technologies de l'Information (TSI)*.

Chapitre 5

Étude utilisateur : exploration des relations existantes entre les attributs des items et le contexte

Sommaire

5.1	But de l'étude	107
5.2	Protocole	108
5.3	Résultats et discussions	112

5.1 But de l'étude

Nous avons montré précédemment que les changements de contextes implicites et les changements de sessions coïncident dans de nombreux cas. L'Hypothèse n° 1 (*cf. supra* 4.2.1 p. 79) que nous avons formulée ne pouvant pas être vérifiée faute de corpus disponibles contenant des informations contextuelles, nous souhaitons voir s'il est possible d'établir les relations entre les attributs et les contextes. Autrement dit, comme nous ne pouvons pas prouver à ce stade que les changements de sessions sont liés à des changements de contextes, nous nous interrogeons sur les liens existant entre les attributs et le contexte. Etant donné que les contextes implicites sont basés sur les attributs et leurs évolutions au cours du temps, montrer qu'il existe effectivement un lien entre les attributs et les conditions/dimensions contextuelles permettrait de renforcer notre thèse selon laquelle les contextes implicites constituent des alternatives sérieuses aux contextes explicites. Bien que certaines études ont démontré l'existence de liens entre les caractéristiques acoustiques des musiques et/ou leurs paroles avec les émotions procurées par ces dernières [Sezgin et al., 2012], la relation entre le contexte explicite et les caractéristiques des items n'a pas été explorée. Certaines approches de recommandation exploitent certes les caractéristiques des items consultés dans un contexte pour trouver des items similaires à recommander dans ce même contexte [Cena et al., 2006]. Cependant, nous n'avons pas trouvé dans la littérature de travaux explicitant les liens contextes-attributs. Le but de cette étude est donc de mettre en lumière les relations existantes entre les caractéristiques des items consultés et les contextes explicites. Plus particulièrement, nous souhaitons savoir si les attributs que nous possédons dans nos corpus de données peuvent être utilisés afin de mettre en évidence ce lien. De plus, cette étude nous permettra d'étudier la perception qu'ont les utilisateurs des différentes dimensions contextuelles et notamment les associations qu'ils font entre les musiques et le contexte. Dans

ce but, nous avons mis en place une étude utilisateur dans laquelle les participants ont eu pour tâche de renseigner les contextes dans lesquels ils écouteront les musiques leur étant proposées. En exploitant les annotations des utilisateurs ainsi que les attributs des musiques que nous possédons, nous montrerons, après avoir présenté le protocole de l'étude, quels sont les liens pouvant exister entre les attributs et les différentes dimensions contextuelles.

5.2 Protocole

Nous avons mis notre étude utilisateur en ligne afin de la rendre disponible au plus grand nombre. Étant donné que le principe de l'étude repose sur l'écoute de musiques pour lesquelles nous souhaitons connaître les préférences contextuelles des utilisateurs, nous avons dû concevoir entièrement le site web nécessaire à cette expérience. En effet, l'utilisation de sites de sondage en ligne tels que SurveyMonkey⁶⁹ ou encore Google Forms⁷⁰ offrent des options limitées en ce qui concerne l'intégration de musiques dans un questionnaire (d'autant plus que les musiques sont choisies en fonction des préférences des utilisateurs). Outre les aspects techniques et étant donné l'effort cognitif requis par la tâche demandée aux utilisateurs, nous avons également procédé à plusieurs études pilotes nous permettant d'établir le meilleur rapport entre la difficulté imposée par la tâche et la quantité/qualité des données récupérées.

Nous avons recruté les participants par e-mail en utilisant une liste de diffusion universitaire⁷¹ et une liste à destination de la communauté scientifique internationale⁷² dans le domaine de la modélisation utilisateur. Le message diffusé dans ces listes contenait une présentation de l'équipe, une brève description de l'étude ainsi qu'un lien vers le site internet de l'étude⁷³. Les 4 phases principales de l'étude sont détaillées ci-après.

Page d'accueil : la première page est composée d'une présentation de l'étude ainsi que d'un formulaire de consentement pour l'acquisition des données dont l'acceptation est obligatoire afin de poursuivre l'étude. Cette page permet également de choisir la langue de l'interface, étant donné que ce questionnaire était également adressé à un public anglophone.

Pré-questionnaire : la seconde page comporte des questions démographiques concernant le sexe, l'âge et la catégorie socioprofessionnelle ainsi que des questions concernant les habitudes d'écoute de musique. Les participants doivent alors préciser le nombre d'heures de musique qu'ils écoutent en moyenne par jour ainsi que l'endroit dans lequel il écoute le plus de la musique (chez soi, au travail, en voiture...). Des questions concernant l'utilisation de plateformes de musique en ligne et la manière dont les musiques sont choisies (selon l'artiste ou la musique) sont également proposées. Nous demandons également de préciser quels sont les genres de musiques écoutés parmi les suivants : rock, pop, rap, country, punk, jazz, hip-hop, classique, folk, métal, électronique, blues. Obtenir les genres appréciés par chaque participant nous permet de lui proposer par la suite des musiques appartenant aux genres choisis, afin de maximiser les chances que les musiques proposées puissent être annotées.

69. <https://fr.surveymonkey.com/>

70. <https://www.google.fr/intl/fr/forms/about/>

71. La liste utilisée est une liste propre à l'Université de Lorraine se nommant "expression-libre" et comportant plus de 4 000 abonnés.

72. <http://www.um.org/get-involved>

73. <http://movit.tv/tut9/>

Annotation des musiques : lors de cette étape, 15 musiques choisies aléatoirement sont proposées à chaque participant en fonction des genres musicaux précisés dans le pré-questionnaire. Les musiques sont jouées à partir de la plateforme Youtube⁷⁴ ce qui permet de pouvoir écouter les morceaux de musique dans leur intégralité sans que les utilisateurs ne soient inscrits (ce qui n'est pas le cas pour d'autres plateformes de lecture). Pour chacune de ces musiques, les participants doivent alors préciser s'ils connaissent déjà la musique ou l'artiste, et s'ils apprécient ou non ces derniers. Si un utilisateur n'aime pas la musique proposée ou encore l'artiste correspondant, la musique suivante est alors automatiquement proposée⁷⁵. En effet, si un utilisateur n'apprécie pas une musique, il ne pourra pas spécifier les contextes dans lesquels il souhaite l'écouter. Dans le cas contraire, l'utilisateur est invité à donner son point de vue par rapport à cette musique selon deux parties distinctes : une partie concernant les émotions suscitées par cette musique et une partie concernant les contextes d'écoute appropriés. La Figure 5.1 ci-dessous représente une capture d'écran sur laquelle nous pouvons voir le lecteur de musique, la partie concernant l'appréciation de la musique ainsi que la partie émotion et la partie contexte. La partie émotion vise à connaître les émotions que la musique procure à l'utilisateur ainsi que les émotions qui pousseraient l'utilisateur à écouter cette musique. Par exemple, un utilisateur peut être dans un état de colère et décider d'écouter une musique relaxante. Nous parlerons alors respectivement d'émotions initiatrices et d'émotions ressenties. Les émotions proposées sont présentées dans le Tableau 5.2 et ont été choisies à partir des travaux de [Schubert, 2003]. Dans la partie concernant les contextes d'écoute, un ensemble de 9 dimensions contextuelles est présenté. Pour chaque dimension, l'utilisateur doit préciser si cette dernière influence son envie d'écouter la musique proposée. Si tel est le cas, un ensemble de conditions contextuelles lui est proposé afin de préciser son choix. Par exemple, si un utilisateur estime qu'il écouterait une musique en fonction de la "météo", il peut ensuite préciser son choix en sélectionnant une ou plusieurs conditions météorologiques (pluie, soleil...). Les dimensions contextuelles ainsi que les conditions contextuelles associées sont résumées dans le Tableau 5.1 ci-dessous. La Figure 5.1 illustre les deux parties de la phase d'annotation que sont la partie émotion et la partie contexte (situation). La partie inférieure de la Figure 5.1 représente l'interface lorsqu'un participant n'a pas encore précisé si une dimension contextuelle influe sur le choix de cette musique, alors que la Figure 5.2 représente les conditions contextuelles correspondantes aux différentes dimensions contextuelles sélectionnées. Si un utilisateur précise qu'une situation (contexte) n'a pas d'incidence sur le choix de cette musique, nous ne demandons évidemment pas de renseigner les conditions contextuelles. Pour chaque musique, nous requérons également de la part des utilisateurs qu'ils décrivent avec leurs propres mots le contexte dans lequel ils écouterait cette musique.

Post-questionnaire : la dernière partie de l'étude est composée d'une série de 11 affirmations pour lesquelles les utilisateurs peuvent donner leur degré d'accord selon une échelle de Likert à 5 modalités. Les affirmations sont résumées dans le Tableau 5.3 ci-dessous.

74. <https://www.youtube.com/>

75. Si un utilisateur n'apprécie pas une musique, une nouvelle musique appartenant aux genres appréciés par l'utilisateur sera tirée au hasard et lui sera proposée par la suite afin que chaque utilisateur puisse annoter 15 musiques

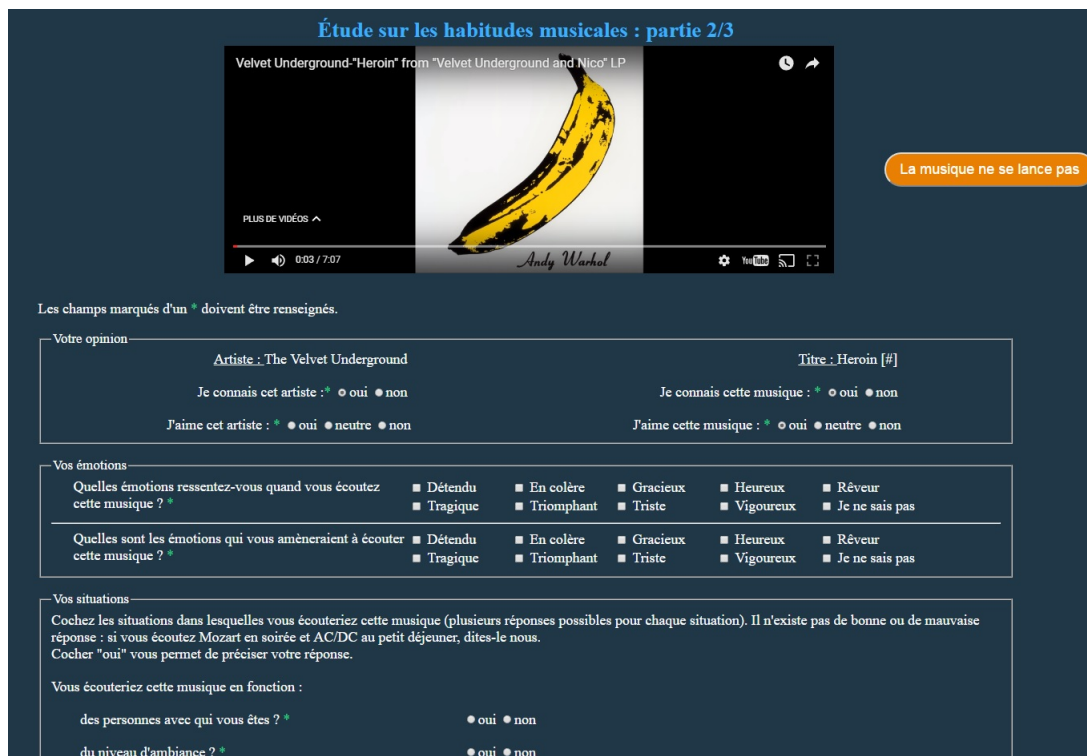


FIGURE 5.1 – Capture d'écran de la phase d'annotation sur l'opinion et les émotions

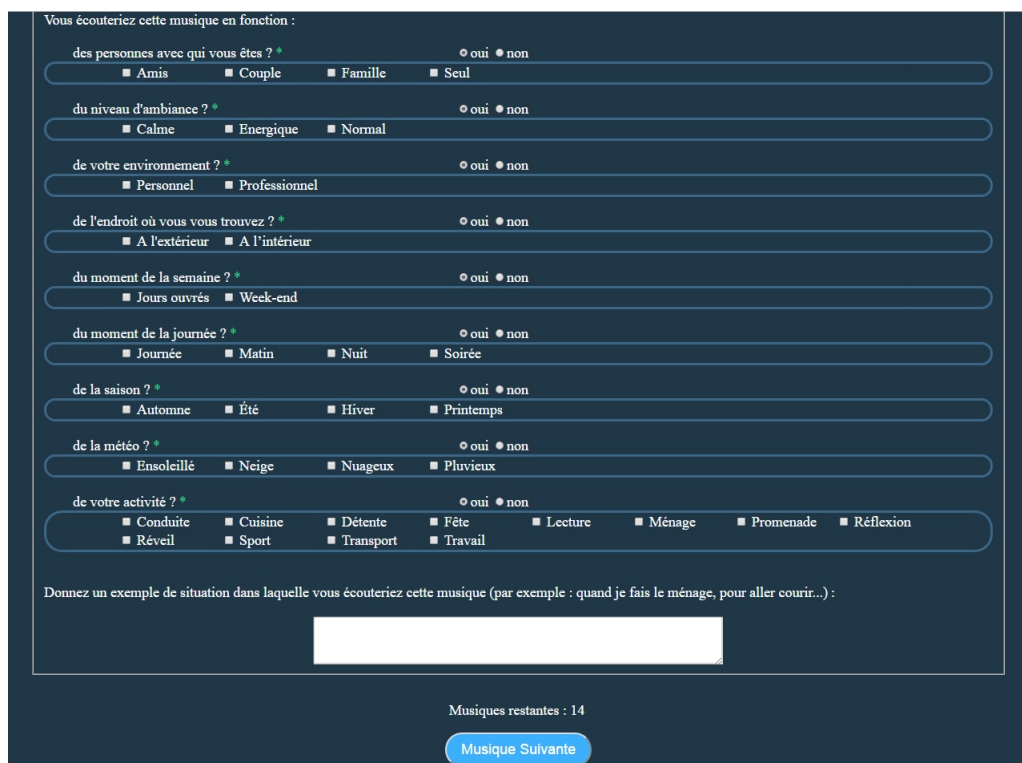


FIGURE 5.2 – Capture d'écran de la phase d'annotation sur les situations contextuelles

TABLE 5.1 – Dimensions et conditions contextuelles utilisées

Dimensions Contextuelles	Conditions Contextuelles
Activité	conduite, cuisine, détente, fête, lecture, ménage, promenade, réflexion, réveil, sport, transport, travail
Jour	matin, journée, soirée, nuit
Énergie	calme, normal, énergique
Environnement	personnel, professionnel
Lieu	intérieur, extérieur
Saison	printemps, été, automne, hiver
Social	seul, famille, amis, couple
Météo	ensoleillé, pluvieux, neigeux, nuageux
Semaine	semaine, week-end

TABLE 5.2 – Émotions utilisées [Schubert, 2003]

Émotions	Détendu, En colère, Gracieux, Heureux, Rêveur, Triste, Tragique, Triomphant, Vigoureux
----------	--

TABLE 5.3 – Affirmations utilisées pour le post-questionnaire

Numéro	Affirmation
A1	J'ai facilement réussi à associer les musiques et les émotions proposées.
A2	Les situations proposées étaient suffisantes pour définir les contextes dans lesquels j'écoute de la musique.
A3	Les musiques que j'ai envie d'écouter dépendent des émotions que j'ai envie de ressentir.
A4	Les musiques que j'ai envie d'écouter dépendent des émotions que je ressens.
A5	L'émotion que je ressens pour une musique dépend de la situation dans laquelle je me trouve.
A6	Les musiques que j'ai envie d'écouter dépendent de ma situation.
A7	Je peux écouter n'importe quelle musique n'importe quand.
A8	Pour une situation donnée, si on vous fournit une recommandation de chanson appropriée, toutes les autres chansons de ce même genre seront également appropriées.
A9	Pour un situation donnée, si on vous fournit une recommandation de chanson appropriée, toutes les autres chansons de ce même artiste seront également appropriées.
A10	Pour une émotion donnée, si on vous fournit une recommandation de chanson appropriée, toutes les autres chansons de ce même genre seront également appropriées.
A11	Pour une émotion donnée, si on vous fournit une recommandation de chanson appropriée, toutes les autres chansons de ce même artiste seront également appropriées.
Modalités utilisées	1 : Tout à fait d'accord, 2 : D'accord, 3 : Ni en désaccord ni d'accord, 4 : Pas d'accord, 5 : Pas du tout d'accord

Les musiques que nous avons sélectionnées sont issues du corpus de Lastfm présenté dans la Section 2.1. Nous avons choisi aléatoirement 30 musiques appartenant à chacun des genres présentés dans la partie 2 du questionnaire que nous rappelons : rock, pop, rap, country, punk, jazz, hip-hop, classique, folk, métal, électronique, blues. Sur l'API de Spotify, il existe des milliers de genres musicaux. Par exemple, il existe plus de 124 genres contenant le terme "rock". Proposer alors l'ensemble des genres aurait uniquement augmenté la charge cognitive des utilisateurs dans le choix des genres. Nous avons choisi de présenter aux utilisateurs uniquement les genres les plus courants que nous avons listés précédemment. Nous avons procédé à un tirage aléatoire sur l'ensemble du corpus initial afin de limiter les biais dus aux musiques populaires et nous avons également choisi de ne pas prendre plusieurs musiques du même artiste. Nous nous sommes également assurés que tous les attributs de chacune des musiques étaient bien disponibles. Nous rappelons par ailleurs que l'ensemble du matériel que nous avons collecté pendant cette étude est disponible à cette adresse : <https://github.com/teamKiwi/umap2017>.

5.3 Résultats et discussions

Dans le but de minimiser l'impact des participants peu investis dans l'étude, nous avons supprimé les réponses des participants n'ayant pas terminé l'étude. Nous avons ainsi obtenu un corpus de données contenant les réponses de 172 utilisateurs ayant en tout effectué 2 580 annotations réparties sur 360 musiques. La parité homme-femme est presque respectée puisque 55,76 % des participants sont des hommes et 44,76 % des participants sont des femmes. En ce qui concerne les habitudes d'écoute des participants, ces derniers ont déclaré écouter en moyenne 2,51 heures de musique par jour. Concernant la tâche d'annotation, le Tableau 5.3 ci-dessous résume la répartition du nombre d'annotations par musique. Nous pouvons constater que la majorité des musiques a été annotée plusieurs fois puisque seules 38 musiques ont été annotées une fois seulement. Cela signifie que les associations entre une musique et les dimensions/conditions contextuelles sont obtenues par plusieurs utilisateurs et que le biais dû aux préférences d'un utilisateur est donc limité.

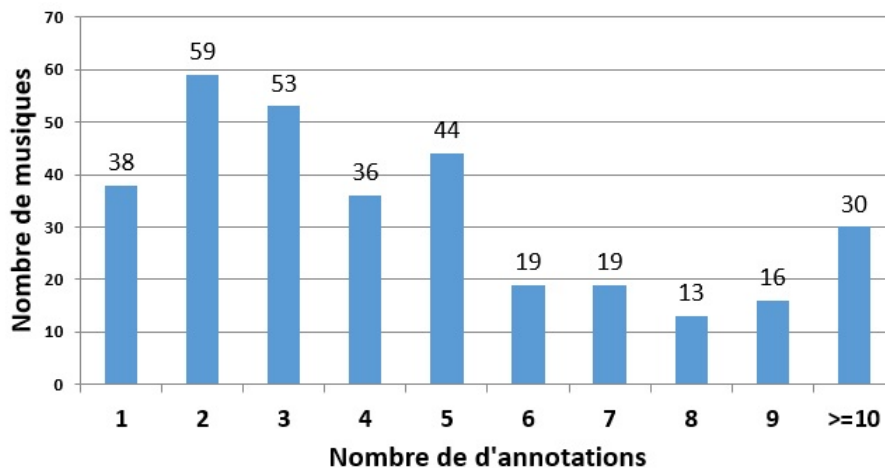


FIGURE 5.3 – Répartition du nombre d'annotations des musiques

En ce qui concerne les annotations, le nombre moyen d'annotations (émotions et contexte) pour chaque musique est de :

- 4,78 dimensions sur les 11 dimensions disponibles ;

— 9,27 conditions sur les 46 disponibles.

Le Tableau 5.4 ci-dessous résume les annotations concernant les musiques et les émotions. Pour une même musique, l'utilisateur peut choisir plusieurs émotions. La colonne "nombre" représente le nombre de fois où une émotion a été attribuée à une musique. Le "ratio" représente le rapport entre le nombre de fois où cette émotion a été choisie et le nombre de musiques ayant été annotées. Cette valeur nous permet de connaître le pourcentage de fois où cette émotion a été choisie. Nous avons demandé aux utilisateurs d'annoter les musiques par rapport à la dimension émotionnelle pour plusieurs raisons. Premièrement, les émotions et donc l'état mental des utilisateurs est une composante importante du contexte. Deuxièmement, même si cela n'est pas le but premier de cette étude, nous souhaitons voir s'il existe un lien entre les attributs des musiques que nous possédons et les émotions procurées ou initiatrices ;

TABLE 5.4 – Distribution des émotions choisies

Émotions	Ressenties		Initiatrices	
	Nombre	Ratio	Nombre	Ratio
En colère	73	2,83	119	4,46
Rêveur	526	20,39	409	15,85
Gracieux	163	6,32	95	3,68
Heureux	483	18,72	458	17,75
Détendu	745	28,88	621	24,07
Triste	181	7,02	238	9,22
Tragique	100	3,88	78	3,02
Triomphant	265	10,27	224	8,68
Vigoureux	380	14,73	260	10,08
Je ne sais pas	354	13,72	363	14,07
Total	3270	—	2865	—

En ce qui concerne les dimensions contextuelles, les résultats sont présentés dans le Tableau 5.5 ci-dessous. Ces résultats nous montrent que les utilisateurs sont plus sensibles à certaines dimensions contextuelles que d'autres. Par exemple, les dimensions contextuelles telles que "activité", "énergie", "environnement" et "social" sont souvent sélectionnées par les utilisateurs. Si une dimension contextuelle est choisie par un utilisateur, nous rappelons que cela signifie que cette dimension a un impact sur le souhait de l'utilisateur d'écouter cette musique. Inversement, les utilisateurs semblent être en moyenne moins sensibles à certaines dimensions telles que la saison ou encore la météo.

TABLE 5.5 – Distribution des dimensions contextuelles choisies

Dimension	# Sélectionné	ratio
Activité	853	20,45
Jour	486	17,05
Énergie	938	32,91
Environnement	698	24,49
Lieux	515	18,07
Saison	104	3,65
Social	882	30,94
Météo	140	4,91
Semaine	172	6,03

Afin de mettre en évidence les liens pouvant exister entre les attributs d'une musique et les contextes dans lesquels elle sera consultée, nous avons préalablement établi les boîtes à moustaches entre les différents attributs et les différentes conditions contextuelles. La Figure 5.4 ci-dessous représente 4 de ces boîtes à moustaches. Ces 4 boîtes à moustaches constituent des exemples tirés de l'ensemble des boîtes à moustaches qui sont présentes dans les annexes de ce manuscrit p. 154.

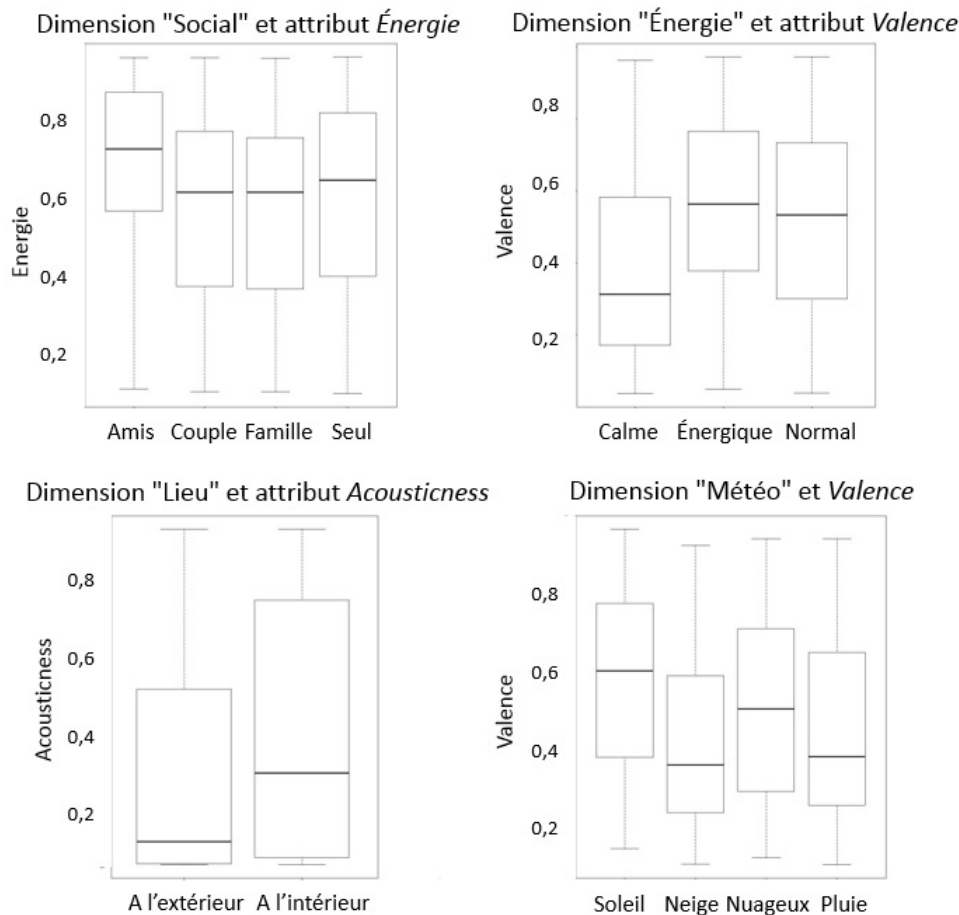


FIGURE 5.4 – Exemples de boîtes à moustaches obtenues

Ces représentations ne nous permettent pas d’affirmer qu’il existe un lien entre les attributs et les contextes d’écoute. Cependant, ces représentations sont un moyen d’obtenir un aperçu de la distribution des valeurs des attributs en fonction des modalités/conditions de chacune des dimensions contextuelles. Pour chacune des boîtes à moustaches que nous avons sélectionnées dans la Figure 5.4, nous pouvons constater qu’au moins une des modalités contextuelles possède une distribution qui semble différente des autres modalités. Par exemple, dans le cas où nous comparons les conditions “calme”, “énergique” et “normal” appartenant à la dimension “énergie” par rapport à l’attribut **valence** des musiques, nous constatons que la distribution de cet attribut pour la condition “calme” est différente des 2 autres distributions. Cependant, il ne semble pas y avoir de différence entre les distributions des conditions “énergique” et “normal”. Dans le cas où nous avons comparé ce même attribut avec les différentes conditions de la dimension “météo”, la boîte à moustaches obtenue nous laisse supposer que les 4 conditions donnent lieu à des distributions différentes entre elles.

Afin de vérifier si les différences que nous observons sont significatives, nous avons effectué un test de Kruskal-Wallis. Nous n’avons pas pu appliquer une ANOVA, suite au rejet de l’hypothèse de normalité de distributions des données. Les résultats du test statistique que nous avons effectué sont résumés dans le Tableau 5.6 ci-dessous. La valeur de la *p-value* représente la probabilité qu’aucune condition contextuelle ne diffère des autres conditions pour un attribut donné. Autrement dit, plus cette valeur est faible, plus il est probable qu’il y ait effectivement un lien entre les valeurs des attributs et les différentes conditions contextuelles de la dimension contextuelle correspondante.

Les résultats que nous avons obtenus nous permettent d’établir plusieurs conclusions. Premièrement, nous pouvons confirmer qu’il existe un lien entre les émotions (ressenties et initiatrices) et les attributs des musiques. En effet, toutes les émotions ressenties et presque toutes les émotions initiatrices possèdent une *p-value* qui est significative. Nous sommes conscients que ces résultats confirment des conclusions déjà établies dans la littérature [Barthet et al., 2013]. Cela nous assure néanmoins de la qualité du matériel que nous avons recueilli, tant au niveau de l’étude utilisateur (réponses sérieuses des volontaires) que des attributs des musiques que nous avons récupérés. Deuxièmement, nous pouvons constater que certains attributs sont très dépendants du contexte contrairement à d’autres. Par exemple, des attributs tels que la “valence” ou l’“énergie” sont des attributs pour lesquels le contexte explicite semble déterminant. Inversement, des attributs tels que la “popularité” de l’artiste ou encore la *liveness* sont peu dépendants du contexte explicite. Cela ne veut pas dire pour autant que ces attributs ne peuvent pas être utiles afin de prendre en compte les préférences des utilisateurs à l’échelle des contextes implicites. Par exemple, si un utilisateur ne consulte que des musiques populaires, il est évident que cet attribut est essentiel afin de ne pas lui proposer des musiques d’artistes indépendants et inversement. Dans ce cas, le contexte explicite n’aura aucun rôle à jouer, alors que le contexte implicite pourra permettre d’adapter les recommandations à ce besoin. D’ailleurs, cette hypothèse est soutenue par les résultats que nous avons obtenus en optimisant les poids des attributs. En effet, dans le Tableau 4.19 page 101, nous pouvons constater que la popularité est le 3^{ème} attribut ayant le plus de poids. L’attribut *liveness* pour sa part est classé 9^{ème} avec le corpus de Deezer et 5^{ème} avec le corpus de Lastfm, ce qui signifie dans ce dernier cas que cet attribut est important lors du découpage en CI. Ainsi, le fait que ces deux attributs ne soient pas mis en avant dans cette étude comme étant fortement liés au contexte explicite des utilisateurs, tend à montrer que ces derniers sont importants mais qu’ils seront mal compris ou acceptés s’ils sont utilisés directement dans les recommandations. De bonnes recommandations peuvent être générées en prenant en compte de fortes variations de ces attributs, mais ces attributs seuls semblent ne pas suffire à fournir de bonnes explications. De plus, une explication possible quant au fait que ces attributs se soient

TABLE 5.6 – Résumé du test de Krusal-Wallis : * p-value ≤ 0.05 , ** p-value ≤ 0.01 , *** p-value ≤ 0.001

Attribut	Emotions				Dimensions contextuelles									
	Resentites	Initiatrices	activité	énergie	environnement	jour	lieu	météo	saison	semaine	social			
Acousticness	2.20E-16***	2.20E-16***	2.20E-16***	2.20E-16***	1.50E-02*	1.82E-01	7.31E-06***	4.79E-03**	3.30E-02*	4.95E-05***	7.04E-13***			
Danceability	2.20E-16***	2.92E-14***	2.52E-13***	6.26E-07***	2.94E-01	6.35E-01	3.44E-02*	1.67E-04**	2.92E-02*	1.46E-03***	3.20E-03**			
Durée	4.23E-05***	6.23E-07***	3.90E-05***	1.78E-03**	2.77E-01	8.33E-02	1.01E-01	4.42E-01	8.15E-01	4.98E-01	1.06E-01			
Énergie	2.20E-16***	2.20E-16***	2.20E-16***	2.20E-16***	2.94E-02*	1.69E-03**	1.74E-07***	8.23E-04***	1.22E-02*	6.55E-05***	1.30E-13***			
Instrumentalness	6.02E-04***	3.06E-03**	2.64E-07***	7.00E-04***	2.86E-02*	2.66E-01	7.33E-02	5.53E-01	1.69E-01	3.16E-01	7.98E-02			
Liveness	1.58E-02*	3.75E-01	7.69E-01	4.09E-01	3.25E-01	6.61E-01	8.42E-02*	6.63E-01	9.80E-01	4.51E-01	9.16E-03*			
Loudness	2.20E-16***	2.20E-16***	2.20E-16***	2.20E-16***	2.18E-02*	2.29E-01	1.48E-06***	1.45E-01	9.27E-03**	4.96E-04***	2.26E-09***			
Popularity art.	1.58E-02*	7.32E-02	8.98E-01	7.86E-01	7.09E-01	6.95E-01	8.40E-01	2.23E-01	4.54E-01	1.44E-01	8.64E-01			
Speechiness	2.20E-16***	2.20E-16***	2.20E-16***	2.20E-16**	4.41E-01	7.23E-01	8.08E-05***	6.69E-02	5.90E-01	4.24E-02*	6.57E-06***			
Tempo	1.04E-05***	2.97E-03**	2.51E-05***	3.78E-07***	6.18E-02	1.95E-01	5.09E-02	5.67E-01	2.76E-01	7.61E-01	1.45E-01			
Valence	2.20E-16**	2.20E-16***	2.20E-16***	2.20E-16***	4.48E-02*	1.63E-05***	1.94E-05***	4.78E-04***	5.78E-05***	1.99E-01	5.14E-05***			

vus attribuer un poids important lors du processus d’optimisation est qu’ils soient corrélés avec certains autres attributs (et que nous ne possédons peut-être pas). Nous dresserons à l’issue de ce chapitre une cartographie permettant d’illustrer l’impact des attributs en termes de découpages en contextes implicites et l’acceptation de leurs associations avec les contextes explicites que nous avons récupérés dans cette étude.

Le test statistique que nous avons effectué dans Tableau 5.7 nous permet d’affirmer qu’en fonction des conditions contextuelles, les utilisateurs choisissent des musiques ayant des attributs particuliers. Cependant, ce test ne nous explique pas si les valeurs des attributs sont dépendantes de toutes les conditions contextuelles ou seulement de certaines d’entre elles. Afin d’y parvenir, nous avons effectué une analyse *post-hoc*⁷⁶ du test de Kruskal-Wallis. Le but de cette analyse est de déterminer s’il existe une différence entre les conditions contextuelles prises deux à deux en comparant les différences des distributions théoriques avec les différences des distributions observées. Si la différence observée est supérieure à la différence critique pour ces deux conditions, alors les deux conditions testées sont effectivement (significativement) différentes. Le Tableau 5.7 ci-dessous représente les résultats de l’analyse *post-hoc* pour l’attribut "valence" et la dimension "météo". Ce test peut être effectué sur l’ensemble des associations "dimensions contextuelles"- "attributs" à partir du moment où le test de Kruskal-Wallis est validé, c’est-à-dire que la *p-value* est inférieure à 0,05.

TABLE 5.7 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension météo et l’attribut "valence"

Paire	Différence observée	Différence critique	Résultat
Ensoleillé-Neigeux	344,84	246,53	Vraie
Ensoleillé-Nuageux	172,26	231,54	Faux
Ensoleillé-Pluvieux	282,99	223,14	Vraie
Neigeux-Nuageux	172,58	242,41	Faux
Neigeux-Pluvieux	61,85	234,39	Faux
Nuageux-Pluvieux	110,73	218,57	Faux

Le Tableau 5.7 nous apprend qu’il existe une différence entre les distributions des valeurs de l’attribut "valence" et les paires de conditions ensoleillé-neigeux et ensoleillé-pluvieux. Autrement dit, la valeur "valence" ne semble impacter le choix des utilisateurs que pour ces deux paires d’attributs. Pour des raisons de place, l’ensemble des analyses *post-hoc* est disponible en annexe p.154. Cependant, nous présentons dans les Figures 5.5-5.15 ci-dessous, une synthèse des relations entre les attributs et les dimensions contextuelles que nous avons mis en lumière. Dans ces figures, lorsqu’un attribut est lié à une dimension contextuelle, cela signifie que les valeurs de cet attribut sont influencés par les conditions de cette dimension. Les fractions présentes sur les arêtes représentent le nombre de paires de conditions pour lesquelles une différence significative est observée (qui a été obtenue à l’aide du test *post-hoc* de Kruskal-Wallis). Par exemple, en ce qui concerne la dimension contextuelle "Activité", le quotient $\frac{44}{66}$ présent sur le lien avec l’attribut "énergie" nous informe que sur les 66 paires possibles formées par les 12 conditions, 44 paires présentent des différences de distributions statistiquement significatives. Autrement dit, pour les paires considérées (*cf. infra* Tableau A.3, p.163), il existe bien une influence de chaque condition sur les valeurs de l’attribut testé (l’"énergie" dans cet exemple).

76. Ce test a été effectué à l’aide du *package* `kruskalmcpgirmess` du logiciel R. <https://www.r-project.org/>

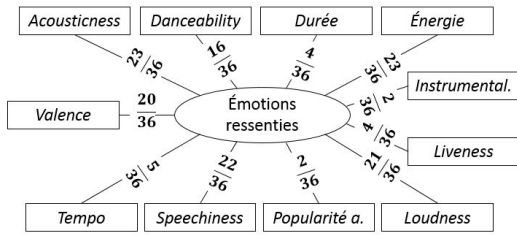


FIGURE 5.5 – Émotions ressenties

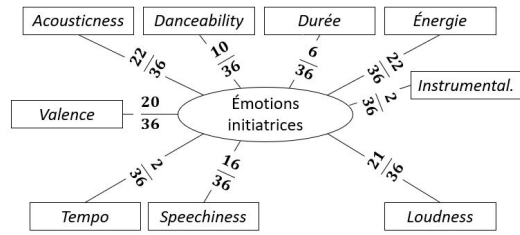


FIGURE 5.6 – Émotions initiatrices

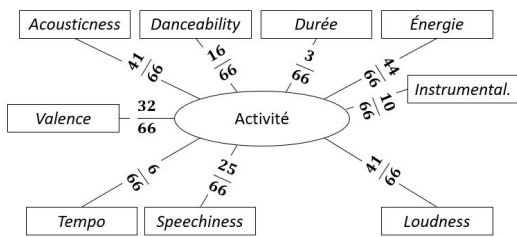


FIGURE 5.7 – Activité

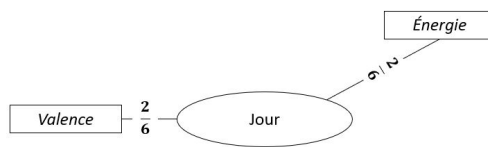


FIGURE 5.8 – Jour

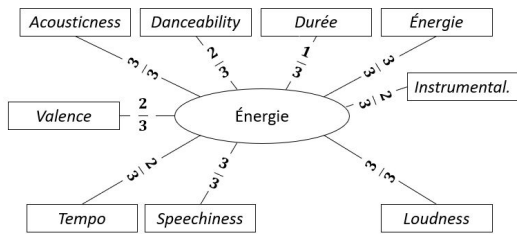


FIGURE 5.9 – Énergie

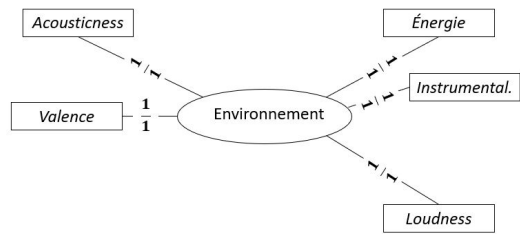


FIGURE 5.10 – Environnement

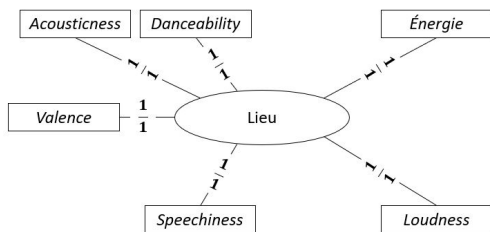


FIGURE 5.11 – Lieu

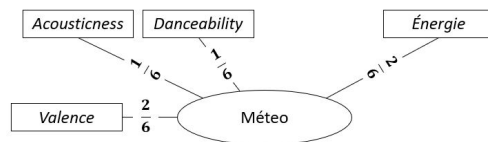


FIGURE 5.12 – Météo

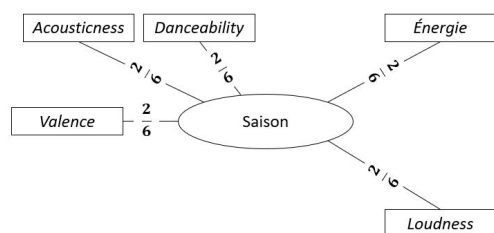


FIGURE 5.13 – Saison

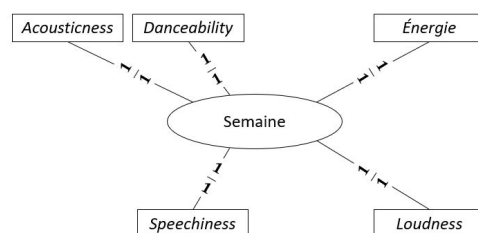


FIGURE 5.14 – Semaine

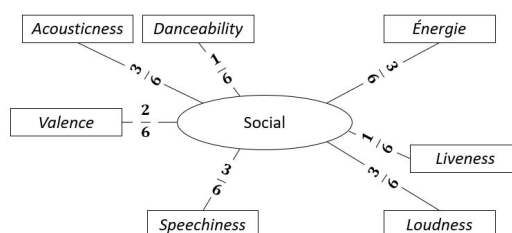


FIGURE 5.15 – Social

À travers ces différentes représentations, nous pouvons constater qu’au sein d’une même dimension, les paires de conditions pour lesquelles il existe des différences significatives ne sont pas nécessairement les mêmes en fonction des attributs. Par exemple, pour la dimension “émotions ressenties” (Figure 5.5), 23 paires de conditions sur les 36 paires possibles se “distinguent” par rapport à l’attribut *acousticness* alors que seulement 2 paires de conditions se “distinguent” par rapport à l’attribut popularité. Ces écarts en termes de nombre de paires pour lesquelles il existe effectivement des différences en fonction des attributs s’observent par ailleurs sur l’ensemble des dimensions contextuelles que nous avons utilisées. De plus, nous pouvons constater que pour certaines dimensions telles que le “jour” ou encore la “météo”, le nombre d’attributs sur lesquelles des paires de conditions se distinguent est limité. Ainsi, alors qu’un changement de condition de la dimension contextuelle “activité” impactera certainement le choix de l’utilisateur sur certains attributs, la période de la journée n’impactera que de manière très limitée les choix de l’utilisateur. Ces résultats nous permettent d’illustrer à quel point le contexte explicite des utilisateurs peut être complexe à prendre en compte dans les SR. En effet, dans notre étude utilisateur, nous avons demandé aux utilisateurs de nous dire s’ils écouteront les musiques proposées en fonction des différentes dimensions du contexte, et ce, de manière indépendante. C’est-à-dire que nous n’avons pas pris en compte les différentes combinaisons des dimensions contextuelles qui peuvent former ce que l’on appelle le contexte (*cf. supra* définition du contexte, p. 27). Par exemple, le contexte d’un utilisateur peut être : se trouver seul en soirée en hiver alors qu’il neige. Or, il est possible que ces 4 conditions contextuelles (“social”=seul, “jour”=soirée, “météo”=neigeux et “saison”=hiver) prises individuellement ne suggèrent pas les mêmes caractéristiques que le contexte les réunissant. Ainsi, un système de recommandation basé sur le contexte explicite devra posséder de nombreuses préférences contextualisées avant de pouvoir mettre ces dernières à contribution dans les recommandations. Cela est d’autant plus vrai que le nombre de conditions contextuelles obtenues est important. Si le système collecte les préférences des utilisateurs ainsi que de nombreuses conditions contextuelles, il lui faudra de nombreux exemples/préférences contextualisés

avant de pouvoir déterminer quels sont réellement les liens existants entre les différentes conditions contextuelles et les préférences. D'autant plus que le système devra alors être certain de pouvoir collecter/posséder les dimensions contextuelles ayant un impact sur les préférences des utilisateurs. Pour ces raisons, nous soutenons que les contextes implicites possèdent l'avantage de s'intéresser directement aux attributs et à leurs variations et donc de ne pas reposer sur un processus d'inférence visant dans un premier à détecter le contexte de l'utilisateur puis de déterminer les items adaptés à ce dernier dans un second temps.

Enfin, les réponses obtenues lors du post-questionnaire sont résumées dans la Figure 5.16 ci-dessous :

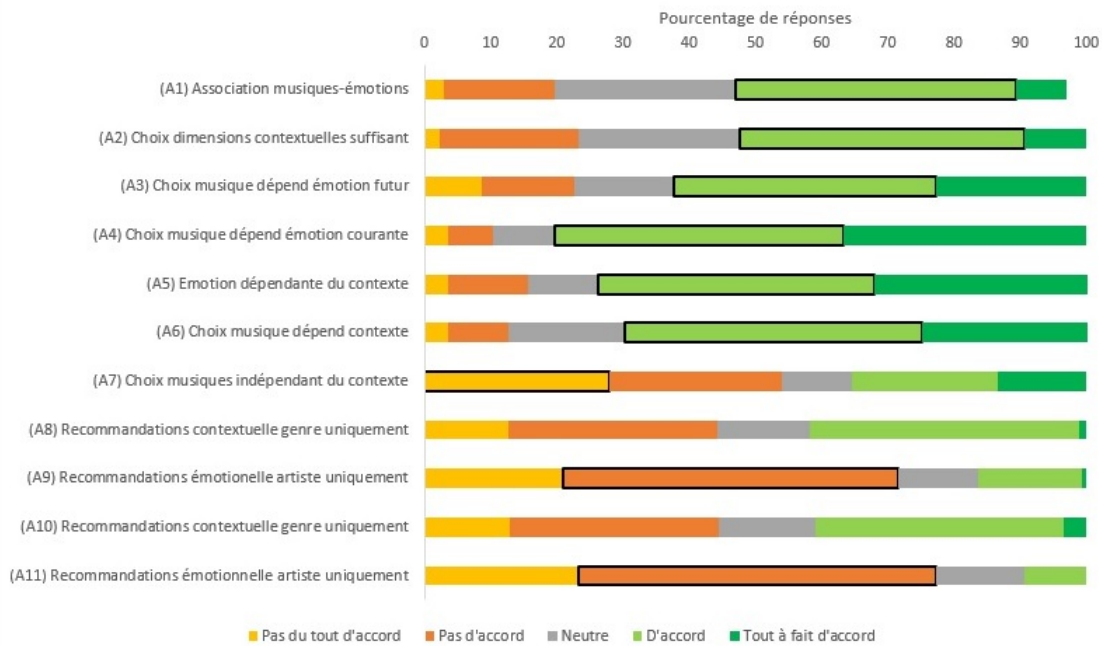


FIGURE 5.16 – Pourcentage de réponses pour chacune des affirmations

Les résultats obtenus à ce questionnaire nous permettent de confirmer certaines conclusions de l'état de l'art mais également d'avancer de nouvelles hypothèses. Premièrement, les réactions aux affirmations A3, A4, A5, A6 et A7 confirment que les dimensions émotionnelles et contextuelles des utilisateurs jouent des rôles primordiaux dans le choix des musiques. La réponse à l'affirmation A7 est intéressante puisqu'elle nous apprend que le contexte explicite n'est pas une dimension à laquelle tous les utilisateurs semblent sensibles. En limitant les recommandations au contexte des utilisateurs, un SR peut alors susciter l'incompréhension d'un utilisateur qui verra ses recommandations différer en fonction de ses contextes alors qu'il n'y voit pas d'intérêt. Deuxièmement, étant donné les réponses aux affirmations A1 et A2, la phase d'annotation des musiques n'a pas posé de réel problème à la majorité des utilisateurs. Cependant les pourcentages obtenus pour les réponses "Pas d'accord" et "Neutre" à ces affirmations nous montrent tout de même que cette tâche reste complexe pour certains utilisateurs. Comme l'ont démontré certaines études, telles que [Odic et al., 2011], il est difficile pour les utilisateurs de se projeter dans des contextes et les réponses données par les utilisateurs peuvent ne pas représenter tous les cas d'usages. Par exemple, nous avons trouvé dans une étude ayant un but très simi-

laire à la nôtre des conclusions sensiblement différentes [Buskirk, 2017]. Dans cette étude, les auteurs ont recoupé les caractéristiques des musiques (identiques à celles dont nous disposons) de 85 milliards d’écoutes avec des données météorologiques en utilisant la position géographique des utilisateurs. Les auteurs ont conclu que l’ensemble des conditions de la dimension météorologique étudiée possède un impact sur la “valence” des musiques écoutées. Nous précisons que ces résultats n’invalident en rien les conclusions de notre étude étant donné qu’ils tendent à renforcer l’idée qu’il existe un lien important entre les contextes explicites et les caractéristiques des musiques que nous possédons. De plus, cette étude a été publiée durant la période pendant laquelle nous avons analysé les résultats de notre étude, ce qui confirme la pertinence de nos travaux. Enfin, les réponses aux affirmations A8, A9, A10 et A11 nous montrent que la similarité entre deux musiques sur certains attributs (*i.e.* ceux des attributs) n’est pas toujours une garantie de satisfaction en matière de recommandation contextuelle. Dans le cas où une musique est effectivement adaptée au contexte d’un utilisateur, recommander une musique du même artiste ou du même genre n’est pas toujours adéquat. Cela confirme le fait que la prise en compte de nombreux attributs des musiques est nécessaire afin de comprendre les besoins des utilisateurs en termes de contexte. Par exemple, si un utilisateur écoute la célèbre musique *Starway to heaven* de Led Zeppelin, des recommandations telles que le morceau *The Immigrant Song* du même artiste ou encore une musique appartenant au même genre tel que *Eruption* de Eddie Van Halen ne sont pas nécessairement de bonnes recommandations. En effet, même si ces dernières sont susceptibles de plaire à l’utilisateur dans l’absolu, elles peuvent toutes les deux ne pas convenir si l’utilisateur recherche des musiques calmes pour danser un slow.

Compte tenu des résultats que nous avons pu obtenir dans cette étude utilisateur, nous pouvons affirmer que les contextes implicites constituent une alternative sérieuse aux contextes explicites. En effet, de nombreuses connexions existent entre les dimensions/conditions contextuelles et les attributs que nous avons utilisés. L’existence de ces liens nous montre que les attributs que nous avons utilisés constituent des dimensions contextuelles implicites pouvant permettre de prendre en compte le contexte des utilisateurs, et ce, sans nécessiter une phase de collecte et d’exploitation de données contextuelles qui comme nous l’avons vu, pose des problèmes au regard du respect de la vie privée des utilisateurs.

Nous avons montré que : les attributs des items consultés sont effectivement liés aux dimensions/conditions contextuelles des utilisateurs. Les contextes implicites déterminés à partir de l’analyse de l’évolution en diversité des attributs des items consultés constituent bien une alternative à la prise en compte du contexte des utilisateurs.

Objectif suivant : proposer des approches de recommandation exploitant les contextes implicites ainsi que l’évolution temporelle de la diversité.

Publication :

- L’Huillier, A., Castagnos, S., and Boyer, A. (2017). Are item attributes a good alternative to context elicitation in recommender systems? In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP ’17, pages 371–372, New York, NY, USA. ACM.

Chapitre 6

Recommandations, diversité et contextes implicites

Sommaire

6.1	Approches de recommandation	123
6.1.1	Recommandation collaborative à base de CI	124
6.1.2	Recommandation basée sur le contenu des CI	132
6.1.3	Anticiper les besoins des utilisateurs	134
6.2	Présenter les recommandations	136
6.2.1	Explication des recommandations	136
6.2.2	Les actions des utilisateurs	140
6.3	Discussions	142

Dans les chapitres précédents, nous avons présenté deux résultats majeurs. Premièrement, nous avons montré qu'il est possible de modéliser l'évolution de la diversité au cours du temps dans le parcours de consultations des utilisateurs. Deuxièmement, en nous servant de cette modélisation, nous avons défini une nouvelle famille de contextes dits implicites. Nous avons montré que les changements de contextes implicites et les changements de contextes explicites sont souvent liés. Suite à ce constat, nous avons alors avancé l'hypothèse selon laquelle l'exploitation des contextes implicites peut constituer une approche permettant de fournir des recommandations adaptées aux contextes explicites des utilisateurs sans connaissance de ces derniers. Dans ce chapitre, nous présentons différentes approches permettant d'exploiter les résultats qui ont été présentés jusqu'à présent. Dans un premier temps, nous nous intéressons à la manière dont les contextes implicites et l'évolution temporelle de la diversité peuvent être utilisés pour calculer les recommandations. Dans un second temps, nous montrons que les contextes implicites et la diversité permettent de fournir facilement des explications quant aux recommandations proposées. Nous expliquons également comment un système de *feedback* adéquat peut réduire les ambiguïtés dues à certains comportements des utilisateurs. Enfin, ce chapitre est conclu par une discussion dans laquelle nous confrontons nos approches de recommandation par rapport aux dimensions que doivent prendre en compte les SR.

6.1 Approches de recommandation

Avant de présenter les différentes approches de recommandation, nous précisons que nous sommes conscients que notre modèle possède l'inconvénient de ne pouvoir s'adapter automa-

tiquement à une modification/évolution du contexte explicite de l'utilisateur. Par exemple, si un utilisateur court en écoutant une musique possédant une énergie forte et qu'il s'arrête pour marcher, un SR contextuel exploitant l'activité physique de l'utilisateur pourrait détecter ce changement de rythme physique et proposer une musique plus calme par la suite⁷⁷. Le modèle *DANCE*, quant à lui, ne prend pas en considération (et ne cherche pas à y parvenir) ce changement de contexte explicite et ne pourra donc pas s'adapter instantanément à cette évolution. Cependant, si l'utilisateur choisit lui-même d'écouter une musique calme, ce changement en termes de diversité pourra être détecté et les musiques suivantes pourront alors être adaptées en conséquence. Ce délai est le prix à payer pour préserver la vie privée des utilisateurs en ne récoltant aucune information sur leur contexte explicite. Bien que l'ensemble de l'historique de l'utilisateur puisse être analysé, nous rappelons que seuls les derniers items consultés, leur ordre et leurs caractéristiques sont exploités. Ces données ne sont en aucun cas croisées ou partagées avec un autre service et sont alors entièrement respectueuses de la vie privée des utilisateurs. Ainsi, même si nous ne pouvons pas nous adapter à un changement de contexte explicite sans intervention de l'utilisateur, nous considérons qu'il ne s'agit pas là d'un problème majeur (d'autant plus que cet aspect ne fait pas partie de nos objectifs de recherche). Nous soutenons que le but de la recommandation est d'assister les utilisateurs mais avant tout, de faire en sorte que les recommandations soient adaptées à leurs envies et certainement pas le contraire. Dans l'exemple précédent où la musique est automatiquement adaptée au contexte de l'utilisateur, ce dernier n'a pas choisi s'il voulait ou non adapter la musique à son nouveau contexte. À ce propos, [Cranor, 2005] explique que le fait de ne s'adapter qu'à des changements initiés par l'utilisateur est un critère important qu'un SR se doit de respecter. Si cette musique ou ce changement de musique devait ne pas lui plaire, il devra lui-même agir pour corriger cette action. Effectivement, en plus d'obliger l'utilisateur à réagir pour corriger le système, cela peut également avoir pour effet de complexifier l'interprétation des retours d'expérience (passer à la musique suivante ou rechercher un autre titre manuellement) de l'utilisateur puisque le SR ne saura pas si : la transition était inadaptée, la musique était non souhaitée dans ce contexte... Inversement, en nous basant uniquement sur les choix des utilisateurs afin de proposer les items les plus adaptés aux caractéristiques des items du contexte implicite, nous nous assurons d'être au plus près de leurs besoins.

Forts de ce constat, nous présenterons plusieurs approches de recommandation fondées uniquement sur le contexte implicite courant de l'utilisateur. Ces approches s'inscrivent dans les deux principales approches de recommandation que nous avons présentées lors de l'état de l'art. La première approche est basée sur la recommandation sociale et la seconde approche s'intéresse au contenu des items. Enfin, nous verrons de quelle manière nous entendons nous servir de l'historique de l'utilisateur afin d'anticiper certaines de ses actions.

6.1.1 Recommandation collaborative à base de CI

De la même manière que les approches collaboratives exploitent les préférences connues d'une communauté d'utilisateurs afin de prédire les préférences non connues des utilisateurs et déterminer leurs recommandations, nous entendons exploiter les contextes implicites de l'ensemble d'une communauté d'utilisateurs. L'approche que nous proposons ici consiste à rechercher dans

77. Dans cet exemple, nous nous situons dans le cas où l'utilisateur écoute sa musique en mode radio, c'est-à-dire que le système choisit à chaque fin de musique la musique suivante. Cette recommandation peut être proposée de deux manières, soit parce que le système possède la connaissance selon laquelle une musique calme est adaptée à la marche, soit parce que les musiques écoutées dans des contextes de "marche" sont majoritairement calmes.

l'ensemble des contextes implicites anonymisés d'un corpus, ceux contenant les items présents dans le contexte implicite courant de l'utilisateur que l'on cherche à recommander. Nous appelons S la liste composée des items du contexte implicite courant. S est donc modifiée à chaque nouvelle consultation de l'utilisateur : une consultation prolongeant le contexte implicite courant aura pour conséquence d'ajouter un item à S , alors qu'un changement de contexte aura pour conséquence de créer une nouvelle liste S ne contenant que l'item de ce nouveau contexte implicite. Une fois que S est défini, l'ensemble des contextes implicites sont testés afin de retrouver ceux contenant les items présents dans S . Les contextes implicites contenant les items de S sont appelés contextes implicites cooccurrents et les items de ces contextes qui sont différents des items de S sont appelés items cooccurrents. Ce sont ces items cooccurrents qui pourront être recommandés à l'utilisateur courant. Compte tenu du fait que le nombre d'items cooccurrents peut être important, une sélection des items à recommander doit être opérée. En effet, le nombre de recommandations à proposer aux utilisateurs est limité et les items retenus peuvent être nombreux, de plus, il faut s'assurer que ces items peuvent plaire à l'utilisateur courant. Ce processus de sélection dépend donc de nombreux facteurs (préférences utilisateur, nouveauté des items recommandés, diversité...) mais également du type de recommandation que l'on souhaite fournir. Par exemple, dans le cas où l'on cherche à recommander uniquement les items les plus consultés dans des contextes similaires, seuls les n -items les plus fréquents seront choisis. Dans le cas où l'on cherche à diversifier au mieux la liste de recommandations, l'ensemble des n -items choisis devra alors maximiser la diversité contenue dans la liste de recommandations (en exploitant une métrique telle que la diversité intra-liste [Ziegler et al., 2005]). Nous présentons dans l'Algorithme 2 le pseudo-code de cette approche de recommandation et dans la Figure 6.1 un schéma résumant son fonctionnement.

Algorithme 2 Pseudo-code simplifié de l'approche de recommandation collaborative basée sur les CI

lgorithme

```

items_cooccurrents, recommandations, S = ∅
pour tous les items  $i$  du Contexte Implicite courant faire
    ajouter  $i$  à  $S$ 
fin pour
pour tous les contextes implicites CI faire
    si  $S \subset CI$  alors
        pour tous items  $j$  de CI faire
            ajouter  $j$  à items_cooccurrents
        fin pour
    fin si
fin pour
Ordonner(items_cooccurrents)
recommandations = Selectionner( $n$ , items_cooccurrents)
retourner recommandations

```

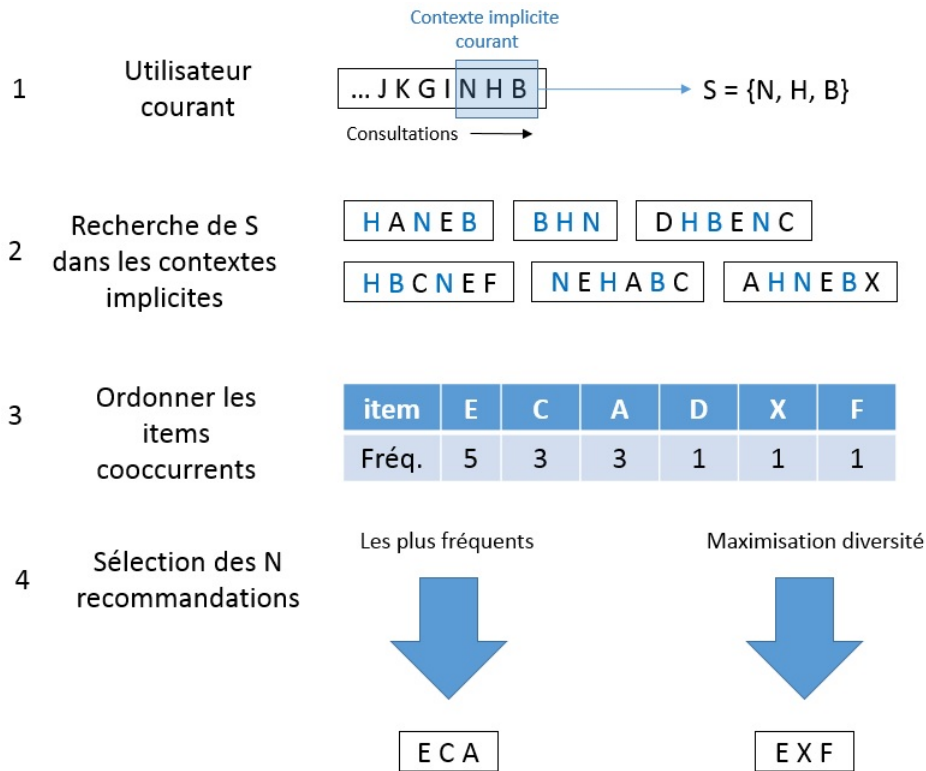


FIGURE 6.1 – Schéma du fonctionnement de l’approche de recommandation collaborative basée sur les CI

Lors de la phase de recherche des contextes implicites cooccurents, il est possible d’utiliser des critères de sélection différents. Par exemple, dans l’Algorithme 2 et dans la Figure 6.1, un contexte implicite est cooccurent s’il contient l’ensemble des items de S et ce sans que l’ordre des items de S soit pris en compte. De la même manière que [Bonnin and Jannach, 2014] ont utilisé l’ordre des items, c’est-à-dire la contrainte de contiguïté, comme un paramètre dans le cadre de la génération automatique de *playlists* de musique, il est possible d’appliquer ce paramètre à cette approche. Il est également possible de retirer un ou plusieurs items à S afin de créer une sous-liste S' permettant d’augmenter la probabilité de trouver des contextes implicites cooccurents. En effet, la contrainte étant moins importante dû au fait que la liste S' contient moins d’items (par rapport à S), la probabilité de trouver des contextes implicites contenant S' est alors plus importante. Il est également possible de considérer qu’un contexte implicite est cooccurent s’il contient un certain pourcentage d’items de S plutôt que l’ensemble des items S . Afin de vérifier s’il est effectivement possible de trouver des items cooccurents dans un ensemble de contextes implicites, nous avons exécuté l’Algorithme 2 sur nos deux corpus de données et sur les deux versions : la version “restreinte” ne contenant que des sessions sans information manquante et la version “brute” composée de l’ensemble des consultations. Dans un premier temps, les corpus de données que nous avons utilisés sont les corpus restreints (cf. *supra* Tableau 4.1 p. 80) dont les statistiques sont rappelées dans le Tableau 6.1 ci-dessous⁷⁸. Lors de l’exécution de cet algorithme, nous n’avons pas tenu compte de l’aspect temporel dans le sens où le contexte implicite pour lequel nous recherchons des items cooccurents (c’est-à-dire celui à partir duquel

⁷⁸. Les valeurs concernant les contextes implicites correspondent aux valeurs obtenues dans Tableau 4.9 p. 89, avec le formule de détection en temps réel.

S est défini) peut être antérieur aux contextes implicites dans lesquels nous allons effectuer nos recherches.

TABLE 6.1 – Rappel des statistiques des deux corpus ne contenant pas d’information manquante

	Lastfm	Deezer
# Utilisateurs	99	2 454
# Musiques	33 477	51 809
# Artistes	5 063	7 503
# Consultations	183 374	249 042
# Contextes implicites	55 178	69 414

TABLE 6.2 – CI et items cooccurrents retrouvés en fonction de la taille de S pour les corpus ne comprenant pas d’information manquante

Corpus	$ S $	# S	# $S \subset CI$	% $S \subset CI$	Moy. $CI_{Coo.}$	Moy. $Items_{Coo.}$
Lastfm	1	55 178	48 510	87,92	15,39	21,26
	2	42 552	20 020	47,05	8,76	12,07
	3	25 417	9 808	38,59	8,30	12,53
	4	15 168	5 949	39,22	8,50	13,00
	5	10 189	3 748	36,78	8,98	12,92
	6	7 054	2 563	36,33	8,59	12,76
	7	5 033	1 665	33,08	5,61	13,79
	8	3 697	1 200	32,46	5,20	13,51
	9	2 785	855	30,70	5,27	13,25
	10	2 146	649	30,24	5,35	12,99
Deezer	1	69 414	61 380	88,43	123,91	62,01
	2	54 635	22 429	41,05	101,13	32,21
	3	32 493	12 254	37,71	94,53	26,67
	4	19 839	7 963	40,14	36,96	29,55
	5	12 472	5 004	40,12	34,45	32,07
	6	8 732	3 706	42,44	25,43	33,24
	7	6 393	2 694	42,14	21,25	34,96
	8	4 940	2 049	41,48	20,04	37,26
	9	3 952	1 609	40,71	19,21	39,82
	10	3 196	1 245	38,95	19,03	43,60

Dans le Tableau 6.1, nous rappelons les caractéristiques des deux corpus “restreints” et dans le Tableau 6.2, nous présentons les résultats des recherches d’items cooccurrents sur les deux corpus de données en fonction de différentes tailles de S . Les abréviations utilisées dans le Tableau 6.2 sont définies comme suit :

- $|S|$: nombre d’items contenus dans S ;
- # S : nombre de fois où S contient un nombre d’items équivalent à S ;
- # $S \subset CI$: nombre de fois où au moins un contexte implicite cooccurrent dont la taille est supérieure à la taille de S est trouvé.
- % $S \subset CI$: pourcentage de fois où au moins un contexte implicite cooccurrent dont la taille est supérieure à la taille de S est trouvé par rapport à # S . Autrement dit, cette information

nous donne la proportion des situations pour lesquelles il est possible d’obtenir un item à recommander par rapport à S ;

- $Moy. CI_{Coo.}$: nombre moyen de contextes implicites cooccurrents contenant l’ensemble des items de S ;
- $Moy. Items_{Coo.}$: nombre moyen d’items présents dans les contextes cooccurrents qui sont différents des items contenus dans S . De plus, ces items sont tous uniques, il n’est question dans ce cas que de connaître le nombre d’items concurrents différents qui ont pu être retrouvés.

Étant donné que nous nous intéressons aux items cooccurrents, seuls les CI cooccurrents dont la taille est supérieure à la taille de S sont comptabilisés. Les CI cooccurrents de tailles identiques à la taille que S ne permettent pas de dégager des items à recommander étant donné qu’ils sont égaux à S .

Nous remarquons dans un premier temps que le nombre de fois où S est de taille 1 correspond également au nombre de contextes implicites total. En effet, à chaque nouveau contexte implicite, S est réinitialisé et n’est constitué que du premier item du nouveau contexte implicite. Dans un second temps, nous pouvons constater que des items cooccurrents sont trouvés dans plus de 30 % des cas et ce même si la taille de S est égale à 10. Bien que, dans ce tableau, la taille de S soit limitée à 10, il est tout à fait possible d’augmenter ce seuil, et ce, jusqu’à la taille maximum des contextes implicites. Nous avons choisi une taille de 10, car à partir de cette valeur, les contextes implicites concernés représentent moins de 5 % des cas. Nous pouvons par ailleurs constater que plus la taille de S est faible, plus le nombre de contextes cooccurrents trouvés est important. En effet, comme nous l’avons précisé précédemment, plus la contrainte est faible, et plus la probabilité de trouver des contextes implicites cooccurrents est importante. Bien que le pourcentage de fois où des contextes implicites sont trouvés diminue avec l’augmentation de la taille de S , le fait que lorsque cette dernière est de 1, des contextes implicites cooccurrents soient retrouvés dans plus de 87 % des cas est très encourageant. En effet, outre le fait de savoir que pour quasiment tout nouveau contexte implicite, des items cooccurrents sont retrouvés, cela signifie que dans le cas où aucun contexte cooccurrent n’est trouvé pour une taille de S supérieure (à 1, c’est-à-dire lors de seconde consultation poursuivant ce contexte), il est possible de modifier S et dans le pire des cas, de ne considérer que le premier item consulté dans S ⁷⁹. Parmi les modifications possibles de S , il est possible de supprimer un à un les items de S (en commençant par les premiers items du contexte implicite courant, c’est-à-dire les plus “éloignés” de la consultation courante) jusqu’à trouver des items cooccurrents. Cette pratique s’apparente au principe du *Skipping* [Bonnin et al., 2012] qui consiste à retirer volontairement des items dans le cadre de la recherche de motifs séquentiels. Afin de limiter les biais dus au fait de supprimer des items de S pour trouver des contextes implicites cooccurrents, il est également possible d’adapter l’Algorithme 2. Plutôt que de ne considérer qu’un seul ensemble d’items S et les items cooccurrents qui leurs sont associés, il est possible de considérer différentes variantes de S et de fusionner les résultats obtenus en ne conservant par exemple que les items qui sont présents dans les différentes listes d’items cooccurrents. Par exemple, si nous prenons le cas où $S = \{2, 3, 4, 5\}$ et pour lequel aucun contexte/item cooccurrent n’est trouvé, il est possible de procéder comme suit :

- (étape 1) créer toutes les sous-listes de S en enlevant un item à S . Dans notre cas, nous aurons donc $S'_1 = \{2, 3, 4\}$, $S'_2 = \{3, 4, 5\}$, $S'_3 = \{2, 4, 5\}$;

79. Le fait que dans 87 % de cas, des items cooccurrents sont retrouvés si S est de taille 1 n’est pas spécifique au items de début de contexte. Des scores similaires sont également obtenus lorsque des items ne se trouvant pas en début de contexte sont utilisés dans S pour $|S| = 1$.

- (étape 2) rechercher les items cooccurrents pour chaque liste S' ;
- (étape 3) conserver tous les items cooccurrents ou ne conserver que les items qui sont communs aux 3 listes d'items cooccurrents ;
- si nécessaire (aucun item cooccurrent n'a été trouvé), recommencer les étapes 2 et 3 en enlevant un item supplémentaire.

Mises à part les alternatives que nous venons de présenter pour trouver des contextes cooccurrents, nous pouvons également constater d'après les résultats présentés dans le Tableau 6.2 que le nombre moyen d'items cooccurrents est toujours au moins supérieur à 12 pour le corpus de Lastfm et supérieur à 25 pour le corpus de Deezer. Ainsi, si le nombre de recommandations présentées à l'utilisateur à chaque pas de temps est faible (3, 4 ou 5 items par cycle de recommandations), le nombre d'items cooccurrents trouvé est suffisamment important pour pouvoir procéder à une sélection de ces derniers (en fonction des préférences de l'utilisateur, de son historique de consultation...). Il est également intéressant de souligner que le nombre d'items cooccurrents trouvés n'est pas nécessairement inversement proportionnel à la taille de S . En effet, même si pour les deux corpus, le plus grand nombre d'items cooccurrents est obtenu pour une taille de S de 1, le nombre moyen d'items cooccurrents obtenus pour une liste S de taille 10 par exemple n'est pas le plus faible. De plus, pour le corpus de LastFm, le nombre moyen d'items cooccurrents retrouvés pour $|S| = 2$ est plus faible que pour $|S| = 3$. Pour le corpus de Deezer, le nombre moyen d'items cooccurrents pour $|S| = 3$ est plus faible que pour $|S| = 9$. Ce phénomène s'explique par le fait que les contextes cooccurrents contenant les items de S sont en moyenne plus petits (et contiennent donc moins d'items cooccurrents) que les contextes cooccurrents contenant les items de S au pas de temps suivant (si le contexte implicite courant est prolongé), contenant alors un item supplémentaire. Troisièmement, nous pouvons constater qu'il existe des différences importantes entre les deux corpus. Ces différences concernent tant le nombre de contextes cooccurrents que le nombre d'items cooccurrents retrouvés. Suite aux analyses complémentaires que nous avons effectuées, nous avons constaté que ces différences sont dues aux fréquences d'écoutes des musiques. En d'autres termes, le nombre de fois où les musiques sont écoutées de nombreuses fois est un critère primordial dans le nombre de contextes implicites cooccurrents retrouvés. Le Tableau 6.3 résume les nombres de consultations d'items identiques de chacun des 2 corpus.

TABLE 6.3 – Nombres d'écoutes d'une même musique sur les deux corpus "restreints"

Bornes		Nombre d'items	
Inférieure (\geq)	Supérieure ($<$)	Lastfm	Deezer
1	250	38 471	51739
250	500	4	34
500	750	2	17
750	1000	-	8
1000	1250	-	4
1250	1500	-	2
1500	∞	-	5

Nous pouvons constater dans le cas du corpus de Deezer que plus de 19 musiques ont été consultées plus de 750 fois, dont 5 étant consultées plus de 1500 fois (ce correspond alors à au moins 7 500 consultations). En revanche, dans le corpus de LastFm, aucune musique n'a été consultée plus de 750 fois. Les musiques possédant un nombre de consultations importantes augmente le nombre de contextes cooccurrents retrouvés S , si elles sont contenues dans S . Ainsi,

TABLE 6.4 – Rappel des statistiques des deux corpus “bruts”

	Lastfm	Deezer
# Utilisateurs	99	2 903
# Musiques	168 084	204 434
# Artistes	17 803	35 326
# Consultations	1 980 587	1 856 552
# Contextes implicites	547 423	574 550

lorsque qu’une musique consultée plus de 1 000 fois est la seule musique contenue dans S , nous trouverons 1000 contextes concurrents. En effet, la seule contrainte dans ce cas pour qu’un contexte implicite soit cooccurrent est qu’il contienne l’item contenu dans S . De plus, ce cas ne se reproduira pas qu’une seule fois pour chaque musique, mais à chaque fois que cette musique constituera le premier item d’un contexte implicite. Ainsi, ces nombres importants de contextes implicites cooccurrents retournés par des items consultés fréquemment tendent à augmenter le nombre moyen de contextes implicites cooccurrents retrouvés pas l’ensemble des items du corpus. Plus le corpus de données est important (en termes de nombre de consultations) et plus les nombres de consultations de mêmes items sont importantes, plus les contextes et les items cooccurrents retrouvés seront nombreux. Le nombre d’items candidats pouvant être recommandé sera d’autant plus important.

Dans le Tableau 6.5, nous reportons les performances obtenues sur les corpus “bruts”. Un rappel des caractéristiques de cette version des corpus est présenté dans le Tableau 6.4 ci-dessus⁸⁰. Comme nous l’avons anticipé dans le paragraphe précédent, les contextes implicites cooccurrents et les items cooccurrents retrouvés sont effectivement plus nombreux. En effet, en comparant les données des corpus “restreints” par rapport aux corpus “bruts” résumés dans le Tableau 6.4, nous pouvons constater que les corpus “bruts” possèdent de 10 fois plus de consultations alors que les nombres de musiques ne sont quant à eux multipliés que par 5. Les musiques sont plus fréquemment consultées en moyenne dans les corpus “bruts” que dans les corpus “restreints” et nous observons bien une augmentation des performances en termes de cooccurrences.

80. Les valeurs concernant les contextes implicites correspondent aux valeurs obtenues dans Tableau 4.15 p. 94, obtenus avec le formule de détection en temps réel.

TABLE 6.5 – CI et items cooccurrents retrouvés en fonction de la taille de S pour les corpus “bruts”

Corpus	S	# S	# S ⊂ CI	% S ⊂ CI	Moy. CI _{Co.}	Moy. Items _{Co.}
Lastfm	1	547 423	531 410	97,07	64,46	137,15
	2	432 160	205 187	47,48	13,35	31,98
	3	274 138	102 435	37,37	12,03	32,15
	4	174 065	66 362	38,12	11,94	32,77
	5	126 265	49 099	38,89	11,27	31,28
	6	94 934	36 757	38,72	8,49	29,31
	7	72 314	27 471	37,99	8,01	28,57
	8	5691	21 668	37,56	7,36	27,55
	9	46 500	16 886	36,31	6,87	26,78
	10	38 504	12 987	33,73	6,53	26,21
Deezer	1	574 550	545 322	94,91	291,23	364,61
	2	438 030	192 218	43,88	44,57	65,53
	3	252 705	74 211	29,37	38,73	58,49
	4	141 550	40 432	28,56	39,70	60,71
	5	91 152	26 979	29,60	38,63	60,34
	6	63 163	18 285	28,95	39,63	65,90
	7	45 253	13 651	30,17	40,85	68,81
	8	34 466	10 621	30,82	40,96	71,36
	9	27 750	8 580	30,92	42,08	74,31
	10	22 554	7 029	31,16	42,68	75,43

Nous avons présenté cette première approche de recommandation dans sa configuration la plus simple. Il est néanmoins possible d’envisager de nombreuses alternatives afin d’améliorer certains aspects tels que la personnalisation ou encore le passage à l’échelle. Par exemple, les musiques possèdent la particularité d’être consultées de nombreuses fois par un même utilisateur. Il est alors possible de donner davantage de poids aux musiques issues de contextes implicites de l’utilisateur courant par rapport aux contextes implicites des autres utilisateurs. Ce poids pourra être adapté à chaque utilisateur en fonction de sa propension à réécouter des musiques qu’il connaît déjà. Outre le fait de privilégier les CI de l’utilisateur courant, il est également possible de privilégier les utilisateurs les plus similaires/proches de l’utilisateur courant en utilisant par exemple des méthodes de *clustering* appliquées en amont aux utilisateurs. Pour des raisons de passage à l’échelle, il est également possible d’utiliser un nombre restreint d’utilisateurs. Étant donné le nombre important de contextes implicites pouvant être isolés, nous pouvons anticiper le fait qu’un système possédant plusieurs milliers voire millions d’utilisateurs⁸¹ ne pourra pas parcourir l’ensemble des contextes implicites en un temps suffisant pour que les recommandations soient fournies en temps réel. Limiter le nombre de CI (déterminé par les utilisateurs voisins choisis) est un moyen de réduire les temps de calcul.

La principale limite de cette approche réside dans le fait que les items à recommander sont dépendants des items initiaux, *i.e.* les items du contexte courant de l’utilisateur que l’on cherche à recommander. Si ces items sont peu ou pas du tout consultés par la communauté, trouver des recommandations sera alors difficile, voire impossible. Cette faible popularité peut s’expliquer soit

81. Deezer : 14 millions d’utilisateurs (<https://www.deezer.com/fr/company>), Spotify 170 millions d’utilisateurs (<https://techcrunch.com/2018/06/01/apple-needs-to-play-nice-with-spotify/>), AppleMusic 50 millions d’utilisateurs (<https://marketrealist.com/2018/06/apple-music-subscribers-reach-50-million>).

par le fait que ces items sont nouveaux dans le catalogue d'items et qu'ils n'ont pas été consultés, soit parce que ces items ne sont consultés que très peu consultés (problème de la longue traîne). Cette situation est similaire aux cas du démarrage à froid ou encore du problème de couverture du catalogue d'items que nous avons évoqués dans la Section 1.1. Cette contrainte peut parfois être contournée. Si aucun contexte implicite contenant l'ensemble des items de S n'est trouvé, il est possible d'effectuer une nouvelle recherche à partir d'un sous-ensemble S' de S . Étant donné que la contrainte de sélection sera moins importante, la probabilité de trouver des contextes implicites contenant S' sera plus grande. Tant que le nombre d'items retrouvés est insuffisant, il est possible de procéder à une modification de S' , en enlevant un item supplémentaire par exemple. L'utilisation des attributs des items et des contextes implicites tels que nous allons la présenter dans la section suivante permet également de pallier plus efficacement cette limite. En effet, pour les raisons que nous venons de citer, il peut être complexe de trouver des items cooccurents. Or, l'utilisation des attributs permet d'augmenter l'espace de recherche et donc d'augmenter la probabilité de trouver des items relatifs aux contextes courants des utilisateurs.

6.1.2 Recommandation basée sur le contenu des CI

Contrairement à l'approche précédente qui n'exploite pas les attributs des items pour déterminer les items à recommander (hormis la phase de sélection où il est possible d'utiliser les attributs pour diversifier la liste de recommandations), cette approche se base sur les caractéristiques du CI courant. Cette approche s'intéresse à 2 points en particulier : adapter les recommandations par rapport aux caractéristiques du contexte implicite courant et adapter les recommandations à l'évolution de la diversité.

Adaptation aux caractéristiques du contexte implicite courant

Selon la définition que nous avons proposée dans la Section 4.1, les items partagent des caractéristiques communes au sein d'un contexte implicite. Ainsi, il est possible d'exploiter les attributs de ces items pour déterminer quelles doivent être les caractéristiques des items à recommander, pour que ces derniers s'accordent au contexte implicite courant. Afin d'y parvenir, il est nécessaire d'établir dans un premier temps le profil du contexte implicite courant. Pour cela les attributs des items peuvent permettre d'établir une représentation/caractérisation de chaque contexte implicite. Dans le domaine de la musique, des travaux similaires ont été réalisés sur la base de *playlists* [Jannach et al., 2014, Pichl et al., 2016]. Par exemple, il est possible de calculer la moyenne de chaque attribut numérique, de calculer les fréquences des termes des attributs sous forme de listes de mots ou encore d'utiliser toute autre représentation statistique permettant de synthétiser l'information disponible. Dans un second temps, connaissant les caractéristiques du contexte implicite courant, il est alors possible de proposer différentes stratégies de recommandation. Une première approche consiste à recommander des items pour lesquels les attributs correspondent aux caractéristiques du contexte courant. Autrement dit, cela revient à rechercher les items qui n'apportent pas ou peu de diversité par rapport au contexte courant. Par exemple, cette tâche peut être réalisée avec des approches visant à déterminer les k plus proches voisins d'un ensemble d'items. Il peut également être intéressant d'emprunter des techniques de recommandations basées sur le contenu [Lops et al., 2011]. Dans ce cas, les items ne sont plus recommandés à un utilisateur, mais à un contexte implicite. Autrement dit, le contexte implicite devient le profil utilisateur pour lequel les recommandations sont calculées. La seconde approche consiste à utiliser les caractérisations de l'ensemble des contextes implicites du système afin de déterminer les contextes qui sont similaires au contexte courant. Cette approche est donc semblable dans son processus de recherche d'items à recommander à l'approche collaborative que

nous avons présentée précédemment. C'est-à-dire qu'à partir d'un ensemble de critères, nous recherchons les autres CI qui satisfont ces critères. Cependant, le principe de sélection des recommandations ne repose plus sur des cooccurrences d'items, mais sur des similarités entre les contextes implicites. Ainsi, le problème du démarrage à froid discuté dans l'approche précédente et la complexité combinatoire n'ont plus lieu d'être, étant donné qu'il est toujours possible de trouver des contextes implicites plus ou moins similaires par rapport au contexte courant. De plus, l'idée d'utiliser les similarités entre contextes a déjà été présentée dans le cas des contextes explicites [Zheng et al., 2015a] et une amélioration des performances a été observée. Appliquer cette idée dans le cadre des contextes implicites est donc très prometteur.

Adaptation à l'évolution de la diversité

La diversité peut être calculée à plusieurs niveaux : pour chaque attribut, pour un sous-ensemble d'attributs ou alors pour l'ensemble des attributs disponibles pour les items. Dans nos formules de détection de contextes implicites, nous avons utilisé la moyenne (potentiellement pondérée) de l'ensemble des diversités de chacun des attributs. Bien que cette valeur nous indique si l'utilisateur diversifie ou spécialise ses consultations de manière générale, s'intéresser aux évolutions de diversité à l'échelle des attributs est nécessaire afin de fournir des recommandations pertinentes. Par exemple, même si nous constatons que l'utilisateur diversifie ses consultations, tout item apportant de la diversité n'est pas forcément souhaitable. La diversité doit être apportée sur les attributs qui ont conduit à cette diversité et non sur les autres, car l'utilisateur pourrait ne pas comprendre le choix du SR. Un utilisateur qui se trouve dans une soirée dansante souhaitera par exemple diversifier les genres de musiques et les types d'instruments (*instrumentalness*), mais conserver tout de même une forte *danceability*. Dans ce cas, si l'utilisateur n'a pas écarté certaines caractéristiques (voir Section 6.2.2), tout item possédant une forte *danceability* est potentiellement une recommandation pertinente. Un autre exemple est illustré par la Figure 6.2 dans laquelle nous représentons un contexte implicite dans le cas d'un parcours de consultations d'offres d'emploi. Bien que le domaine de la musique soit utilisé dans les travaux présentés dans ce manuscrit, notre modèle peut être utilisé dans de nombreux domaines applicatifs (*cf. supra* Chapitre 2), p.47). Nous pouvons constater que, même si la diversité générale n'augmente pas de manière significative à la 5^{ème} consultation, la diversité concernant la localisation a fortement augmenté. En termes de recommandation, il faudra dans ce cas veiller à conserver les valeurs des autres attributs tout en maximisant la diversité de l'attribut localisation (en accord avec ce que nous connaissons des préférences de l'utilisateur).

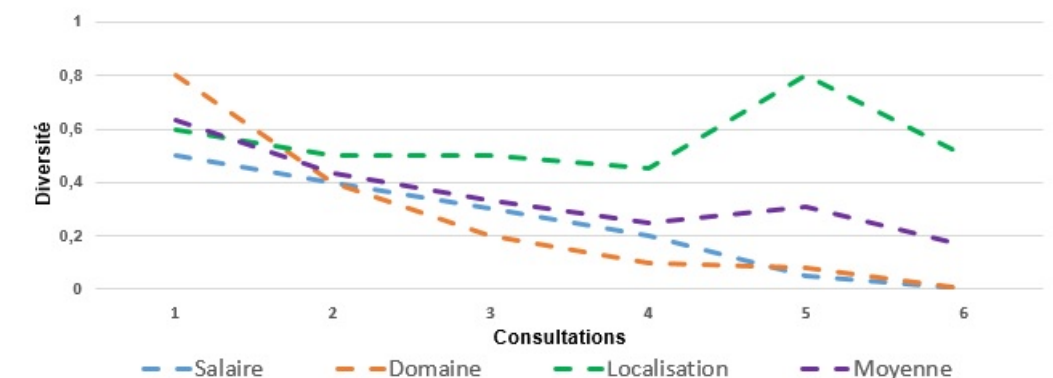


FIGURE 6.2 – Cas d'un attribut variant fortement au sein d'un contexte implicite dans le parcours d'un utilisateur sur un réseau social professionnel

6.1.3 Anticiper les besoins des utilisateurs

Notre modèle n'est pas en mesure de s'adapter à un changement de contexte explicite instantanément comme pourrait le faire par exemple un SR contextuel analysant les moindres changements de conditions contextuelles. Malgré cela, il est tout de même possible de compenser ce délai en anticipant un changement de contexte implicite. Ainsi, à défaut de posséder des informations nous indiquant que l'utilisateur modifie son contexte explicite, nous pouvons modéliser le comportement de l'utilisateur dans le but d'anticiper certaines de ses actions. Nous rappelons qu'un contexte implicite peut être représenté/caractérisé par les attributs des items qui le composent. Dès lors, les contextes implicites peuvent être synthétisés sous cette forme et deux contextes ne possédant pas les mêmes items peuvent tout de même partager les mêmes caractéristiques. En utilisant des techniques d'apprentissage automatique, il est alors possible de comprendre et de prédire la manière dont les CI s'articulent entre eux à l'échelle d'un utilisateur ou encore de l'ensemble de la communauté d'utilisateurs. Les probabilités de transitions entre les différents CI peuvent être déterminées et exploitées en vue de fournir des recommandations. Le but visé sera de pouvoir anticiper le changement de contexte d'un utilisateur en proposant des items appartenant au contexte suivant le plus probable ou alors de prolonger le contexte courant de l'utilisateur. Cependant, une représentation uniquement basée sur les attributs peut ne pas suffire à définir l'état à un instant t d'un système probabiliste. Deux contextes implicites peuvent avoir les mêmes caractéristiques et avoir des tailles différentes. Or, la probabilité de transition ne sera pas la même pour un contexte implicite selon que ce dernier contient 2 ou 5 items. Par exemple, la probabilité de transition peut être plus importante lorsque de très peu d'items ou alors de nombreux items sont contenus dans un CI. Dans le premier cas, des consultations d'items différents pourront donner lieu à des contextes implicites de caractéristiques différentes sans pour autant créer un nouveau CI (la diversité apportée n'est pas suffisante) alors que dans le second cas, il est question de transition vers un nouveau contexte implicite étant donné que le contexte courant comporte de nombreuses consultations. Ainsi, compte tenu des caractéristiques du contexte implicite courant et donc des probabilités associées de transitions pour les CI suivants, les recommandations seront adaptées afin de couvrir les différentes éventualités. En fonction des choix effectués par l'utilisateur parmi les items proposés, le SR devra ensuite mettre à jour ses probabilités et adapter de nouveau les items à recommander en fonction des choix de l'utilisateur. De plus, les attributs des items choisis permettront de comprendre davantage les besoins de l'utilisateur.

Parmi les approches probabilistes existantes, les processus Markoviens sont adaptés à notre problème étant donné qu'ils permettent de modéliser des problèmes de décisions séquentielles. En outre, l'utilisation de modèles de Markov afin de fournir des recommandations a déjà été explorée [Shani et al., 2005, Hariri et al., 2015]. De plus, étant donné que le contexte implicite de l'utilisateur peut être influencé par son état mental qui peut changer à tout moment, nous soutenons que les choix des utilisateurs ne dépendent que du moment présent. Dans notre cas, le problème est modélisé comme suit : à chaque contexte implicite $C_n \in C$ où C représente l'ensemble des caractéristiques permettant de représenter un CI, nous pouvons associer des probabilités de transitions pour les états suivants qui sont :

- prolonger le CI courant ;
- changer de CI.

La Figure 6.3 propose un exemple de modélisation des transitions de contextes implicites d'un utilisateur. Les nœuds représentent les contextes implicites pour lesquels nous connaissons les caractéristiques. Ils sont résumés par une lettre et leur taille. Les nœuds d'une même couleur représentent des items appartenant aux mêmes contextes implicites. Par exemple C_z & $t = 3$

signifie que le contexte implicite possède les caractéristiques z et qu'il contient 3 items. Les arcs représentent les probabilités de transition entre les différents CI. Dans notre cas, nous nous trouvons dans le cas d'un processus Markovien et non d'une chaîne de Markov dont les propriétés requièrent de se trouver dans un espace d'états finis. Autrement dit, à moins de définir/fixer l'ensemble des caractéristiques des CI ainsi que le nombre d'items les composants (car un état résulte dans notre cas de la composition de ces deux dimensions), nous ne pouvons pas parler de chaîne de Markov.

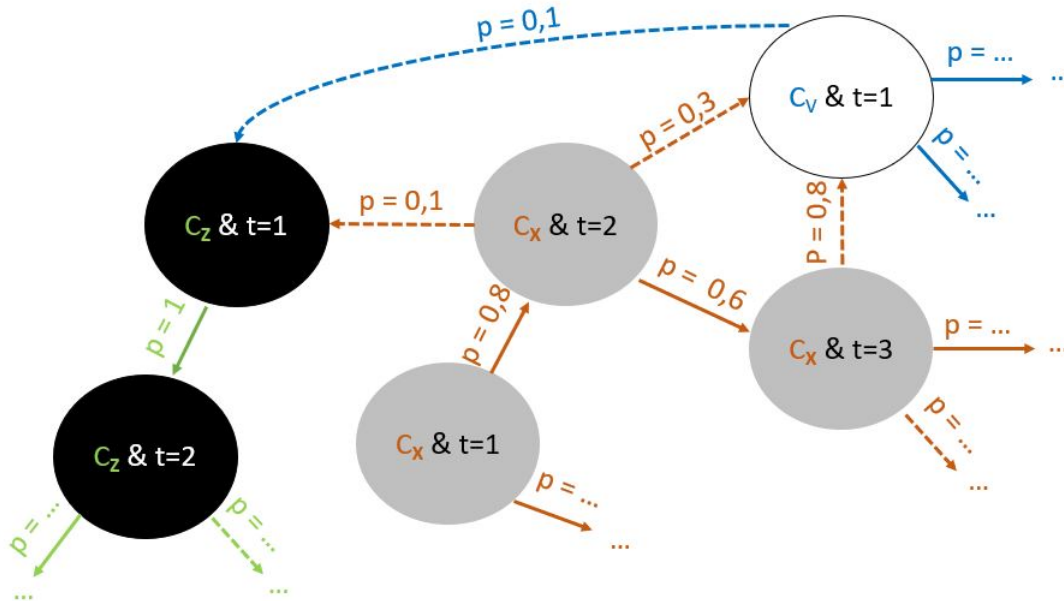


FIGURE 6.3 – Représentation des transitions de CI à l'aide d'un processus Markovien

Sur cette figure, nous pouvons distinguer les deux transitions susmentionnées : la prolongation du contexte implicite courant (représenté en trait plein) et le changement de contexte implicite (représenté en pointillé). La connaissance des probabilités de transition permet d'adapter les recommandations en fonction des actions les plus probables de l'utilisateur. Cette information peut être utilisée pour calculer l'ensemble ou seulement une partie de la liste qui sera recommandée à l'utilisateur. Il est alors possible de ne recommander que des items correspondant au contexte implicite suivant le plus probable ou de recommander des items appartenant à différents contextes en fonction de leurs probabilités d'apparition respectives. Par exemple, dans le cas de la Figure 6.3, si l'utilisateur se trouve dans le contexte C_x de taille 2, les probabilités de transitions sont les suivantes :

- prolonger le contexte courant ($p = 0,6$) ;
- arrêter le contexte courant pour un contexte C_v ($p = 0,3$) ;
- arrêter le contexte courant pour un contexte C_z ($p = 0,1$).

Si l'on répartit les items en fonction des probabilités et que la liste de recommandations contient 6 items, 3 items auront pour objectif de prolonger le contexte courant, 2 items appartiendront au contexte C_v et un item au contexte C_z . Si au contraire seul le CI ayant la probabilité la plus forte est utilisé, les 6 items proposés viseront à prolonger le contexte C_x .

Les changements de contextes implicites induisent cependant la particularité de créer des contextes implicites ne contenant qu’une seule musique et pour lesquels la caractérisation dépendra des musiques suivantes. De ce fait, déterminer les caractéristiques d’un nouveau contexte peut être plus délicat en fonction de la manière dont les contextes implicites sont caractérisés. En effet, en analysant l’évolution des caractéristiques des items, nous pouvons capturer une dynamique que l’utilisateur instaure au fur et à mesure de ses consultations. Par exemple, un utilisateur peut écouter des musiques possédant une “énergie” de plus en plus importante et c’est cette évolution d’énergie qui peut permettre de caractériser le CI. Or, lorsqu’il n’y a qu’une seule musique dans un CI, il n’est pas possible de capturer une telle dynamique et donc il ne sera pas possible de déterminer les caractéristiques de ce contexte implicite courant. Cependant, tout comme il est possible de déterminer les probabilités de transitions entre les contextes lorsqu’ils sont identifiés, il est également possible de déterminer les évolutions les plus probables d’un CI de taille 1 et donc d’adapter les recommandations en conséquence. Ainsi, il convient de traiter les CI de taille 1 de manière différente puisque la caractérisation de ces derniers est différente des CI contenant plusieurs items.

6.2 Présenter les recommandations

Le calcul des recommandations ne constitue qu’une partie du processus de recommandation. Une présentation adaptée des recommandations ainsi que la présence d’explications concernant les recommandations sont également des aspects importants. En effet, ces deux aspects influencent l’acceptation et l’adoption des recommandations par les utilisateurs [Herlocker et al., 2000, Tintarev and Masthoff, 2007, Jones, 2010, Sharma and Ray, 2016]. Dans cette section, nous aborderons uniquement l’aspect explication des recommandations, étant donné que la présentation des recommandations relève essentiellement du domaine de l’Interface Homme-Machine (IHM) que nous ne traitons pas dans ce manuscrit. Nous expliquerons dans un premier temps de quelles manières les contextes implicites ainsi que l’évolution temporelle de la diversité peuvent être mis à profit pour la génération d’explications. Dans un second temps, nous présenterons différentes stratégies permettant d’améliorer la compréhension du comportement des utilisateurs. Comme nous l’avons vu au travers de notre étude utilisateur, les attributs auxquels les utilisateurs semblent sensibles par rapport à leurs contextes explicites d’écoute ne sont pas les mêmes que les attributs permettant la détection des changements de CI. Par exemple, la popularité est l’attribut qui possède le 3^{ème} poids le plus important lorsqu’il s’agit d’optimiser le découpage en CI (*cf. supra* Tableau 4.19 p. 101) alors que cet attribut n’est lié qu’à une seule dimension contextuelle qui est l’émotion ressentie (*cf. supra* Tableau 5.6 p. 116). Ainsi, il est nécessaire de s’assurer que les dimensions qui seront utilisées dans la présentation des recommandations seront effectivement pertinentes pour les utilisateurs. Nous verrons également que l’obtention de certains retours des utilisateurs quant aux items recommandés peut faciliter la compréhension des besoins des utilisateurs.

6.2.1 Explication des recommandations

Les explications fournies avec les recommandations permettent d’améliorer l’utilité des systèmes de recommandation. Par exemple, [Pu et al., 2009] démontrent que l’introduction d’explications accompagnant les recommandations dans le cadre de SR à base de critiques permet de doubler le taux d’acceptation des recommandations par les utilisateurs. Si un utilisateur ne

comprend pas les raisons pour lesquelles certains items lui sont recommandés, il est probable que ce dernier n'utilise pas les recommandations et qu'il devienne également méfiant vis-à-vis du SR.



FIGURE 6.4 – Explications des recommandations sur le site Amazon ⁸²

Sur la Figure 6.4 nous pouvons voir que les recommandations sont accompagnées d'explications telles que : “Recommandé pour vous”, “Parce que vous avez acheté...”, “Parce que vous avez dit que vous aviez...”. Alors que la première phrase sert à présenter les recommandations, les deux suivantes expliquent à l'utilisateur sur quelles bases sont déterminées les recommandations. Ces explications possèdent une double utilité. Premièrement, elles permettent de montrer à l'utilisateur qu'il s'agit de recommandations personnalisées, car les explications visent directement les actions effectuées précédemment par l'utilisateur [Nart and Tasso, 2014, Ricci et al., 2015]. Deuxièmement, elles renseignent l'utilisateur sur les raisons pour lesquelles ces items lui sont recommandés (“Parce que vous avez acheté...”, “Parce que vous avez dit que vous aviez...”), ce qui permet d'apporter de la transparence [Ardissono et al., 2003, Tintarev and Masthoff, 2007], ainsi que de la confiance [Cramer et al., 2008, Al-Taie, 2013] chez l'utilisateur puisque ces deux dimensions sont liées. Par ailleurs, nous pouvons noter que le système utilise différentes traces d'usage afin de calculer les recommandations, il se base à la fois sur les items qui ont été achetés et sur les items possédés par l'utilisateur. Autrement dit, le système diversifie ses stratégies de recommandations ne sachant pas, à ce moment, s'il doit se focaliser uniquement sur ce qu'a acheté l'utilisateur ou sur ce qu'il dit posséder étant donné que ces deux dimensions peuvent révéler des profils utilisateurs différents. Par exemple, un utilisateur peut déclarer posséder des livres sur l'économie et n'avoir acheté que des livres de philosophie.

Malgré leur utilité avérée, les explications des recommandations sont la plupart du temps uniquement associées aux préférences connues des utilisateurs et ne sont pas exploitées à l'échelle des consultations des utilisateurs. Nous entendons par là que, tout comme le montre la Figure 6.4, même si les recommandations peuvent permettre à l'utilisateur de choisir un livre, elles ne sont pas basées sur ses besoins immédiats, et par extension sur son contexte implicite actuel. Même s'il existe une approche de recommandation basée sur les critiques permettant à l'utilisateur de spécifier ses critères, les explications fournies sont explicitement basées sur les critères que

82. <https://www.amazon.fr/>

l'utilisateur a précisés et donc n'apportent que peu de valeur ajoutée du point de vue explicatif [Pu et al., 2009, Chen and Pu, 2012]. Autrement dit, les explications fournies dans ce cas n'apportent, pas selon nous, de réel avantage étant donné qu'elles se basent sur les critères que l'utilisateur a lui-même spécifiés. De plus, d'après [Chen and Pu, 2012], qu'il soit basé sur les choix que l'utilisateur spécifie avant de commencer à utiliser le système (*user-initiated critiquing systems*) ou sur l'ensemble des critères sélectionnés au fur et à mesure des interactions avec le système (*system-suggested critiquing systems*), un système à base de critiques requiert un investissement conséquent. Par ailleurs, comme nous avons pu le voir dans nos travaux, les contextes implicites peuvent être multiples au sein d'une session et l'utilisateur devra dans ce cas préciser à chaque nouveau contexte les dimensions/critères qui sont importants pour lui. Enfin, nous n'avons pas trouvé dans la littérature de travaux s'intéressant à l'explication des recommandations lorsque ces dernières sont contextualisées. Comme le mentionne [Zheng, 2016] dans son état de l'art sur les CARS, fournir des explications pour les systèmes de recommandation contextuels constitue un défi scientifique encore non résolu. Cependant, contrairement aux approches exploitant les facteurs contextuels, nous soutenons l'idée selon laquelle les contextes implicites peuvent facilement être utilisés afin de fournir des explications pertinentes pour les utilisateurs. Par exemple, dans le cas où un utilisateur écoute de la musique, nous pouvons imaginer des explications pour les approches par contexte explicite et par contexte implicite :

CONTEXTE EXPLICITE : parce que vous êtes au bord de la mer avec vos enfants.

CONTEXTE IMPLICITE : parce que vous écoutez des musiques calmes.

Premièrement, les explications fournies sur la base du contexte explicite peuvent surprendre l'utilisateur, car les informations concernent directement sa vie privée. Comme nous l'avons vu dans l'état de l'art, les SR contextuels exploitent de nombreuses informations qui, si elles étaient utilisées à des fins d'explications, montreraient alors à l'utilisateur que le système connaît quasiment ses moindres faits et gestes. Deuxièmement, cette explication n'apporte que peu d'aide à l'utilisateur si ce n'est que ce dernier a conscience que le système possède une connaissance sur son contexte et que les recommandations y sont alors adaptées. Cependant, cela n'est pas suffisant pour expliquer à l'utilisateur pourquoi ces items sont adaptés à son contexte. Bien que les explications permettent de donner une certaine transparence sur la manière dont sont calculées les recommandations, elles ne permettent pas pour autant de choisir entre les différents items proposés. Nous pensons que les seules explications permettant à l'utilisateur de choisir entre différentes recommandations contextualisées doivent porter sur les contenus des items et dans ce cas, les contextes implicites sont particulièrement adaptés. Les recommandations étant la plupart du temps constituées d'items que l'utilisateur ne connaît pas, il apparaît primordial de pouvoir lui permettre de faire la distinction entre ces derniers. De par leur capacité à isoler les périodes pendant lesquelles les caractéristiques partagées entre les items sont similaires, les contextes implicites peuvent servir de base aux explications. Un exemple d'explication dans le domaine de la littérature serait "d'autres livres sur l'Italie au XV^{ème} siècle" dans le cas où ces caractéristiques représentent le contexte implicite de l'utilisateur.⁸³ De même qu'un contexte implicite permet de donner des explications, une évolution particulière en diversité peut également permettre d'orienter l'utilisateur au travers de différentes recommandations. Par exemple, dans le domaine de la musique, la Figure 6.5 représente les cas dans lesquels les explications se basent sur le

83. Il est évident que cette explication n'est pas utile dans le cas où un utilisateur recherche explicitement des ouvrages sur l'Italie au XV^{ème} siècle. Cependant, en navigant sur un site au travers des recommandations, un utilisateur peut s'éloigner de sa première recherche, élargir ou encore restreindre ses critères de recherche. Il devient alors important de lui apporter des explications concernant son contexte implicite afin qu'il puisse comprendre ce que le système a interprété de ses besoins.

contexte implicite et sur l'évolution en diversité.



FIGURE 6.5 – Explications des recommandations sur le site Amazon

Dans le premier cas, la diversité globale est très faible, ce qui indique que l'utilisateur consulte des items très similaires entre eux. L'explication fournie vise alors à montrer que les besoins de l'utilisateur sont compris et que les recommandations sont adaptées aux attributs qui sont partagés entre les consultations appartenant à son contexte implicite courant. Dans le second cas, seule la diversité de la *danceability* est faible. Pour cette raison, les explications portent sur cette dimension qui semble importante chez cet utilisateur. En effet, les attributs autres que la *danceability* étant divers, il est alors souhaitable de conserver cette diversité tout en maximisant la valeur de l'attribut qui lie les consultations entre elles. La pierre angulaire de cette approche réside dans la capacité à pouvoir isoler les attributs qui sont pertinents de ceux que ne le sont pas. Ce phénomène sera d'autant plus complexe que le nombre d'attributs utilisés sera important. Cependant, le nombre important d'interactions existant entre les utilisateurs et les items constituent des bases d'apprentissage volumineuses sur lesquelles de nombreuses informations peuvent être extraites. Être en mesure de générer des explications de qualité nécessitera un temps d'apprentissage pour optimiser les différents paramètres intervenant dans notre modèle. Au cours de cet apprentissage, le système peut commettre des erreurs, ce qui ne fait pas nécessairement référence à un échec suite auquel un utilisateur cesserait d'utiliser le système mais à une prise de décision d'un utilisateur qui témoignerait du fait que les explications n'étaient pas adaptées. De plus, nous pouvons envisager l'utilisation d'outils permettant de savoir si les caractéristiques que nous avons isolées comme étant pertinentes le sont réellement. Par exemple, nous pouvons imaginer qu'après plusieurs consultations, une boîte de dialogue s'ouvre afin de demander à l'utilisateur si un attribut en particulier est important pour lui. En veillant bien évidemment à ne pas abuser de ce genre d'outil, il sera alors possible d'utiliser les réponses obtenues en déterminant par exemple que lorsque deux attributs en particulier varient ensemble, l'explication doit porter sur au moins un de ces attributs.

Enfin, la diversité peut également être utilisée afin d'aider l'utilisateur à choisir entre les différentes recommandations qui lui sont proposées. Outre les explications qui peuvent accompagner une liste de recommandations, il est important d'aider l'utilisateur à faire un choix entre différentes alternatives sans qu'il ait besoin de consulter chacun des items pour savoir si ces derniers

lui correspondent. Pour ce faire, il est possible de fournir de courtes descriptions pour chacun des items proposés. Ces dernières peuvent être affichées par défaut ou à la demande de l'utilisateur (via un bouton ou au survol avec le curseur de la souris par exemple) afin de ne pas surcharger l'interface. Cependant, au lieu d'expliquer en quoi les items correspondent aux caractéristiques des items précédents (utilisation de la similarité), nous soutenons l'hypothèse selon laquelle le fait de mettre en avant les différences de chacun des items permet à l'utilisateur de choisir plus facilement un item. La Figure 6.6 illustre cette hypothèse et représente la différence entre des explications basées sur la similarité et la diversité.

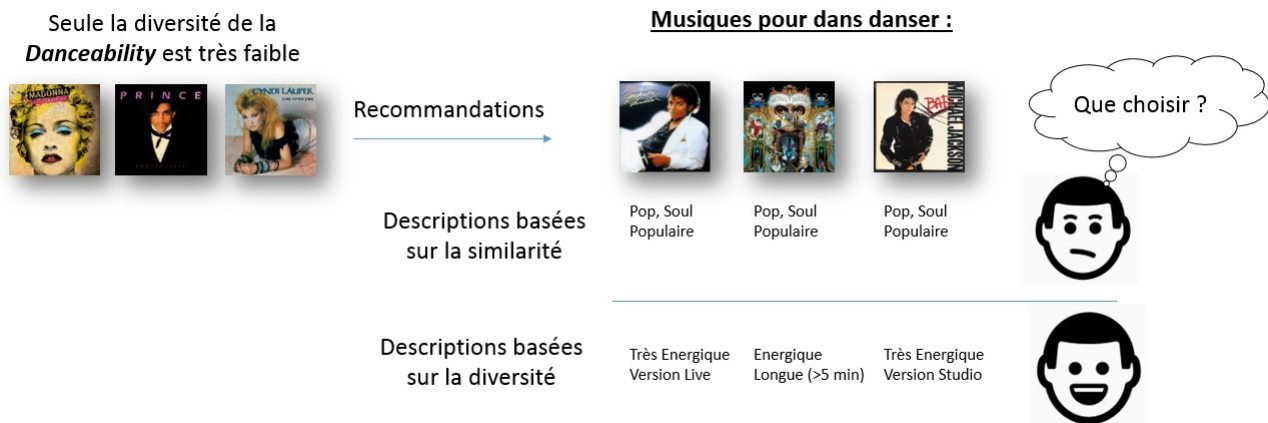


FIGURE 6.6 – Explications des recommandations sur le site Amazon

Cet exemple est proche du deuxième cas de la figure précédente à la différence que l'ensemble des musiques proposées appartient au même artiste. Nous avons fait ce choix d'une part, car cela permet d'illustrer davantage nos propos, mais également, car ce cas peut se produire si de nombreuses musiques de cet artiste sont présentes dans des contextes similaires ou que l'analyse des usages de la communauté révèle que les musiques de cet artiste s'avèrent être les meilleures recommandations. Avec cet exemple, il est évident que les explications basées sur les similarités ne permettront pas à l'utilisateur de choisir entre les différents items si ce dernier ne les connaît pas. Ne possédant aucun critère autre que l'image (la photo du produit, la pochette de l'album dans le cas de la musique, l'image de l'affiche du film...) pour distinguer les items entre eux, il devra s'en remettre au hasard pour sélectionner une recommandation. Inversement, en mettant en avant les différences existantes entre les items, l'utilisateur peut prendre une décision en fonction des caractéristiques présentées. De plus, comme nous le verrons dans la section suivante, la présentation des différences entre les items permet au travers du choix des utilisateurs d'obtenir des informations supplémentaires quant à ses besoins.

6.2.2 Les actions des utilisateurs

Tout comme les explications accompagnant les recommandations sont importantes pour les utilisateurs, celles accompagnant les actions des utilisateurs sont importantes pour le système. Dans l'état de l'art, nous avons montré que les traces explicites possèdent l'inconvénient de nécessiter une action de la part des utilisateurs [Jawaheer et al., 2014] et qu'il est, dans la mesure du possible, préférable d'exploiter les traces implicites des utilisateurs. Cependant, malgré cet inconvénient, nous soutenons que les systèmes actuels et plus particulièrement les approches basées sur le contexte implicite ne peuvent se passer de ces traces explicites, permettant par ailleurs de donner à l'utilisateur un contrôle sur ses préférences [Cranor, 2005]. Cela est d'autant

plus vrai dans les cas où un utilisateur ne souhaite pas consulter un item pour lequel il a cependant affirmé sa préférence. Dans le domaine de la musique par exemple, si un utilisateur ne souhaite pas écouter une musique qui lui est proposée en mode radio, il peut soit le préciser au travers d'un système de notation (en attribuant une note faible ou utiliser un système de pouce vers le bas), soit passer à la musique suivante. Outre le fait de savoir que cet item n'est pas apprécié à cet instant, la/les raison(s) pour laquelle cet item n'est pas apprécié reste inconnue. Or, cette information est d'autant plus importante qu'elle conditionne le comportement que doit alors adopter le SR. Dans le cas de musique, lorsqu'un item est proposé et qu'un utilisateur passe à la musique suivante ou ne choisit pas les recommandations, un utilisateur peut ne pas apprécier :

- la musique ;
- l'artiste ;
- que cette musique lui soit de nouveau proposée ;
- que cette musique lui soit proposée à ce moment précis.

Un SR ne peut et ne doit pas prendre le risque de ne plus recommander une musique qui a été rejetée par l'utilisateur sans être certain que ce dernier ne l'apprécie pas ou alors pas dans ce contexte. Un utilisateur peut par exemple souhaiter découvrir de nouvelles musiques et alors refuser toutes les recommandations de musiques qu'il connaît déjà. Or, si le système ne possède pas cette information, et qu'une musique est rejetée, il peut juger que cette dernière était inadaptée à ce contexte. Ce comportement du système n'est alors pas adapté. Le système devrait par exemple attendre un certain délai avant de recommander de nouveau cet item et non pas associer une préférence négative par rapport à ce contexte. De plus, en respectant la vie privée de l'utilisateur, nous diminuons la quantité d'informations à partir desquelles nous pouvons expliquer le comportement de l'utilisateur. Les SR exploitant des facteurs contextuels possèderaient l'information selon laquelle cette musique n'est pas souhaitée dans ce contexte explicite. Nos approches basées sur le contexte implicite ne pourront pas établir ce lien. Pour cette raison, nous soutenons que notre approche de recommandation doit être accompagnée d'un système de *feedback* adéquat et qu'il est préférable de solliciter à bon escient l'utilisateur plutôt que de collecter des informations sur sa vie privée. La Figure 6.7 ci-dessous représente un schéma des *feedbacks* qu'il serait souhaitable de récupérer chez l'utilisateur dans le domaine de la musique. Les formes en traits pleins représentent les éléments d'interface qui sont toujours visibles et les traits en pointillés représentent les éléments complémentaires qui apparaissent une fois qu'un élément est sélectionné.

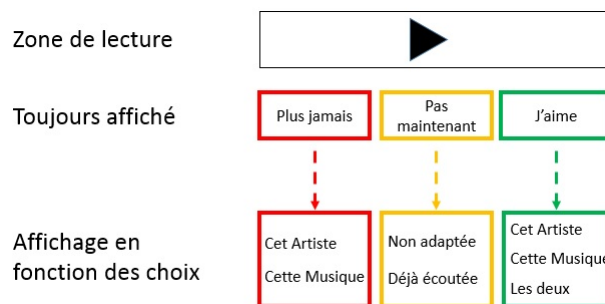


FIGURE 6.7 – Schéma des explications des choix pouvant être données par un utilisateur

Dans ce manuscrit, nous avons utilisé la métrique de diversité au niveau des consultations pour connaître son évolution au cours du temps, ainsi qu'au niveau d'une liste de recommandations afin de générer des explications pour les recommandations. Outre ces deux cas, la diversité

peut également être utilisée lorsqu’une recommandation est choisie dans une liste de recommandations afin de comparer ses caractéristiques par rapport aux autres items (non choisis). Comme nous l’avons montré sur la Figure 6.6, notre modèle permet de mettre en avant les caractéristiques particulières/diverses des items proposés. De ce fait, lorsqu’un utilisateur sélectionne un item parmi un ensemble d’items, il sélectionne les caractéristiques qui l’importe et rejette donc par la même occasion ceux qu’il ne désire pas/moins. Ainsi, lorsqu’un utilisateur sélectionne une recommandation dans une liste de recommandations, l’information apportée par son choix est double puisqu’elle renseigne : ce qu’il a choisi et ce qu’il n’a pas choisi. Si un système ne se base que sur les choix de l’utilisateur pour calculer ses recommandations, il cherchera à répondre aux besoins des utilisateurs tout en incluant de la diversité dans les items recommandés afin de pouvoir lui offrir le plus d’alternatives possibles. Cependant, un système qui exploite en plus les items non sélectionnés possède des informations supplémentaires : ce que ne souhaite pas l’utilisateur. Par exemple, si un utilisateur ne sélectionne jamais une musique qui est un “live” (*Liveness* élevée) lorsque cette caractéristique est explicitement présentée (cf. Figure 6.6) à plusieurs reprises dans des listes de recommandations qui mettent en avant ce critère, la probabilité que cette caractéristique intéresse l’utilisateur est très faible. Dès lors, il est possible d’enrichir le modèle de préférences courant de l’utilisateur en fonction des caractéristiques qui sont présentées à l’utilisateur, de celles qui sont choisies, mais également de celles qui ne sont pas choisies à plusieurs reprises. Outre, les préférences des attributs, ce choix de l’utilisateur nous informe de ces préférences si les items partagent des caractéristiques similaires. En effet, lorsque l’utilisateur choisit un item parmi plusieurs, il sélectionnera celui qui lui plait le plus même tous les items font partie de ces préférences.

6.3 Discussions

Si nous reprenons la liste des dimensions et des facteurs humains que doivent couvrir les SR établie par [Shani and Gunawardana, 2011] que nous avons présentée dans l’état de l’art 1.1.4, nous pouvons constater que l’exploitation des contextes implicites et de l’évolution de la diversité permettent de couvrir nombre d’entre elles :

- **Couverture** : si l’approche de recommandation collaborative ou encore l’approche visant à exploiter les contextes similaires en termes de caractéristiques sont utilisées seules, certains problèmes de couverture seront inéluctables. Du fait que ces approches ne permettent de retrouver que les items contenus dans les contextes implicites existants, une couverture maximale ne peut pas être garantie. Cependant, il est possible d’augmenter la couverture en recommandant non pas directement les items retrouvés, mais des items qui leur sont similaires (en termes d’attributs).
- **Confiance système** : les recommandations sont obtenues sur la base de la diversité et de l’évolution des attributs des consultations récentes et sont accompagnées d’explications. Nous nous assurons donc que l’utilisateur comprenne pourquoi les recommandations lui sont proposées.
- **Confiance utilisateur** : de par la manière dont sont calculées les recommandations et le fait que des explications puissent être facilement fournies aux utilisateurs, notre approche permet d’apporter la transparence nécessaire afin de donner confiance à l’utilisateur envers le SR.
- **Nouveauté** : l’ajout d’une nouvelle musique ne posera pas de problème du moment que les attributs de cette musique sont connus. Les nouveaux items doivent être consultés

plusieurs fois afin d'appartenir à des contextes implicites. Cependant, ces items peuvent être recommandés en utilisant des calculs de similarité. De la même manière, l'arrivée d'un nouvel utilisateur ne nécessitera que quelques interactions avec le système afin de constituer un profil utilisateur minimal sur lequel calculer les recommandations. De plus, notre modèle se basant essentiellement sur les consultations courantes de l'utilisateur, des recommandations pourront être proposées dès les premières consultations.

- **Sérendipité** : l'aspect collaboratif de la fouille de contextes implicites s'avère être un moyen efficace de proposer des items qui n'auraient pas été spontanément consultés par l'utilisateur que l'on cherche à recommander. Cependant, notre approche étant basée sur les contextes implicites dans lesquels les items partagent des caractéristiques communes, cet aspect reste modéré. Une étude utilisateur devrait permettre de connaître le potentiel que possèdent nos approches de recommandation vis-à-vis de cette dimension.
- **Diversité** : notre approche permet à chaque pas de temps de s'adapter au niveau de diversité de l'utilisateur et sa prise en compte constitue le cœur de nos approches.
- **Utilité** : les recommandations proposées tiennent à la fois compte du CI courant de l'utilisateur et du niveau de diversité ce qui nous permet de nous adapter au besoin des utilisateurs à court terme. De plus, une modélisation du niveau de diversité et le suivi du contexte de l'utilisateur réalisé à chaque nouvelle consultation rendent notre modèle très réactif aux changements de comportements des utilisateurs.
- **Robustesse** : nous avons pu tester notre approche de modélisation face aux données manquantes. Cependant des expériences supplémentaires doivent être conduites pour en connaître les conséquences en termes de recommandations. Certaines expériences doivent également être conduites afin de tester l'impact de données frauduleuses (profils utilisateurs aberrants par exemple).
- **Respect de la vie privée** : à aucun moment nos modèles n'exploitent d'informations personnelles. En effet, du point de vue de la modélisation du comportement des utilisateurs comme de la phase de calcul des recommandations, seules les caractéristiques des items sont exploitées. Les seules informations que nous possédons sur les utilisateurs sont les historiques de consultation ainsi que les préférences des utilisateurs qui sont des données dont la protection peut être assurée via des méthodes d'anonymisation. Notre approche ne souffre donc pas de problème lié au respect de la vie privée.
- **Capacité d'adaptation** : l'utilisation des contextes implicites est une manière de cerner et de définir les besoins des utilisateurs. La réactivité de notre approche permet à chaque nouvelle consultation de connaître les besoins de l'utilisateur courant. De plus, étant donné que nous travaillons à l'échelle des attributs des items, nous pouvons facilement nous adapter aux besoins spécifiques des utilisateurs. Par exemple, si nous constatons que certains attributs sont systématiquement évités ou inversement que d'autres sont toujours présents, nous pouvons sélectionner les items correspondant à ces critères.
- **Passage à l'échelle** : nous avons montré que la complexité de notre modèle en termes de modélisation de la diversité et de détection de CI était faible et constante. La phase de recommandation peut être plus sujette à des contraintes de temps de calcul étant donné qu'elle peut nécessiter de parcourir de nombreux contextes implicites. Cependant, cet aspect nécessite d'être développé davantage afin de savoir quelles sont les dispositions nécessaires à prendre pour fournir des recommandations en temps réel. Par exemple, il est possible d'utiliser des techniques de *clustering* de manière hors-ligne afin de connaître les CI les plus similaires au CI pour lequel des recommandations doivent être déterminées. De plus, certaines propositions possèdent l'avantage d'être moins coûteuses que d'autres. Par exemple,

si les contextes implicites sont caractérisés à partir d'un ensemble de caractérisations possibles définies au préalable (caractérisation 1 : haute "énergie" et haute "instrumentalness"; caractérisation 2 : faible "énergie" et haute "loudness"; caractérisation 3 : forte "danceability" et "énergie" croissante...) alors il n'est plus question de parcourir les CI un par un mais de récupérer les CI possédant cette caractéristique.

Dans le Tableau 6.6, nous présentons un résumé des forces et des faiblesses de chaque approche de modélisation et de recommandation que nous avons mentionnées dans ce manuscrit.

Approche	Forces	Faiblesses
Collaborative à partir des id	<ul style="list-style-type: none"> – Sérendipité – Items déjà consultés dans des CI antérieurs 	<ul style="list-style-type: none"> – Temps de recherche – <i>Cold start</i> sur les items – Dépendant de l'historique
Collaborative à partir des attributs	<ul style="list-style-type: none"> – Augmente l'espace de recherche – Items déjà consultés dans des CI antérieurs 	<ul style="list-style-type: none"> – Nécessite d'isoler les attributs pertinents – Dépendant de l'historique
Contenu du CI	<ul style="list-style-type: none"> – Augmente l'espace de recherche – Non dépendant de l'historique 	<ul style="list-style-type: none"> – Nécessite d'isoler les attributs pertinents
Adaptation à l'évolution en diversité	<ul style="list-style-type: none"> – Réactivité aux changements – Compréhension des besoins 	<ul style="list-style-type: none"> – Nécessite d'isoler les attributs pertinents
Prédire les actions des utilisateurs	<ul style="list-style-type: none"> – Anticipation des besoins 	<ul style="list-style-type: none"> – Requiert une phase apprentissage – Risque d'orienter l'utilisateur – Nécessite une caractérisation du CI pertinente
Explication des recommandations	<ul style="list-style-type: none"> – Aide à la décision – Permet de gagner la confiance des utilisateurs 	<ul style="list-style-type: none"> – Nécessite d'isoler les attributs pertinents

TABLE 6.6 – Avantages et inconvénients des approches proposées

Nous avons atteint notre **troisième** objectif : proposer différentes approches de recommandation basées sur les contextes implicites. Les approches proposées visent à répondre à plusieurs besoins des utilisateurs : leurs besoins immédiats ainsi que leurs futurs besoins. Nous avons également montré de quelle manière les contextes implicites et notamment l'évolution de la diversité pouvaient être mis à contribution dans le but des explications vis-à-vis des recommandations.

Objectif suivant : Évaluer les performances de nos approches de recommandation en situations réelles.

Chapitre 7

Conclusions et perspectives

7.1 Conclusions

L'importance des SR n'est plus à démontrer. Qu'ils soient utilisés ou non de pair avec d'autres outils numériques afin d'assister les utilisateurs face à des volumes de données chaque jour plus importants, ils ont été, sont et resteront des outils indispensables. Nous avons vu au travers de l'état de l'art que les SR ont connu deux grandes phases dans leur évolution. La première phase a consisté en la mise au point et l'amélioration d'approches permettant de prédire les préférences des utilisateurs sur lesquelles étaient basées les recommandations. L'objectif étant à ce moment-là de répondre à la question "Comment prédire le plus justement possible les préférences d'un utilisateur?". Durant cette période, la précision dans l'estimation des préférences des utilisateurs était utilisée afin de mesurer la performance d'un SR. La fin de cette première phase et le début de la suivante sont marquées par l'apparition de travaux démontrant que la place de la précision des prédictions des préférences est toute relative dans l'objectif principal des SR : assister les utilisateurs. Cette seconde phase est guidée par le besoin de replacer l'utilisateur, en tant qu'humain possédant des processus de décision complexes, au cœur du processus de recommandation. L'objectif devient alors d'apporter des réponses à la question : "Quels sont les facteurs humains qui doivent être pris en compte et comment les inclure dans le processus de recommandation?". Malgré les nombreux progrès réalisés suite à cette prise de conscience, la fin ne justifiant pas les moyens, certaines approches de recommandation doivent cependant être remises en question. Par exemple, même si la qualité des recommandations est effectivement améliorée par la prise compte du contexte des utilisateurs, les approches employées reposent essentiellement sur l'exploitation des données personnelles. Or, à l'heure où la protection et le respect de la vie privée sont au cœur des préoccupations⁸⁴, ces approches ne peuvent qu'être discutables. Indépendamment de ce problème de respect de la vie privée, certaines études démontrent que la prise en compte des facteurs humains tels que la diversité ne peut pas se résumer à un score à maximiser et qu'il faut aller en amont, en étudiant le comportement humain, pour parfaire les approches de recommandation. En effet, de nombreux travaux se sont focalisés sur la diversité, comme cela fut le cas pour la précision, en tant que scores à maximiser afin de valider la performance d'une approche. Or, la diversité est une dimension complexe dont l'utilisation peut être néfaste du point de vue de la satisfaction des utilisateurs si elle n'est pas correctement maîtrisée. En partant des conclusions de certains travaux établissant un lien entre le contexte et la diversité, nous nous sommes alors interrogés sur le potentiel de cette dimension, notamment en ce qui concerne son évolution au cours du temps.

84. <https://www.cnil.fr/fr/textes-officiels-europeens-protection-donnees>

Partant de ces constats et n'ayant pas trouvé dans la littérature de travaux s'intéressant à ce sujet, nous avons développé le **premier modèle de l'état de l'art permettant de modéliser la diversité au cours du temps**. Ce modèle permet de connaître pour chaque nouvelle consultation d'un utilisateur l'apport en diversité (calculé sur les caractéristiques des items) par rapport aux consultations précédentes et donc, de savoir comment évolue la diversité au cours du temps. En fonction de cette évolution, il est possible de déterminer si un utilisateur spécialise, diversifie ou stabilise ses consultations, et ce, pour l'ensemble des attributs disponibles. Cette approche de modélisation peut être paramétrée à de nombreux niveaux et sa complexité faible et constante lui permet d'être déployée en temps réel. Étant donné que seules les formules de diversité utilisées sont dépendantes de la nature des items traités, notre modèle peut être intégré dans de nombreux domaines applicatifs. Parmi les domaines applicatifs potentiels, nous pouvons notamment citer la musique en ligne, domaine sur lequel nous avons fait nos études, l'e-éducation, le e-commerce ou les réseaux sociaux professionnels. . . la seule contrainte à respecter étant que les items consultés forment des séquences d'items contigus pour lesquels les temps de consultation sont relativement faibles. Connaître la propension d'un utilisateur à diversifier ses consultations constitue une information précieuse à elle seule quant aux caractéristiques des items devant être recommandés. Par ailleurs, en nous basant sur des travaux mentionnant l'existence d'une relation entre le contexte d'un utilisateur et son besoin en diversité, nous avons exploité notre modèle pour étudier ce lien.

Le modèle DANCE que nous avons développé et formalisé dans ce manuscrit s'intéresse à l'évolution de la diversité au cours du temps afin d'isoler certaines séquences de consultations répondant à des caractéristiques particulières que nous nommons **contextes implicites**. Un contexte implicite correspond au laps de temps pendant lequel des consultations contiguës partagent des caractéristiques communes, ce qui se traduit par un faible niveau de diversité. Afin de détecter des changements de contextes implicites, nous nous intéressons aux moments où la diversité augmente de façon remarquable. De la même manière qu'il est possible de paramétrer l'approche de modélisation de la diversité, les conditions permettant de détecter un changement de contexte implicite sont également paramétrables. Étant donné la correspondance entre les changements de contextes explicites et implicites que nous avons révélée, nous avons pu conclure que les contextes implicites pouvaient servir d'alternatives aux contextes explicites. De plus, nous avons pu voir au travers d'une étude utilisateur que les attributs que nous exploitons sont effectivement liés aux caractéristiques des contextes explicites. Ainsi, plutôt que de déterminer les causes (les contextes explicites) des choix des utilisateurs, nous nous intéressons aux choix eux-mêmes afin d'adapter les recommandations aux besoins courants des utilisateurs. Outre le fait de ne pas se reposer sur la collecte et l'exploitation d'informations sensibles, les contextes implicites possèdent l'avantage d'être génériques et de pouvoir être définis dans n'importe quel domaine contrairement aux contextes explicites qui peuvent être dépendants du domaine applicatif.

En ce qui concerne la phase de recommandation, plusieurs utilisations possibles de l'évolution temporelle de la diversité et des contextes implicites ont également été présentées dans ce manuscrit. Nous avons par exemple proposé deux approches permettant d'exploiter l'ensemble des contextes implicites d'une communauté d'utilisateurs afin de calculer les recommandations. L'évolution de la diversité offre pour sa part plusieurs avantages, soit dans le calcul des recommandations, soit dans le réordonnement d'une liste de recommandations générée par d'autres approches. La diversité peut également s'avérer efficace afin de produire des explications sur l'utilité des recommandations. Enfin, une connaissance et une maîtrise du niveau de diversité de la liste de recommandations permet non seulement d'anticiper certains besoins des utilisateurs, mais également d'améliorer la compréhension de leur comportement.

Pour conclure ce manuscrit, nous rappelons que les systèmes de recommandation, tout comme l'ensemble des outils numériques déployés à l'heure actuelle, ne doivent pas oublier que leur finalité est avant tout d'assister les utilisateurs. Les facteurs humains, les processus de décision et le caractère parfois imprévisible de l'humain sont autant de difficultés qu'il nous faut appréhender et comprendre afin de fournir de meilleures approches de recommandation. Cependant, cela ne peut se faire au détriment du respect de la vie privée des utilisateurs qui doivent également pouvoir rester maîtres de l'information qui leur est proposée.

7.2 Perspectives

Le travail réalisé dans le cadre de cette thèse ayant abouti à notre connaissance à la proposition du premier modèle de l'état de l'art permettant de modéliser l'évolution de la diversité au cours du temps, ainsi que la proposition d'une nouvelle famille de contextes, les perspectives offertes sont nombreuses. Nous avons décidé de les classer en deux catégories, celles étant réalisables à court terme et celles nécessitant une planification à moyen terme.

7.2.1 Perspectives à court terme

Les approches de détection de changement de contexte implicite que nous avons proposées ont essentiellement servi à tester nos différentes hypothèses. Nous souhaitons améliorer ces approches et ce, à différents niveaux. Premièrement, nous projetons de confronter les contextes implicites que nous détectons au jugement des utilisateurs. Cette approche nous permettra de confirmer que les contextes implicites font sens pour les utilisateurs, mais également d'obtenir des retours d'expérience nous permettant de mettre au point d'autres méthodes de détection. Deuxièmement, il est possible d'adapter les fonctions de détection au profil de chaque utilisateur. Un utilisateur écoutant majoritairement des musiques de genre rock sera sans doute sensible aux variations de diversité de certains attributs qui ne seront pas perçues ou jugés pertinents pour un utilisateur ayant des goûts plus éclectiques. Troisièmement, la gestion des données manquantes peut être améliorée par l'utilisation de méthodes probabilistes visant à prédire les valeurs des attributs les plus probables étant donnés les attributs connus, les préférences de l'utilisateur ainsi que le contexte implicite courant.

Par ailleurs, nous avons présenté dans le dernier chapitre de nombreuses approches de recommandation et d'exploitation des contextes implicites. Nous entendons donc mettre en place des tests utilisateurs afin d'évaluer ces différentes approches de recommandation en nous comparant aux approches de l'état de l'art. Bien que certains résultats puissent être obtenus à partir des corpus hors ligne, nous soutenons l'idée selon laquelle les bénéfices de notre modèle et des différentes approches de recommandation ne pourront être maximisés qu'en situation réelle d'utilisation. L'utilisation d'une étude utilisateur nous permettra également d'interagir directement avec les utilisateurs et de collecter des explications concernant les variations de diversité que nous pouvons obtenir en temps réel. Ces interactions pourraient par exemple permettre de tester l'hypothèse selon laquelle il existe des périodes de transitions entre deux contextes implicites qui se traduisent par une diversité élevée sur plusieurs consultations. Ou encore, de la même manière que nous nous sommes intéressés aux musiques passées, une perspective d'utilisation de l'évolution de la diversité au cours du temps serait la détection de sessions de navigation parallèle ou encore la détection de séquence de consultations contenant plusieurs objectifs. Sur un site de e-commerce par exemple, un utilisateur peut consulter des items pour faire un cadeau à la fois pour son conjoint et pour ses amis. Bien qu'il existe deux objectifs différents et donc

deux besoins différents, l'utilisateur peut consulter à la suite des items pour l'un ou l'autre de ces cadeaux, générant ainsi beaucoup de diversité. Utiliser la diversité pour isoler chacun des objectifs (en permettant les consultations afin de voir s'il n'existe pas plusieurs contextes implicites entrecroisés) permettrait d'assister plus efficacement l'utilisateur plutôt que de considérer l'ensemble des consultations comme appartenant à un seul et même objectif.

Enfin, comme nous l'avons mentionné, notre approche est générique et peut être déployée dans de nombreux domaines applicatifs. Cette perspective pourra être étudiée dès lors que nous posséderons d'autres jeux de données respectant les contraintes imposées par notre modèle, possédant ou non des items de natures différentes.

7.2.2 Perspectives à moyen terme

La caractérisation des contextes implicites est un point essentiel sur lequel reposent plusieurs approches de recommandation et d'explication que nous avons proposées. Cette caractérisation peut être réalisée comme nous l'avons présenté en déterminant certaines valeurs remarquables des attributs (valeurs moyennes ou extrêmes, très fortes stabilités de certains attributs...) dans un premier temps. Cependant, améliorer la caractérisation des contextes implicites pourrait permettre d'assister au mieux les utilisateurs tout en améliorant la compréhension que nous avons de leur comportement. Pour cette raison, nous envisageons d'étudier et de proposer différentes techniques de caractérisation. Et ce, afin de prendre en compte la dynamique de certains attributs des items contenus dans un contexte implicite, d'étudier les relations pouvant exister entre ces attributs ou encore de déterminer les attributs les plus importants dans chaque contexte implicite.

Jusqu'à présent, nous avons restreint l'exploitation de notre modèle aux consultations courantes des utilisateurs afin de connaître les besoins immédiats en diversité. Exploiter l'évolution temporelle de la diversité à une échelle de temps plus importante constitue cependant une perspective intéressante. En d'autres termes, en plus d'avoir des besoins en diversité plus ou moins importants sur certains attributs en fonction de leur contexte courant, nous pensons que les utilisateurs possèdent un besoin en diversité qui peut couvrir des périodes plus grandes qu'un contexte. Dans le cas de la musique, un utilisateur peut avoir des besoins en diversité très importants pour certains attributs pendant une période et avoir par la suite des besoins plus restreints en termes de diversité. Ce point peut être illustré par la saison de l'année qui possède une influence sur les caractéristiques des musiques écoutées. L'idée n'est pas de s'appuyer sur la saison pour adapter les recommandations mais de détecter certaines périodes (telles que les saisons) pendant lesquelles certaines caractéristiques sont moins ou plus présentes que d'autres. Détecter ces différentes périodes permettrait de mieux comprendre les besoins des utilisateurs.

De plus, outre les attributs que nous avons présentés et utilisés dans ce manuscrit, l'API de Spotify nous permet également d'obtenir une analyse du spectre audio des musiques. L'analyse du spectre audio correspond à un découpage de la musique en segments où chaque segment représente un événement particulier tel la présence d'une percussion à la batterie ou d'une note d'un instrument. Une liste de caractéristiques est donnée pour chaque segment et il existe plusieurs milliers de segments par musique⁸⁵. L'utilisation de cette information peut être intéressante étant donné qu'elle se passe de toute subjectivité (contrairement aux attributs *valence* et *danceability* par exemple) et permettrait alors d'obtenir des informations "brutes" sur lesquelles la similarité/diversité d'une musique peut être calculée. Alors que l'ensemble des attributs d'une musique représente en moyenne 700 octets de données, le spectre audio représente 700 Ko, soit

85. <https://developer.spotify.com/console/get-audio-analysis-track/>

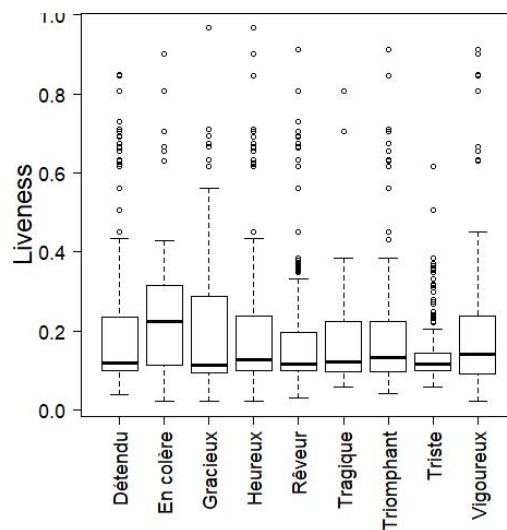
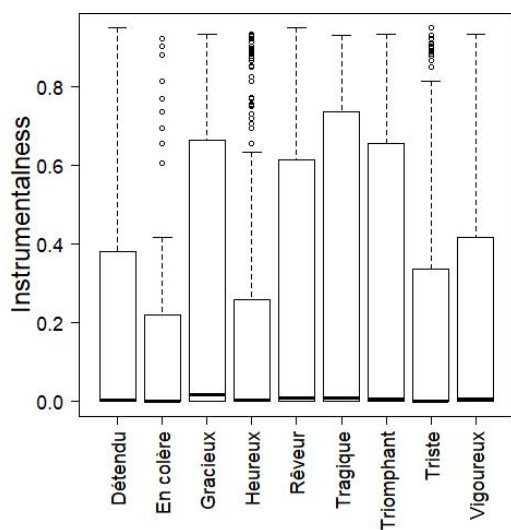
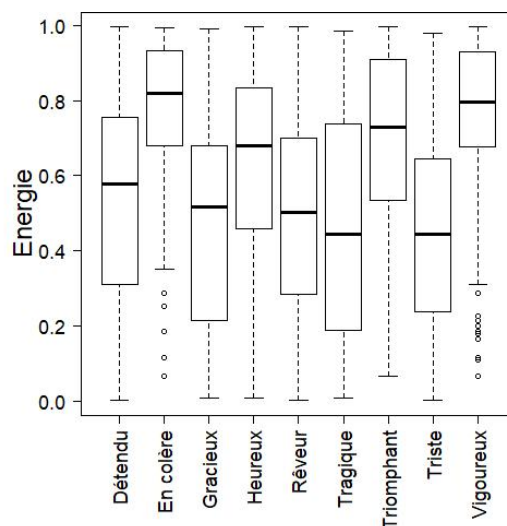
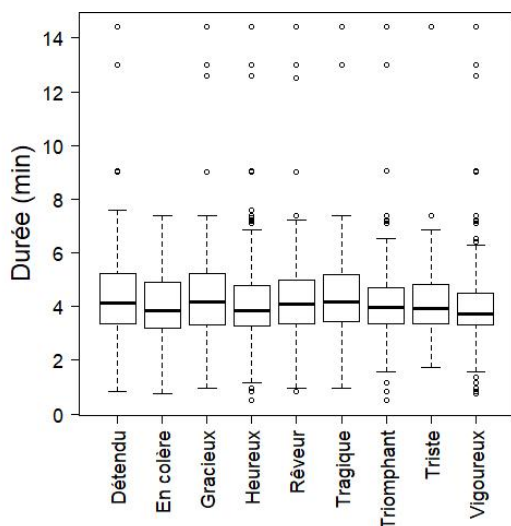
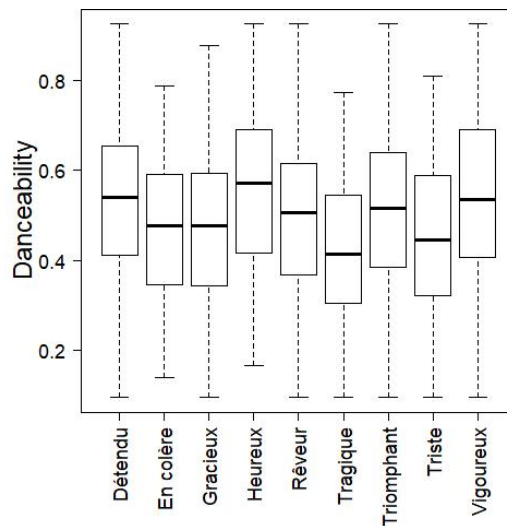
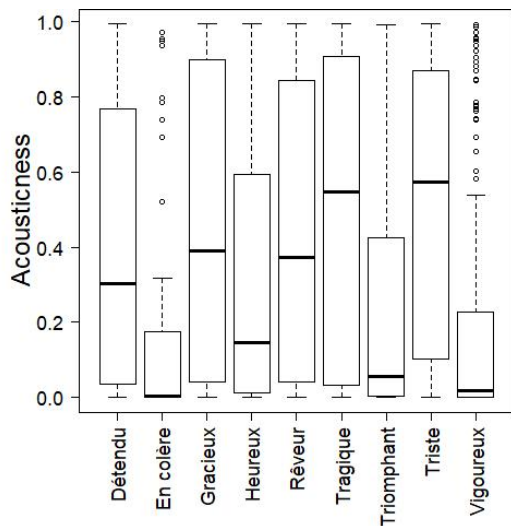
1 000 fois plus. Non ne savons pas actuellement si cette différence peut avoir un impact dans le calcul en temps réel de la diversité. De nouvelles formules de diversité devront également être développées afin d'exploiter ces informations. Le défi majeur de cette perspective réside dans la quantité d'information à traiter. En effet, exploiter des grandes quantités d'information aura pour conséquence d'ajouter du bruit qu'il faudra pouvoir isoler afin de ne conserver que l'information utile pour le SR et les utilisateurs. Cependant, nous sommes convaincus que l'analyse du spectre audio permet d'améliorer la compréhension des choix des utilisateurs étant donnée la complexité d'une musique (en termes de représentation).

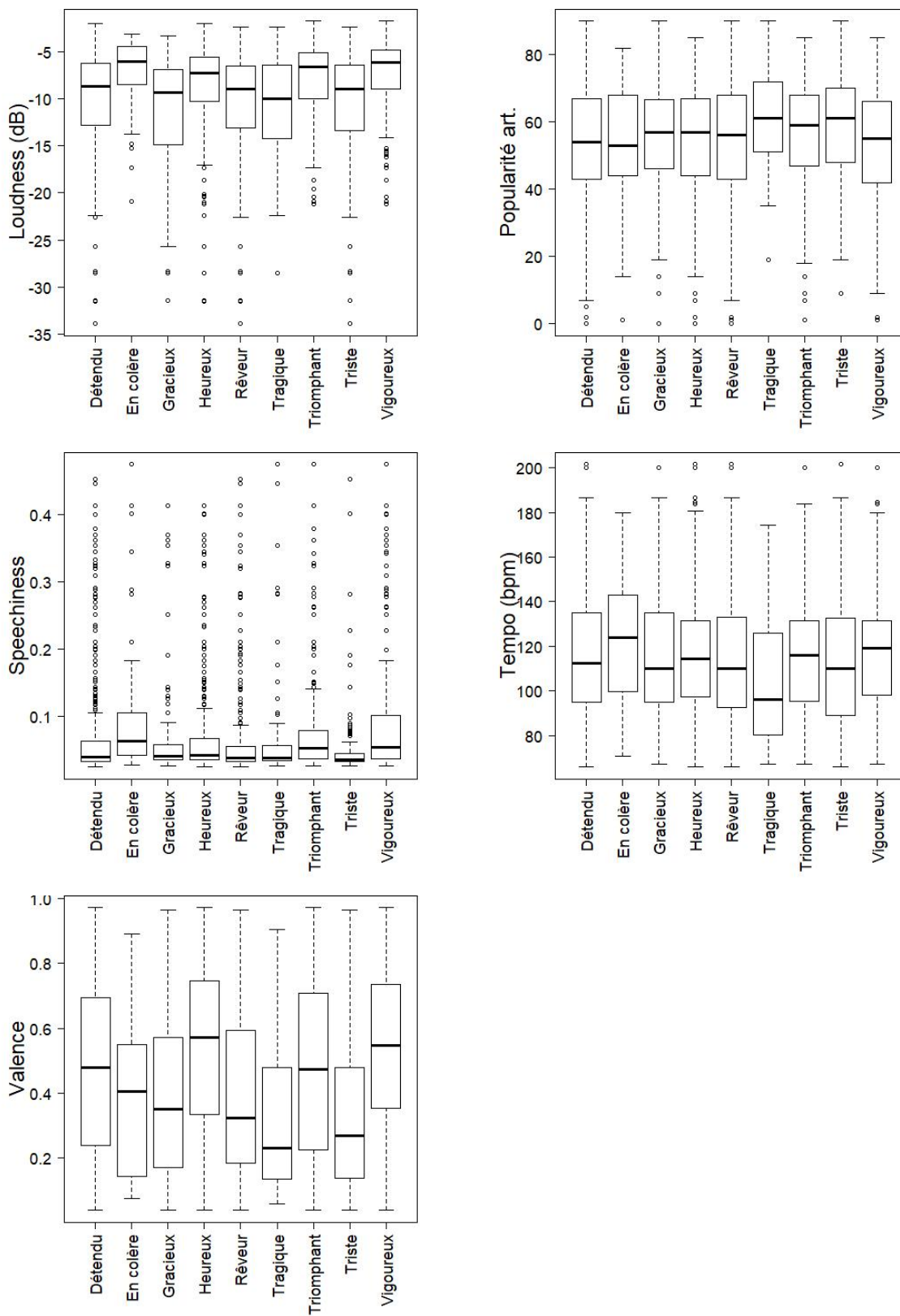
Annexe A

Boîtes à moustaches et tests de comparaisons multiples, chapitre 5

Émotions ressenties

Boîtes à moustaches





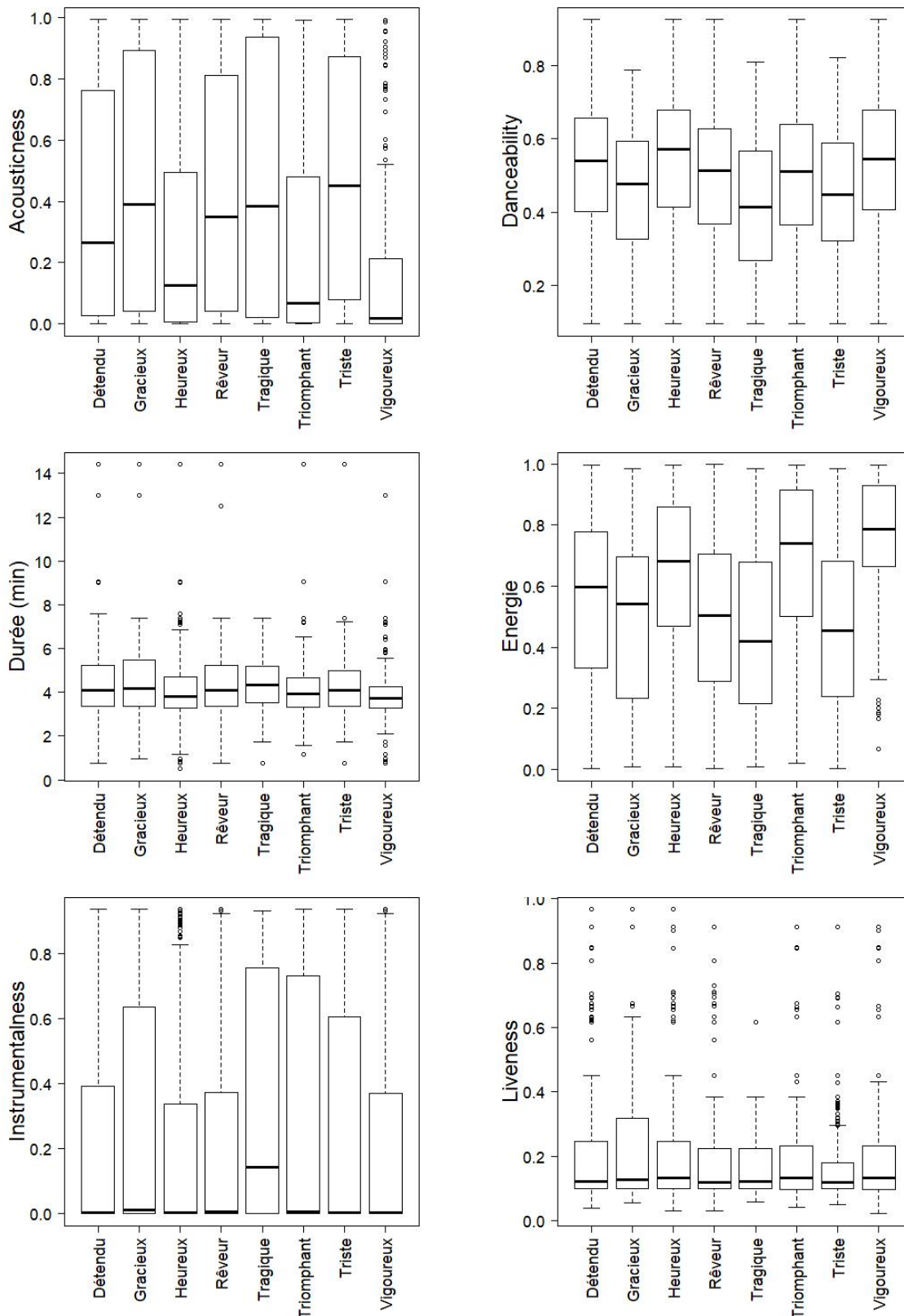
Comparaisons multiples du test de Kruskal et Wallis

TABLE A.1 – Comparaisons multiples du test de Kruskal et Wallis pour les émotions ressenties

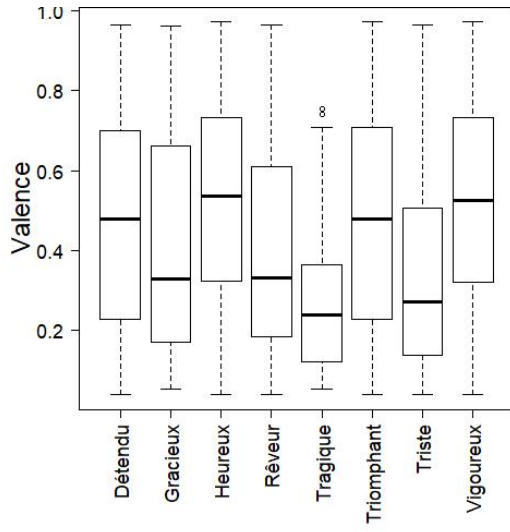
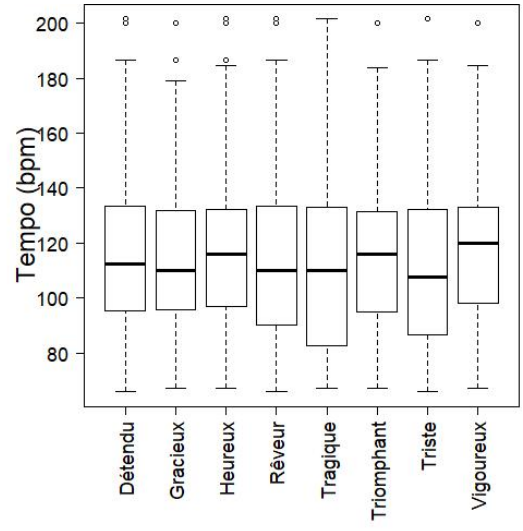
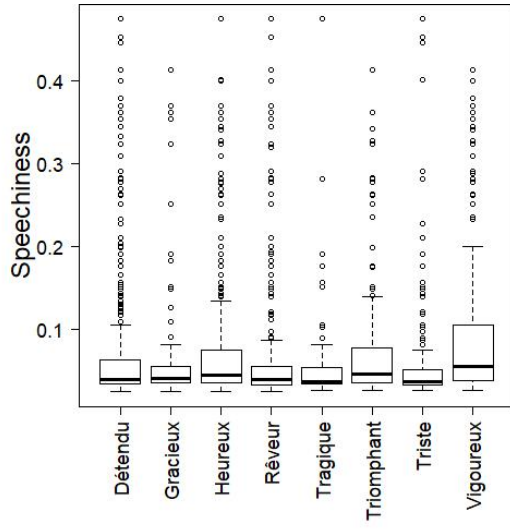
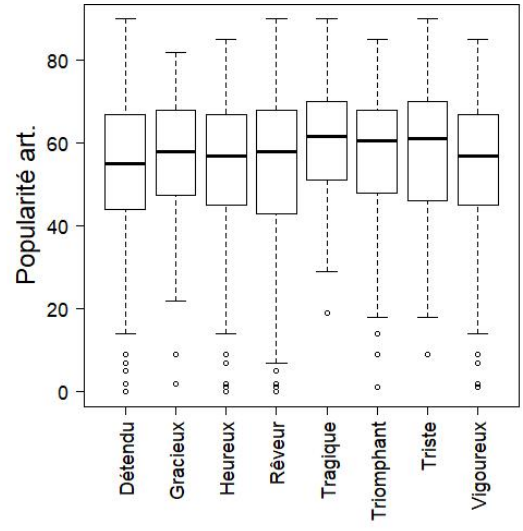
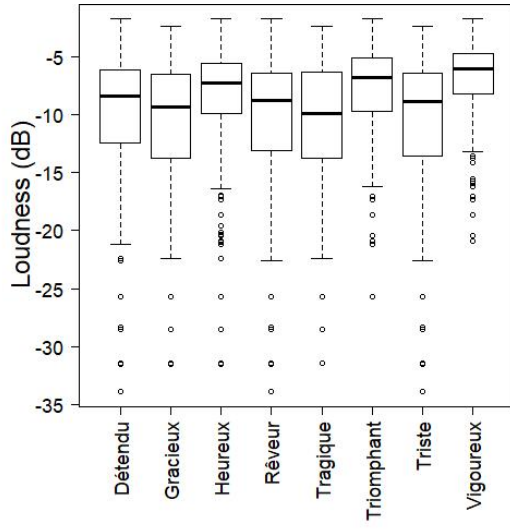
	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Détendu-En colère	vrai	faux	faux	vrai	faux	vrai	vrai	vrai	faux	faux	faux
Détendu-Gracieux	faux	vrai	faux	faux	faux	faux	faux	faux	faux	faux	faux
Détendu-Heureux	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Détendu-Rêveur	faux	vrai	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Détendu-Tragique	faux	vrai	faux	faux	faux	faux	faux	faux	vrai	vrai	vrai
Détendu-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Détendu-Triste	faux	vrai	faux	vrai	faux	faux	faux	vrai	faux	vrai	faux
Détendu-Vigoureux	vrai	faux	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
En colère-Gracieux	vrai	faux	faux	vrai	faux	vrai	vrai	vrai	faux	faux	faux
En colère-Heureux	vrai	vrai	faux	vrai	faux	faux	faux	vrai	faux	vrai	faux
En colère-Rêveur	vrai	faux	faux	vrai	faux	vrai	vrai	vrai	faux	faux	faux
En colère-Tragique	vrai	faux	faux	vrai	faux	faux	vrai	vrai	vrai	faux	faux
En colère-Triomphant	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
En colère-Triste	vrai	faux	faux	vrai	faux	vrai	vrai	vrai	faux	faux	faux
En colère-Vigoureux	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Gracieux-Heureux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux	faux	vrai	faux
Gracieux-Rêveur	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Gracieux-Tragique	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Gracieux-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Gracieux-Triste	faux	faux	faux	faux	faux	faux	faux	vrai	faux	faux	faux
Gracieux-Vigoureux	vrai	vrai	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Heureux-Rêveur	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Heureux-Tragique	vrai	vrai	faux	vrai	faux	faux	vrai	faux	vrai	vrai	faux
Heureux-Triomphant	vrai	vrai	faux	faux	faux	faux	faux	vrai	faux	vrai	faux
Heureux-Triste	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Heureux-Vigoureux	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Rêveur-Tragique	faux	vrai	faux	faux	faux	faux	faux	faux	faux	faux	faux
Rêveur-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Rêveur-Triste	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Rêveur-Vigoureux	vrai	vrai	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Tragique-Triomphant	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	vrai	vrai	faux
Tragique-Triste	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Tragique-Vigoureux	vrai	vrai	vrai	vrai	faux	faux	vrai	vrai	vrai	vrai	vrai
Triomphant-Triste	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Triomphant-Vigoureux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Triste-Vigoureux	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux

Émotions initiatrices ⁸⁶

Boîtes à moustaches



86. Cf. *supra* p.109, les émotions initiatrices sont les émotions dans lesquelles se trouvent les utilisateurs et qui les incitent rechercher une émotion en particulier. Par exemple, un utilisateur peut être triste et vouloir écouter une musique joyeuse.

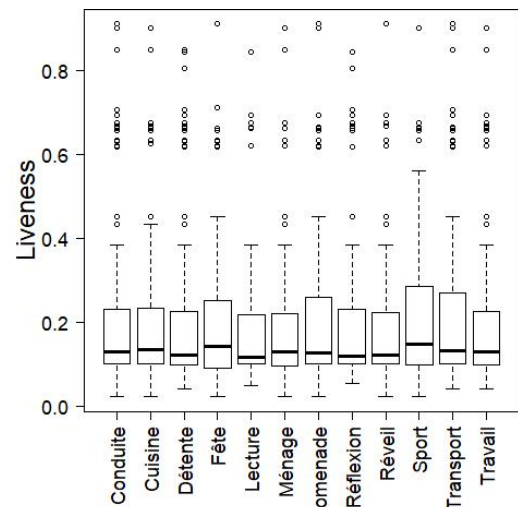
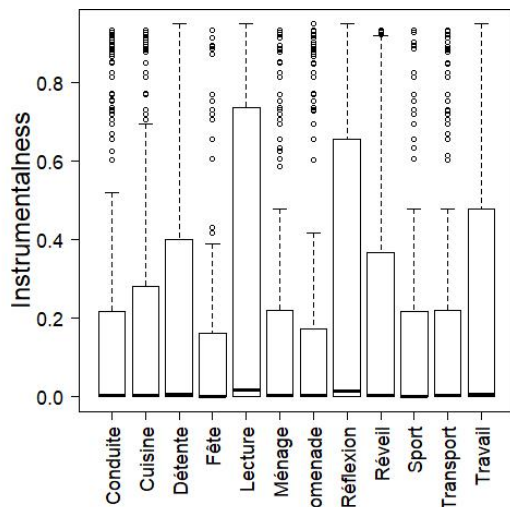
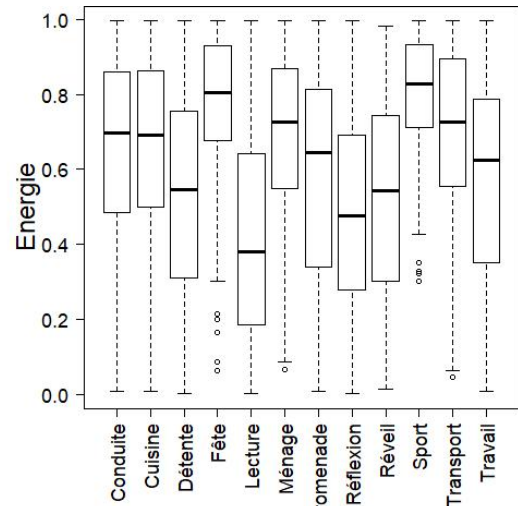
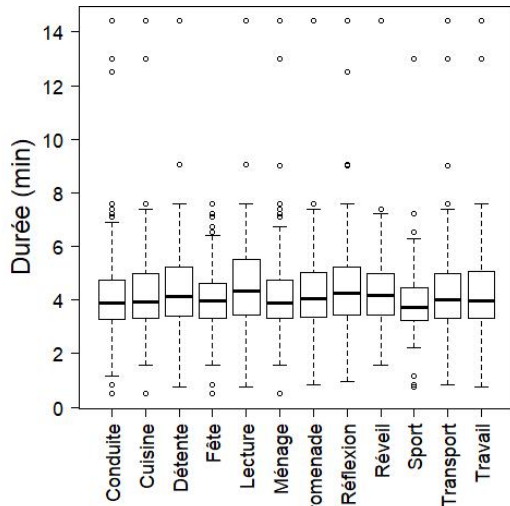
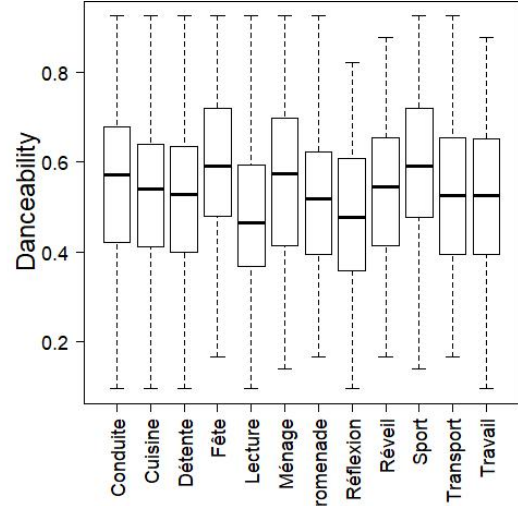
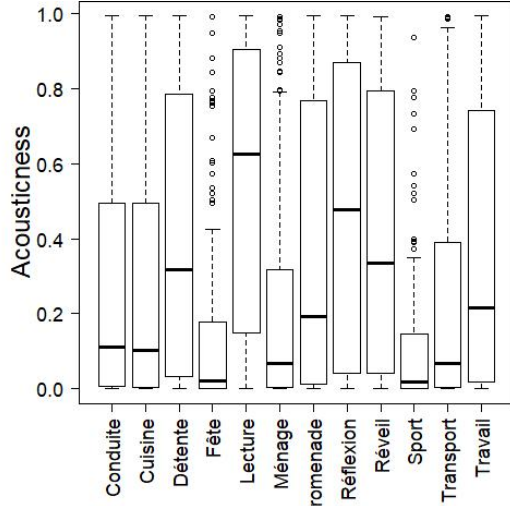


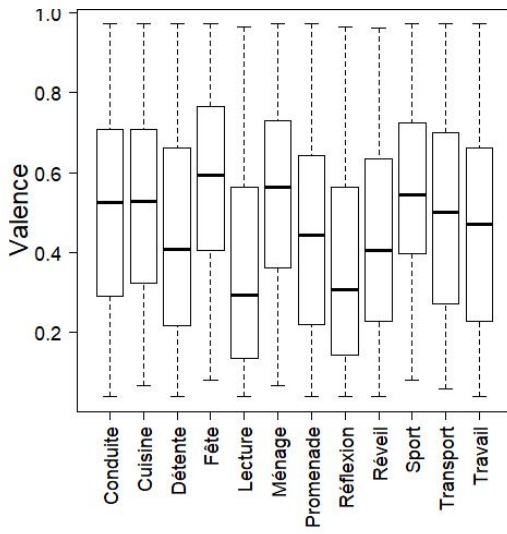
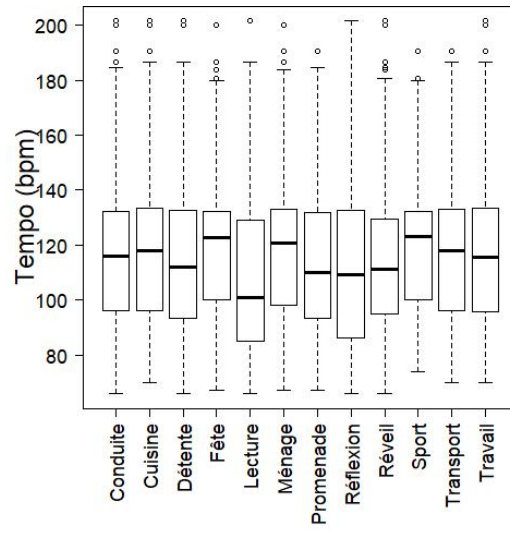
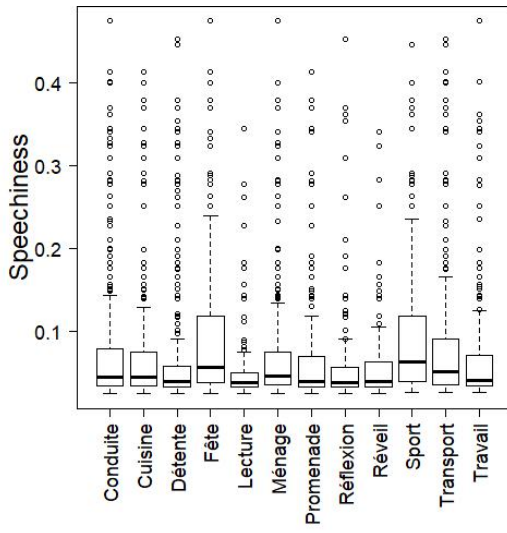
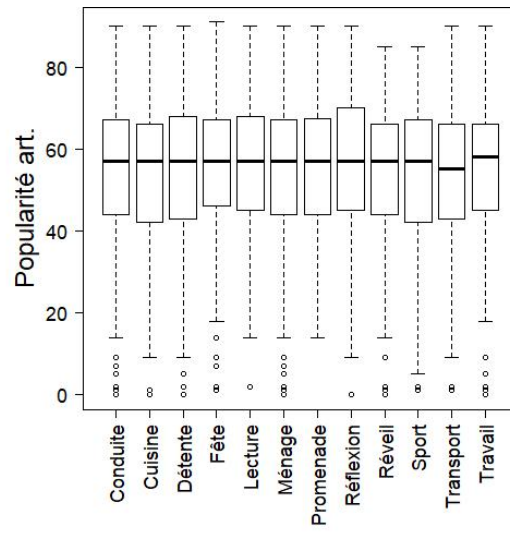
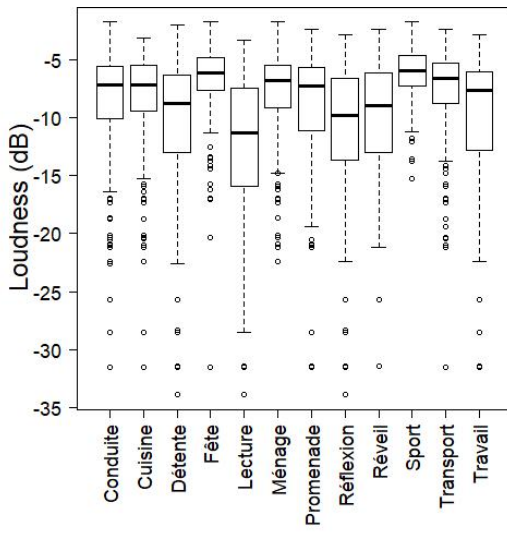
A.0.1 Comparaisons multiples du test de Kruskal-Wallis

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Détendu-En colère	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Détendu-Gracieux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Détendu-Heureux	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Détendu-Rêveur	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Détendu-Tragique	faux	vrai	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Détendu-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Détendu-Triste	faux	vrai	faux	vrai	faux	faux	faux	faux	faux	vrai	faux
Détendu-Vigoureux	vrai	faux	vrai	vrai	faux	faux	vrai	vrai	faux	faux	faux
En colère-Gracieux	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
En colère-Heureux	vrai	vrai	faux	faux	faux	faux	faux	faux	faux	vrai	faux
En colère-Rêveur	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
En colère-Tragique	vrai	faux	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
En colère-Triomphant	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
En colère-Triste	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
En colère-Vigoureux	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Gracieux-Heureux	vrai	vrai	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Gracieux-Rêveur	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Gracieux-Tragique	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Gracieux-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Gracieux-Triste	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Gracieux-Vigoureux	vrai	faux	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Heureux-Rêveur	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Heureux-Tragique	vrai	vrai	faux	vrai	vrai	faux	vrai	faux	faux	vrai	faux
Heureux-Triomphant	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Heureux-Triste	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	vrai	vrai	faux
Heureux-Vigoureux	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Rêveur-Tragique	faux	vrai	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Rêveur-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Rêveur-Triste	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Rêveur-Vigoureux	vrai	faux	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Tragique-Triomphant	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Tragique-Triste	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Tragique-Vigoureux	vrai	vrai	vrai	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Triomphant-Triste	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Triomphant-Vigoureux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Triste-Vigoureux	vrai	vrai	vrai	vrai	faux	faux	vrai	vrai	vrai	vrai	faux

Dimension “activité”

Boîtes à moustaches





Comparaisons multiples du test de Kruskal-Wallis

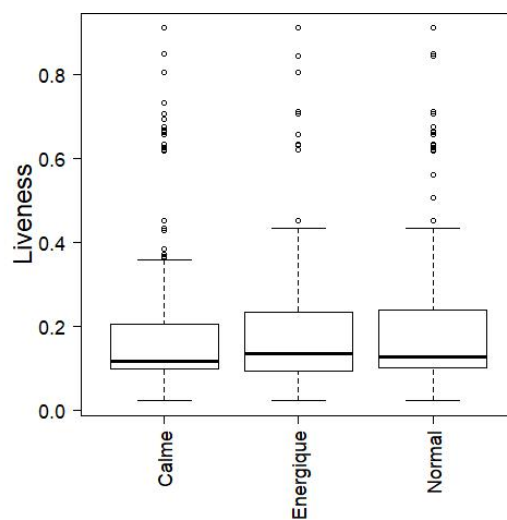
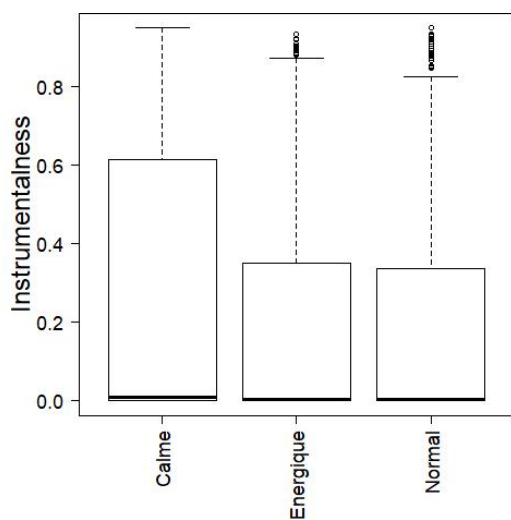
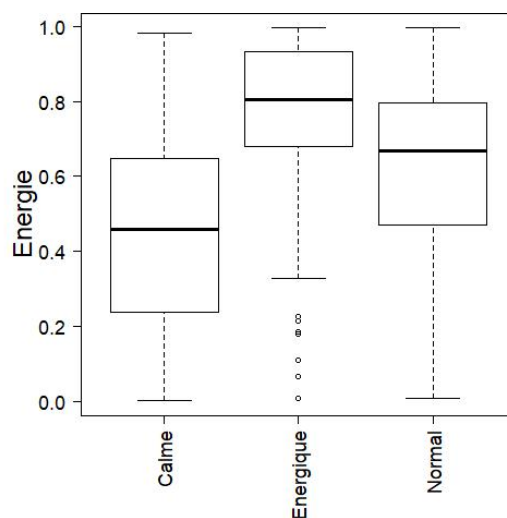
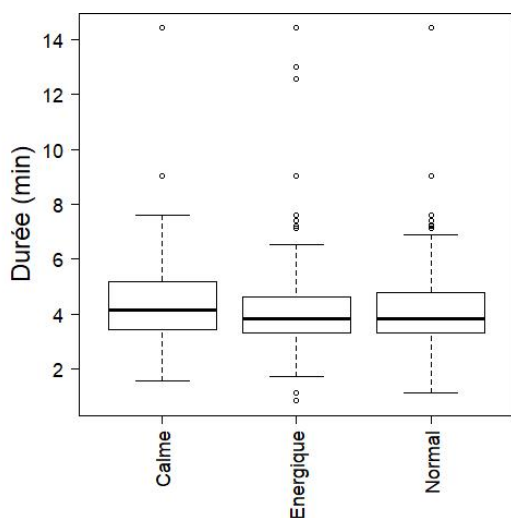
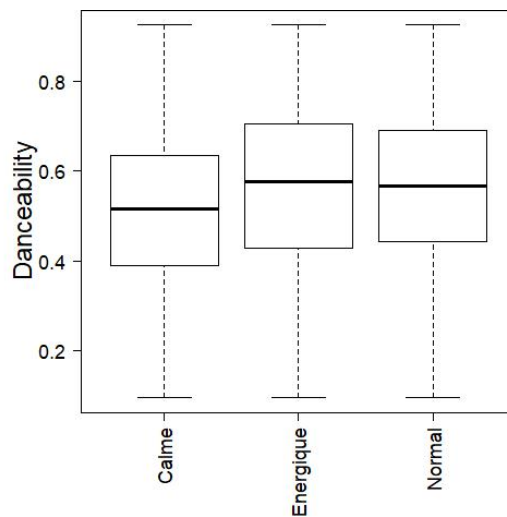
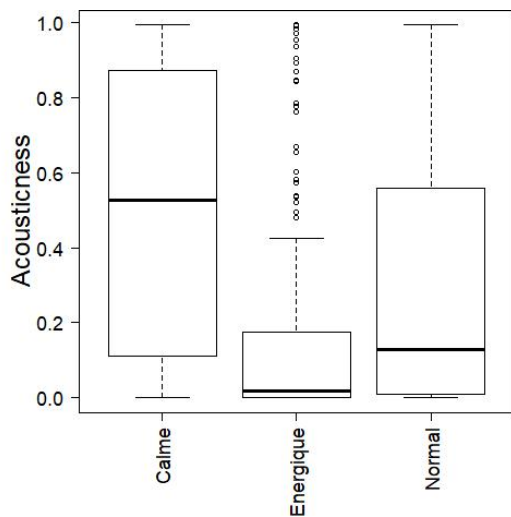
TABLE A.3 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension activité

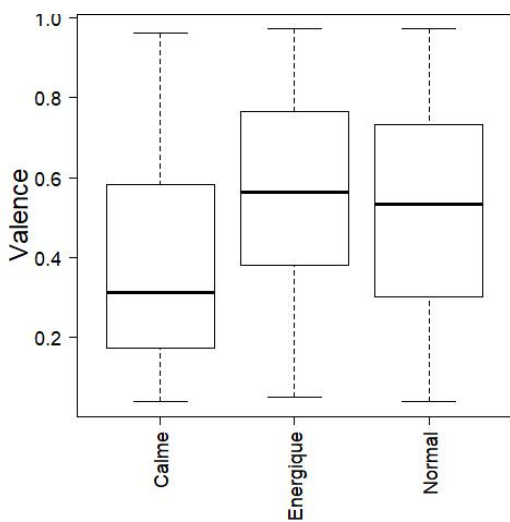
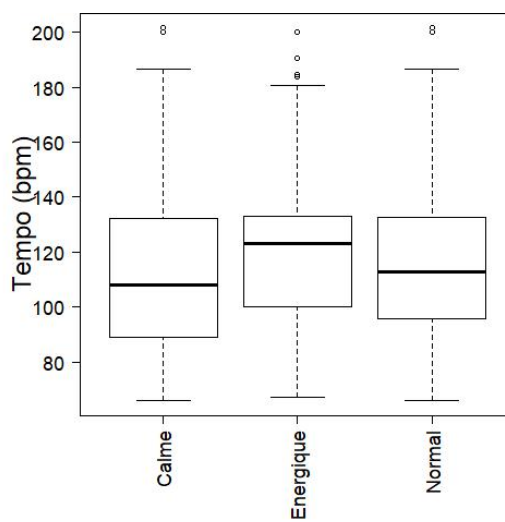
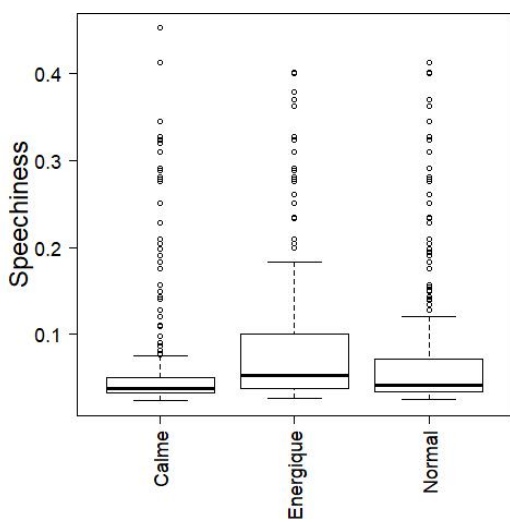
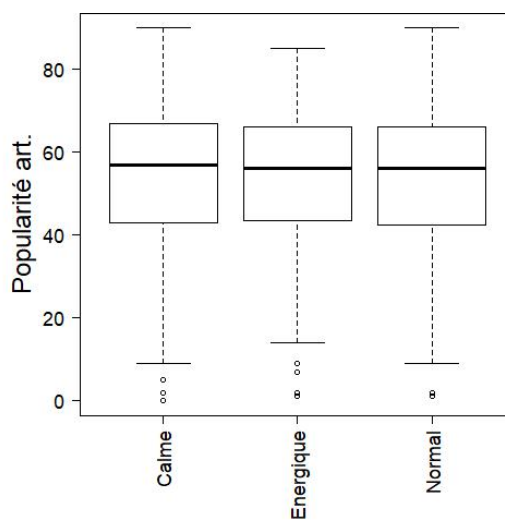
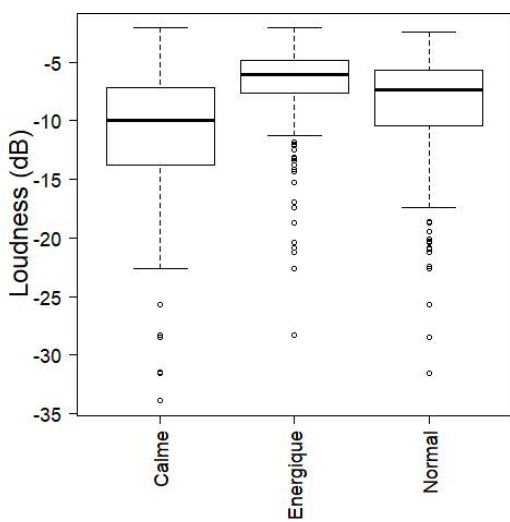
	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Conduite-Cuisine	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Conduite-Détente	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Conduite-Fête	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Conduite-Lecture	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	vrai	vrai	faux
Conduite-Ménage	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Conduite-Promenade	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Conduite-Réflexion	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Conduite-Réveil	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Conduite-Sport	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Conduite-Transport	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Conduite-Travail	faux	faux	faux	vrai	faux	faux	faux	faux	faux	faux	faux
Cuisine-Détente	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Cuisine-Fête	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Cuisine-Lecture	vrai	faux	faux	vrai	faux	faux	vrai	vrai	vrai	vrai	faux
Cuisine-Ménage	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Cuisine-Promenade	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Cuisine-Réflexion	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Cuisine-Réveil	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Cuisine-Sport	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Cuisine-Transport	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Cuisine-Travail	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Détente-Fête	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Détente-Lecture	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Détente-Ménage	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Détente-Promenade	faux	faux	faux	faux	faux	faux	vrai	faux	faux	faux	faux
Détente-Réflexion	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Détente-Réveil	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Détente-Sport	vrai	vrai	vrai	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Détente-Transport	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Détente-Travail	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Fête-Lecture	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	vrai	vrai	faux
Fête-Ménage	faux	faux	faux	vrai	faux	faux	faux	faux	faux	faux	faux
Fête-Promenade	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Fête-Réflexion	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Fête-Réveil	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Fête-Sport	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Fête-Transport	vrai	vrai	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Fête-Travail	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Lecture-Ménage	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	vrai	vrai	faux
Lecture-Promenade	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Lecture-Réflexion	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux

Lecture-Réveil	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Lecture-Sport	vrai	vrai	vrai	vrai	vrai	faux	vrai	vrai	vrai	vrai	faux
Lecture-Transport	vrai	faux	faux	vrai	vrai	faux	vrai	vrai	vrai	vrai	faux
Lecture-Travail	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Ménage-Promenade	vrai	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Ménage-Réflexion	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Ménage-Réveil	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Ménage-Sport	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux
Ménage-Transport	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Ménage-Travail	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Promenade-Réflexion	faux	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Promenade-Réveil	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Promenade-Sport	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Promenade-Transport	faux	faux	faux	vrai	faux	faux	faux	faux	faux	faux	faux
Promenade-Travail	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Réflexion-Réveil	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Réflexion-Sport	vrai	vrai	vrai	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Réflexion-Transport	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Réflexion-Travail	faux	faux	faux	vrai	faux	faux	faux	faux	faux	vrai	faux
Réveil-Sport	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Réveil-Transport	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Réveil-Travail	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Sport-Transport	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Sport-Travail	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Transport-Travail	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux

Dimension “énergie”

Boîtes à moustaches





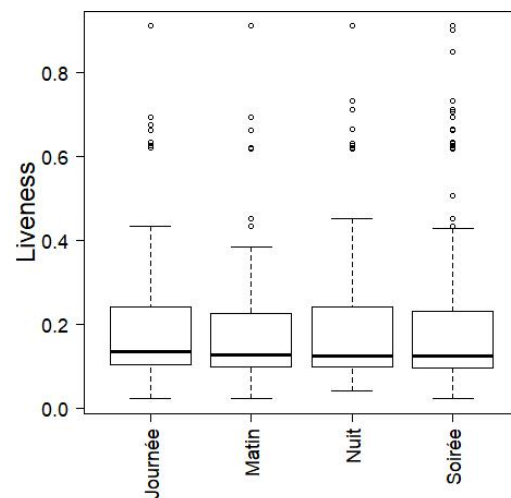
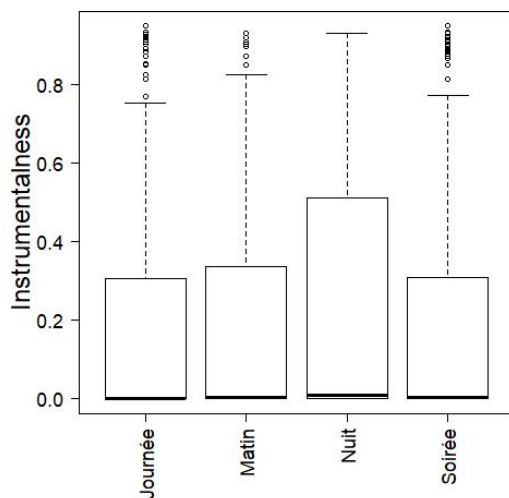
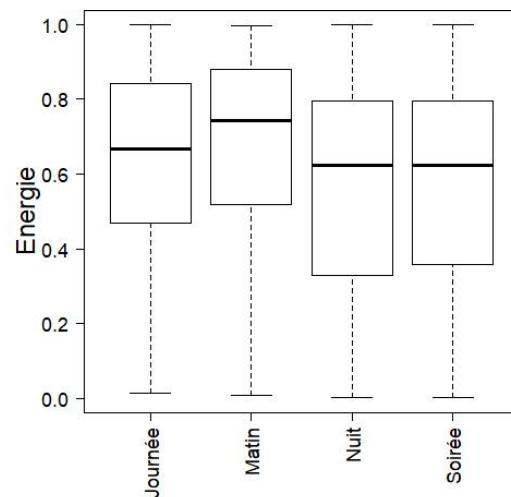
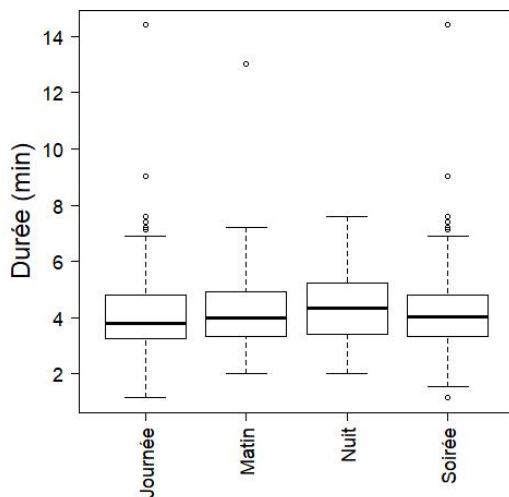
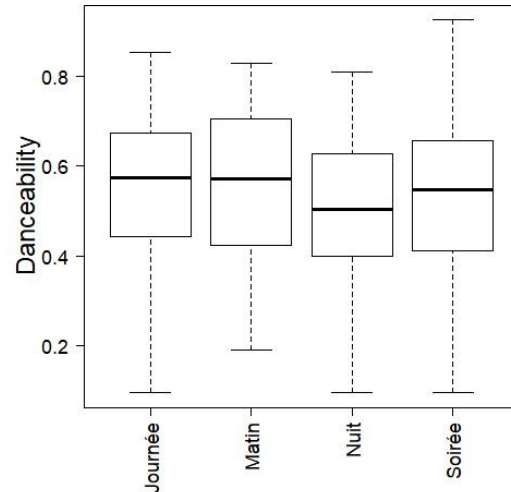
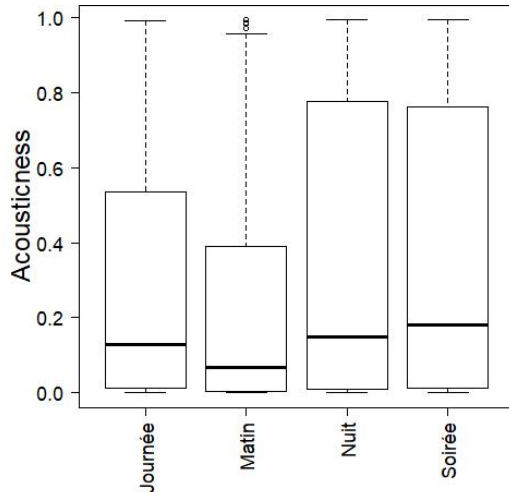
Comparaisons multiples du test de Kruskal-Wallis

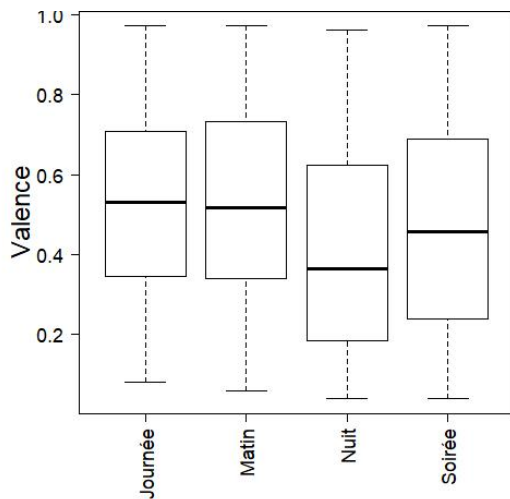
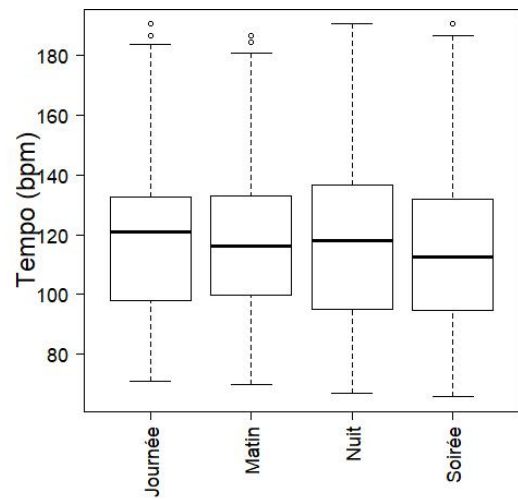
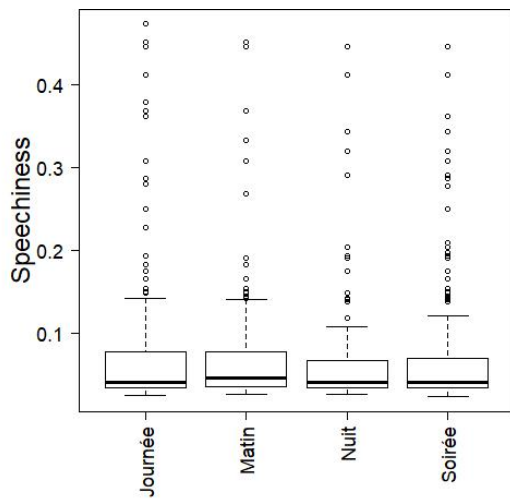
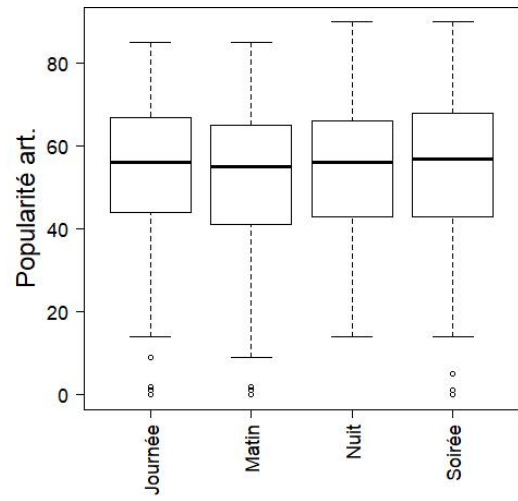
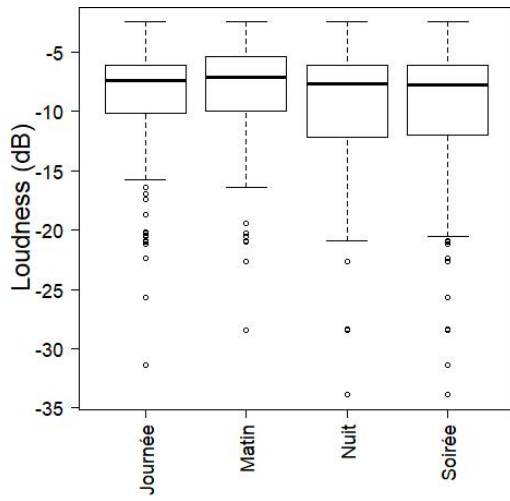
TABLE A.4 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension énergie

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Calme-Énergique	vrai	vrai	vrai	vrai	vrai	faux	vrai	vrai	vrai	vrai	faux
Calme-Normal	vrai	vrai	faux	vrai	vrai	faux	vrai	vrai	faux	vrai	faux
Énergique-Normal	vrai	faux	faux	vrai	faux	faux	vrai	vrai	vrai	faux	faux

Dimension “jour”

Boîtes à moustaches



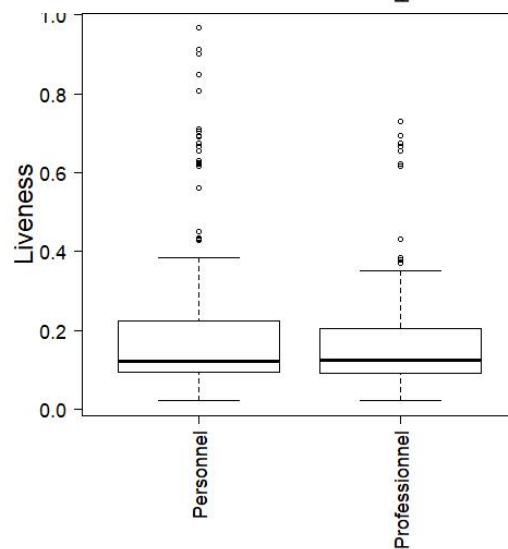
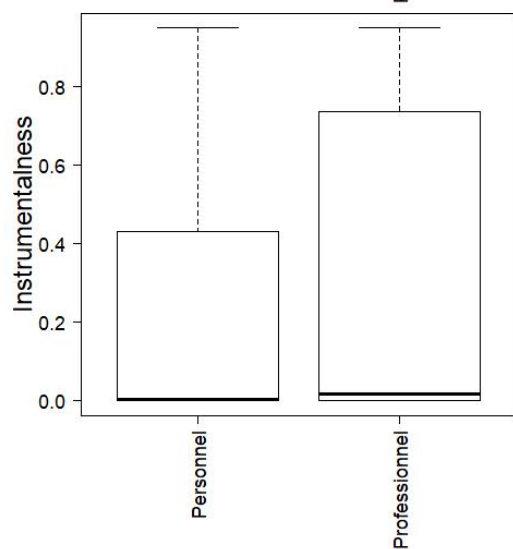
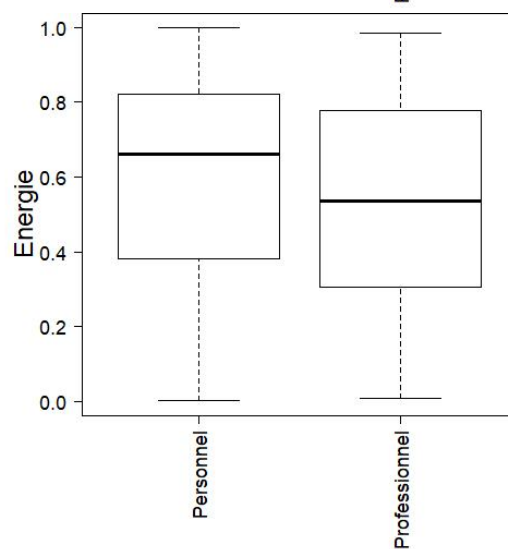
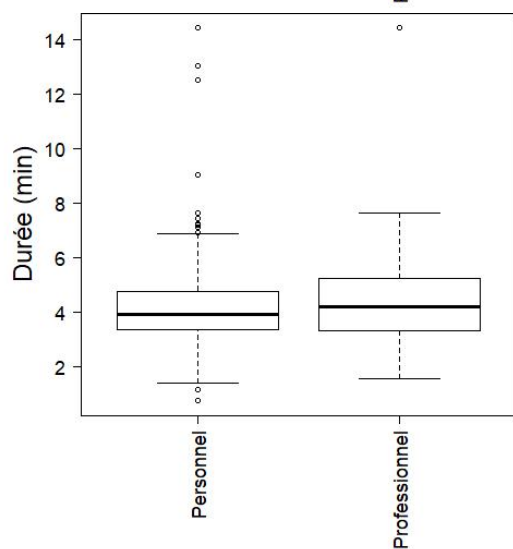
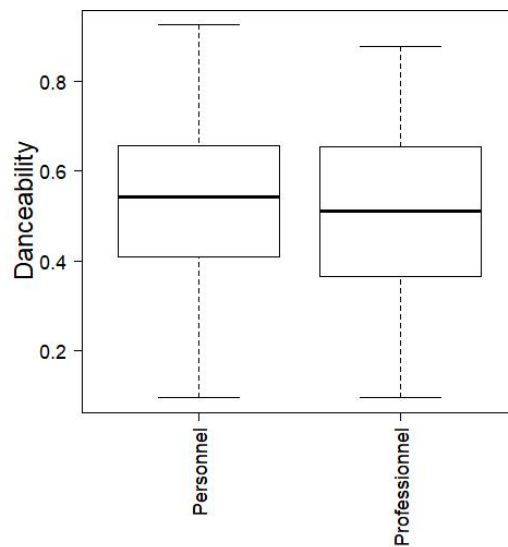
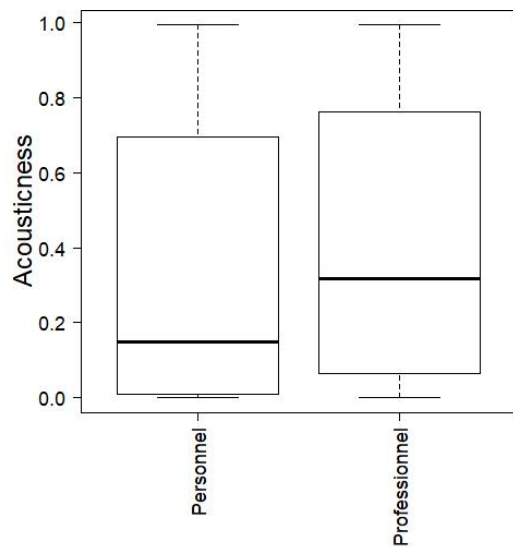


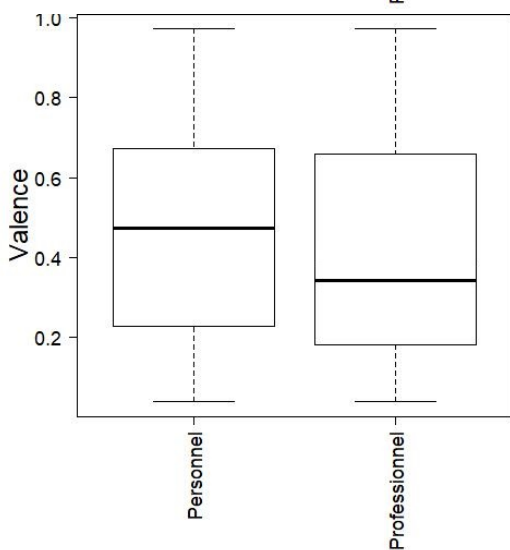
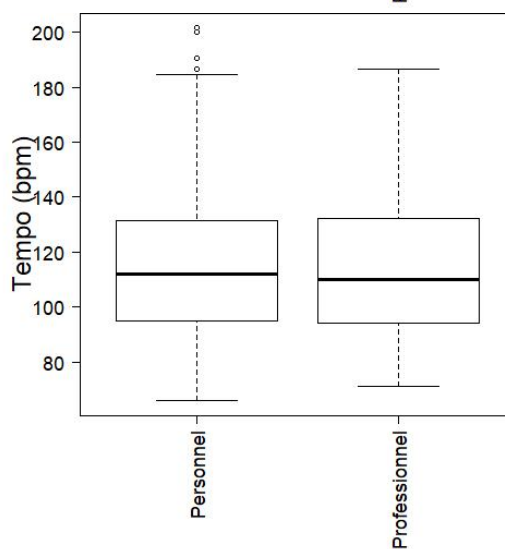
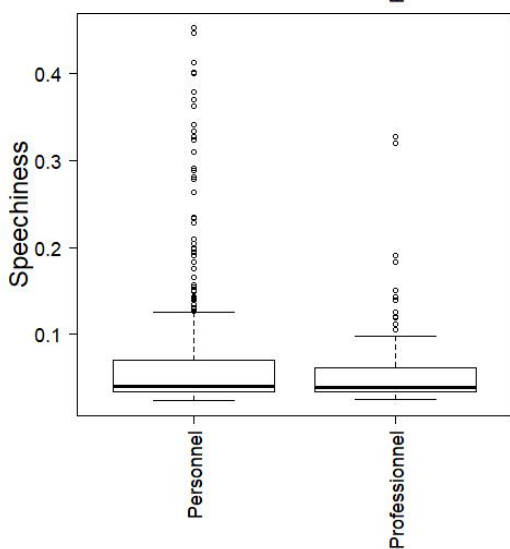
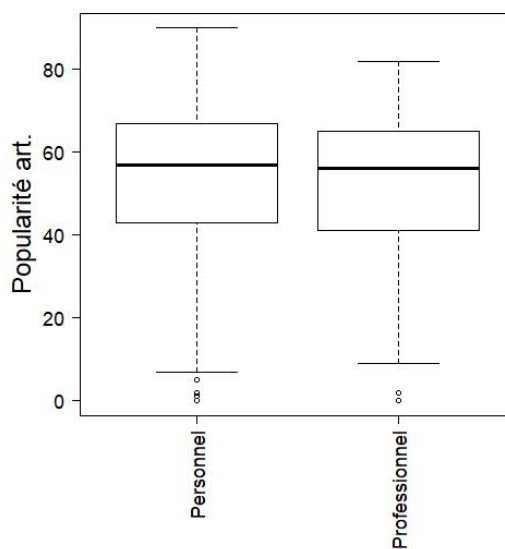
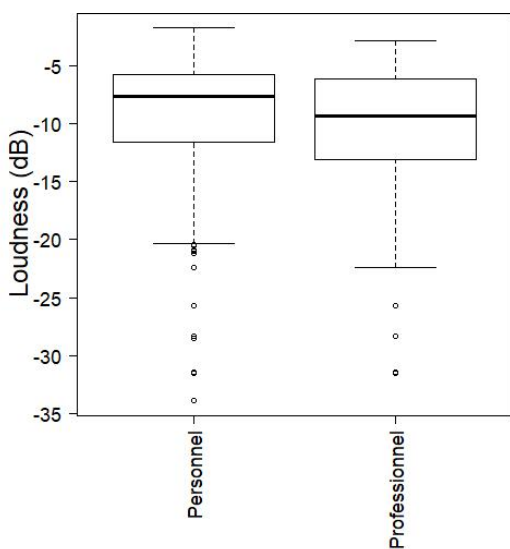
Comparaisons multiples du test de Kruskal-Wallis

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Journée-Matin	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Journée-Nuit	faux	faux	faux	faux	faux	faux	faux	faux	faux	vrai	faux
Journée-Soirée	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Matin-Nuit	faux	faux	faux	vrai	faux	faux	faux	faux	faux	vrai	faux
Matin-Soirée	faux	faux	faux	vrai	faux	faux	faux	faux	faux	faux	faux
Nuit-Soirée	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux

Dimension “environnement”

Boîtes à moustaches





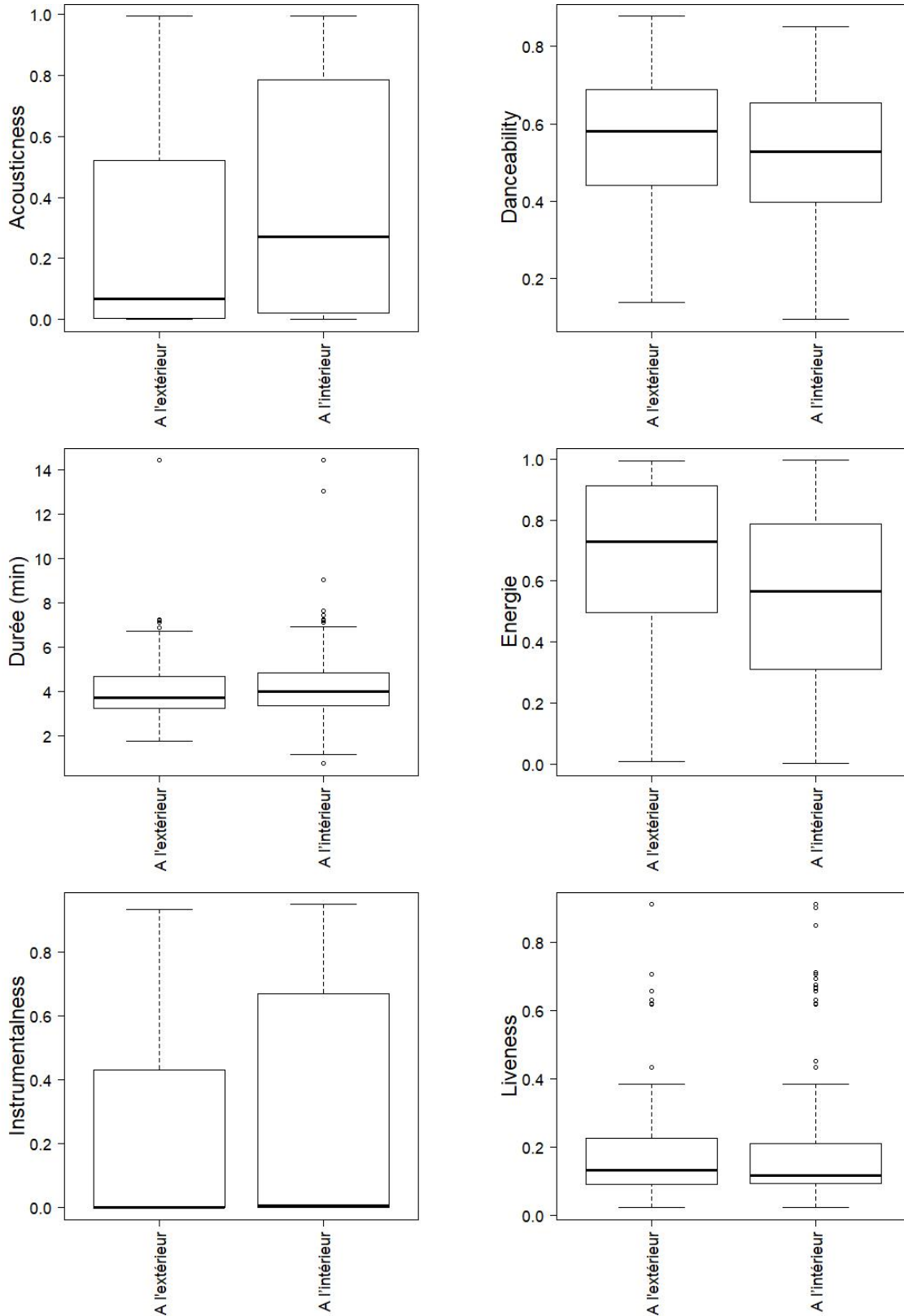
Comparaisons multiples du test de Kruskal-Wallis

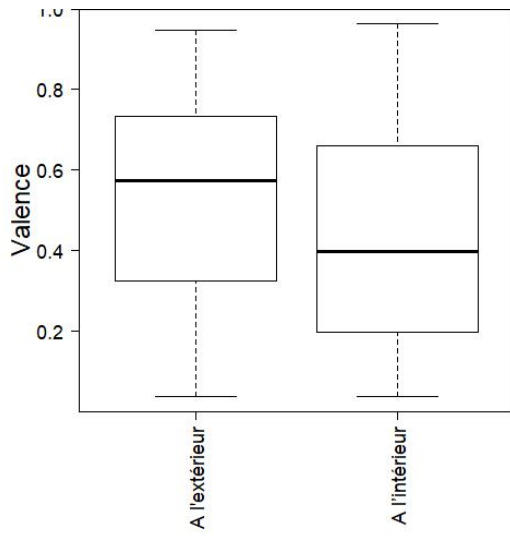
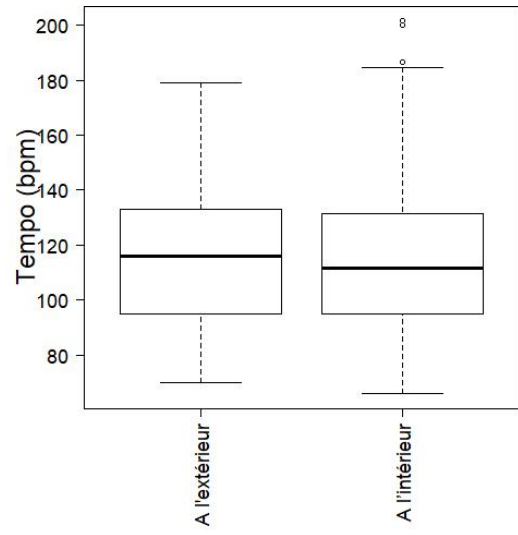
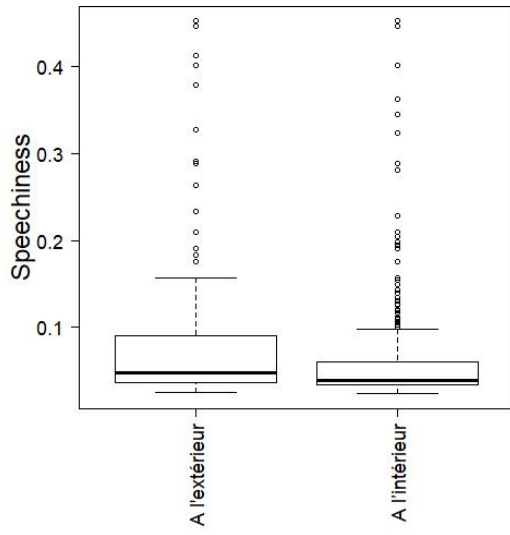
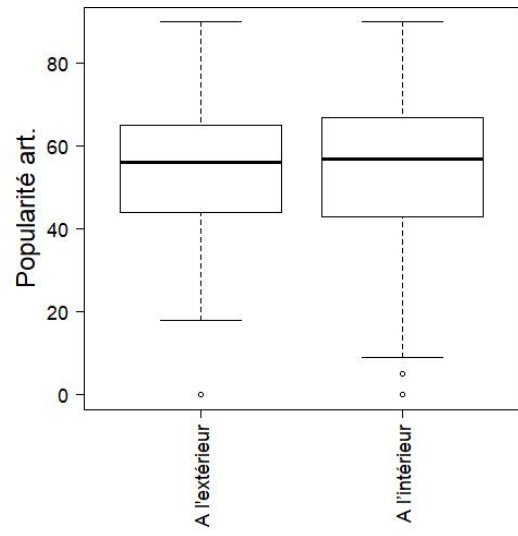
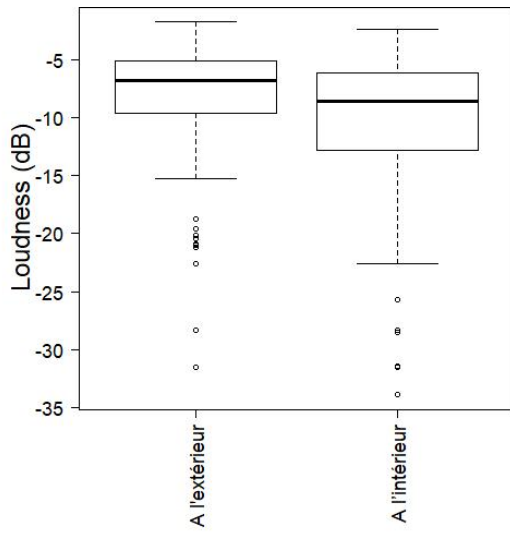
TABLE A.6 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension environnement

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Perso.-Profess.	vrai	faux	faux	vrai	vrai	faux	vrai	faux	faux	vrai	faux

Dimension “lieu”

Boîtes à moustaches





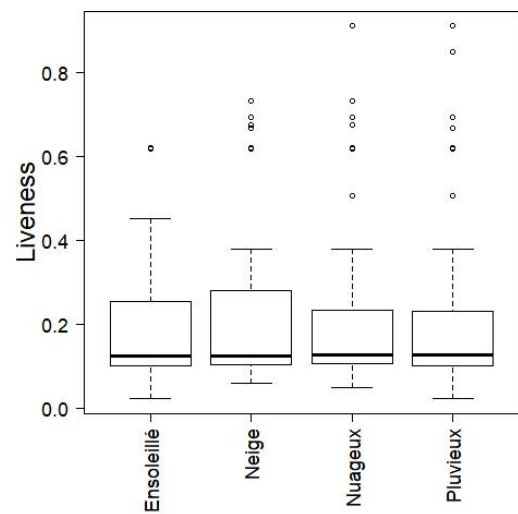
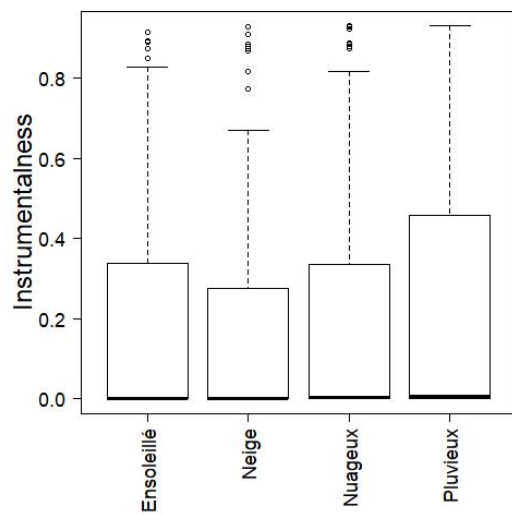
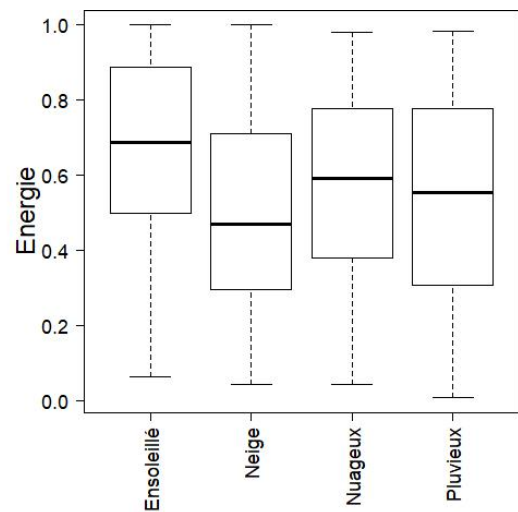
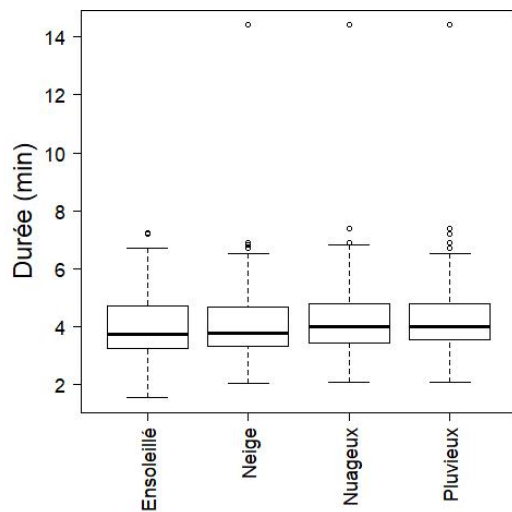
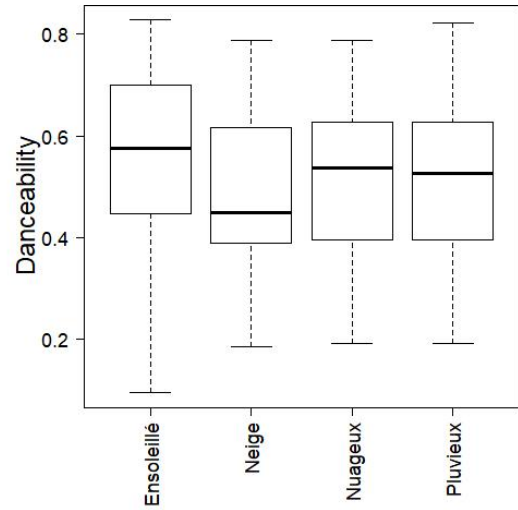
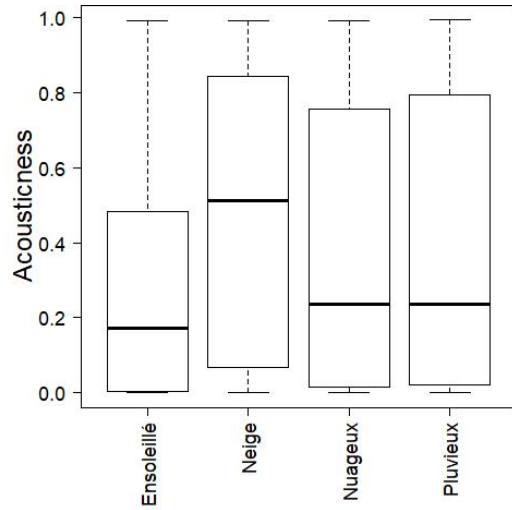
Comparaisons multiples du test de Kruskal-Wallis

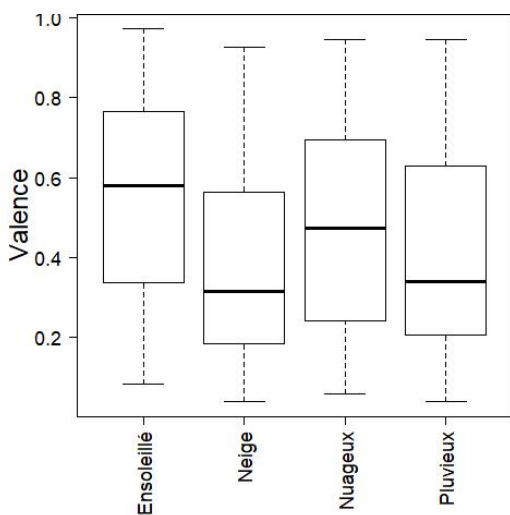
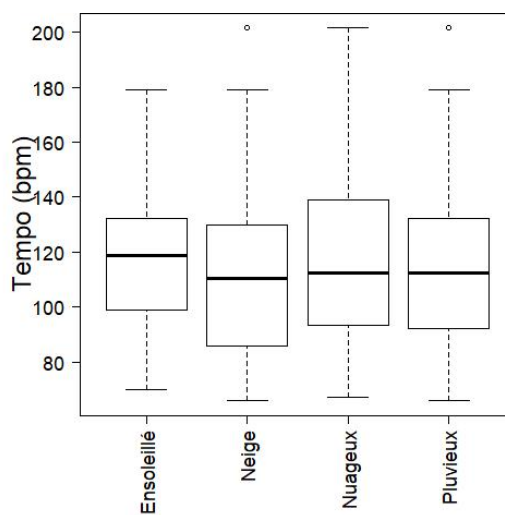
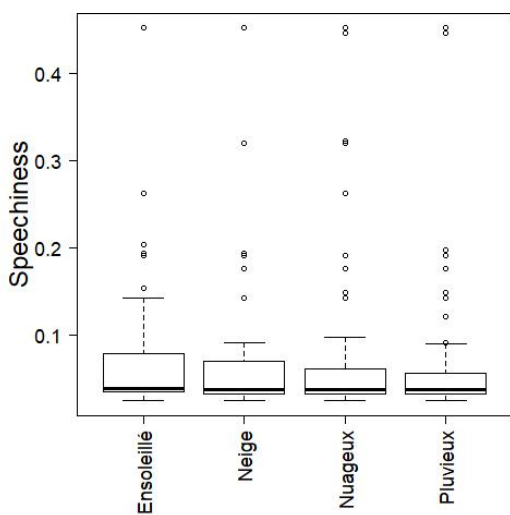
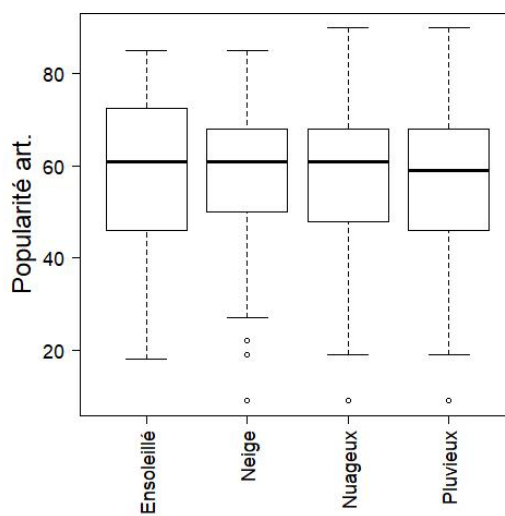
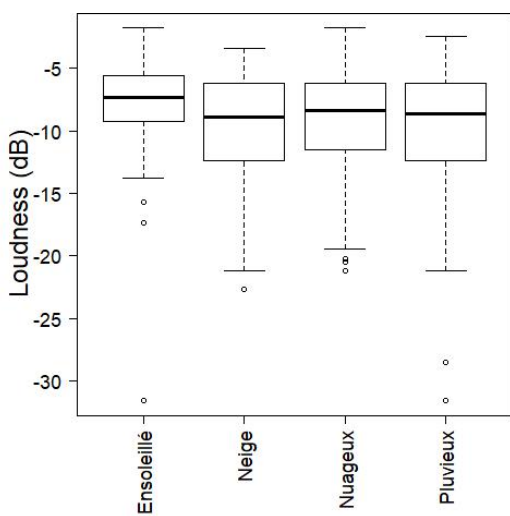
TABLE A.7 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension lieu

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
À l'extérieur-A l'intérieur	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux

Dimension “météo”

Boîtes à moustaches





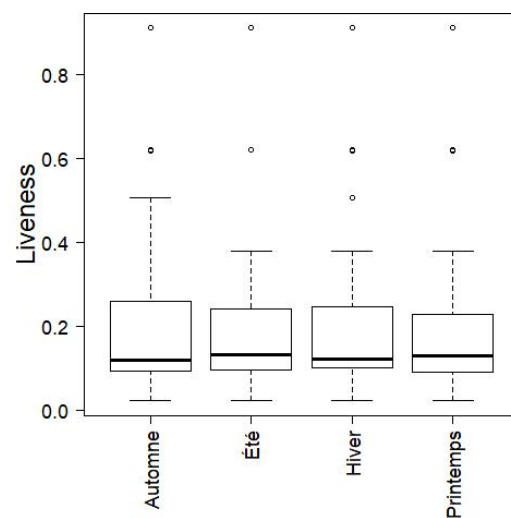
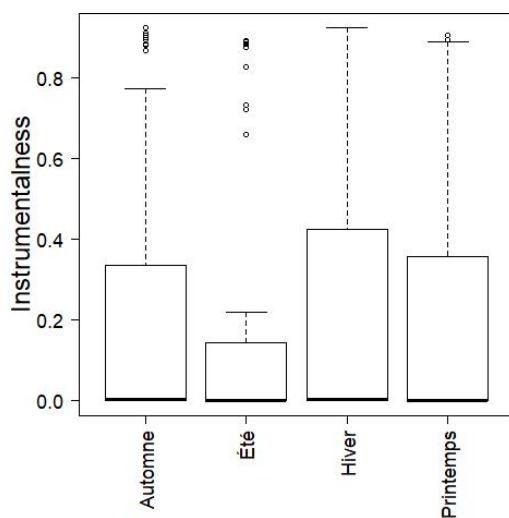
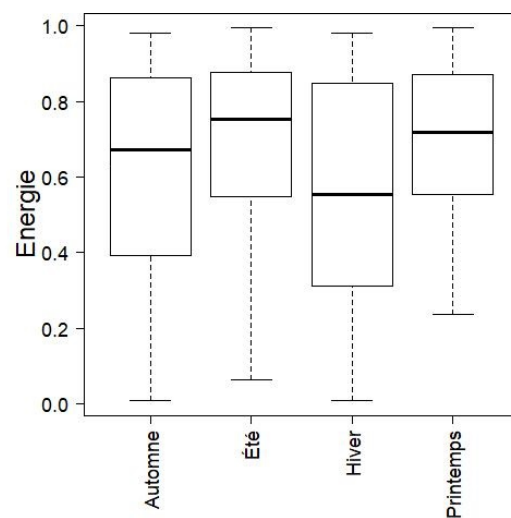
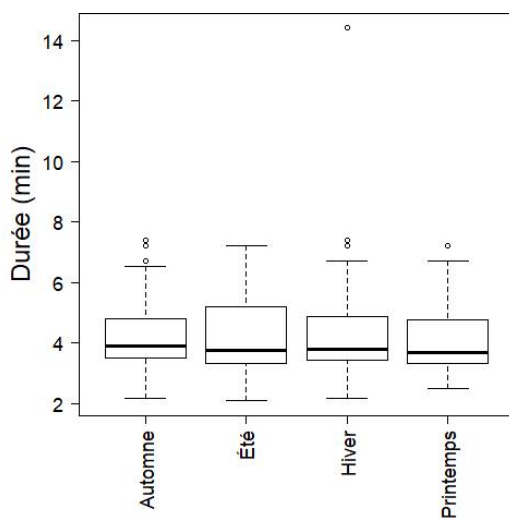
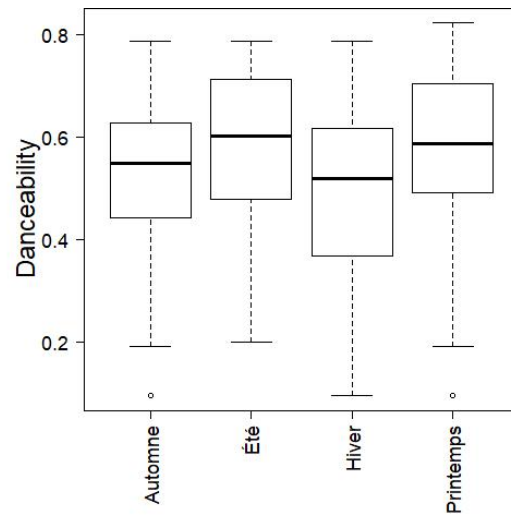
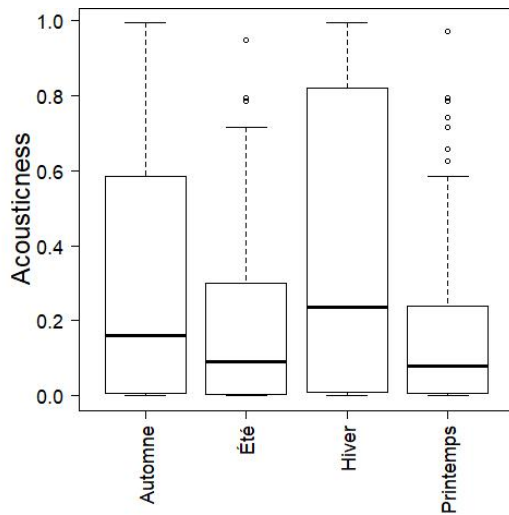
Comparaisons multiples du test de Kruskal-Wallis

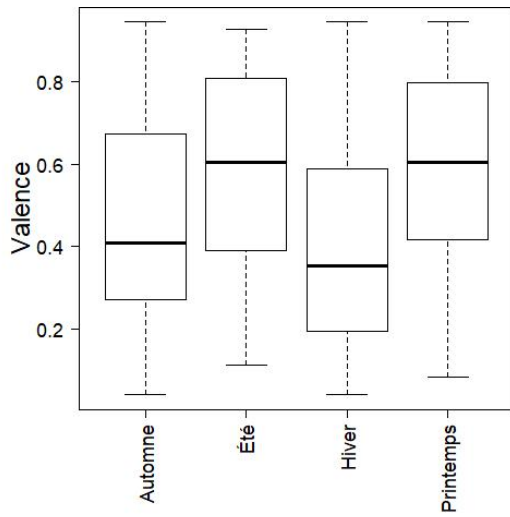
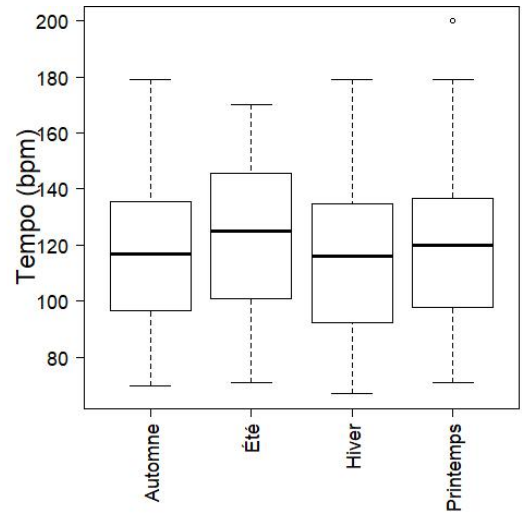
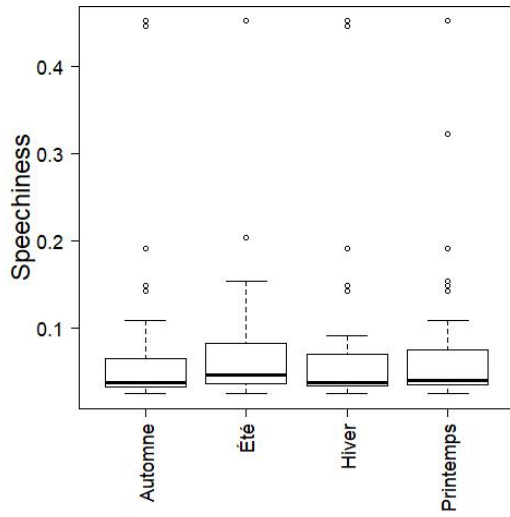
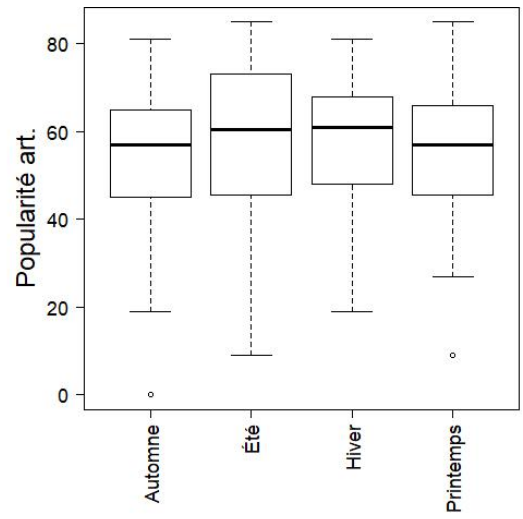
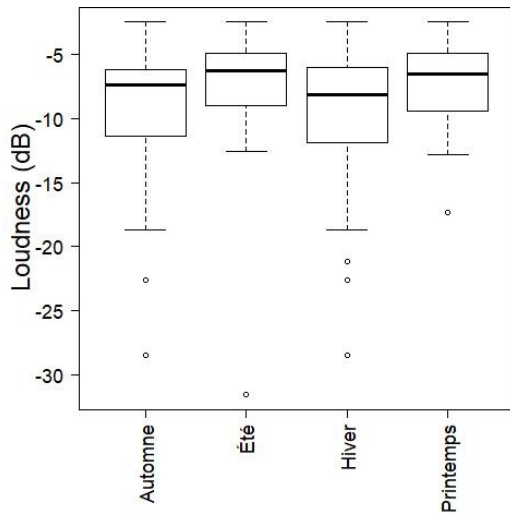
TABLE A.8 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension météo

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Ensoleillé-Neige	vrai	faux	faux	vrai	faux	faux	faux	faux	faux	vrai	faux
Ensoleillé-Nuageux	faux	vrai	faux	faux	faux	faux	faux	faux	faux	faux	faux
Ensoleillé-Pluvieux	faux	faux	faux	vrai	faux	faux	faux	faux	faux	vrai	faux
Neige-Nuageux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Neige-Pluvieux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Nuageux-Pluvieux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux

Dimension “saison”

Boîtes à moustaches





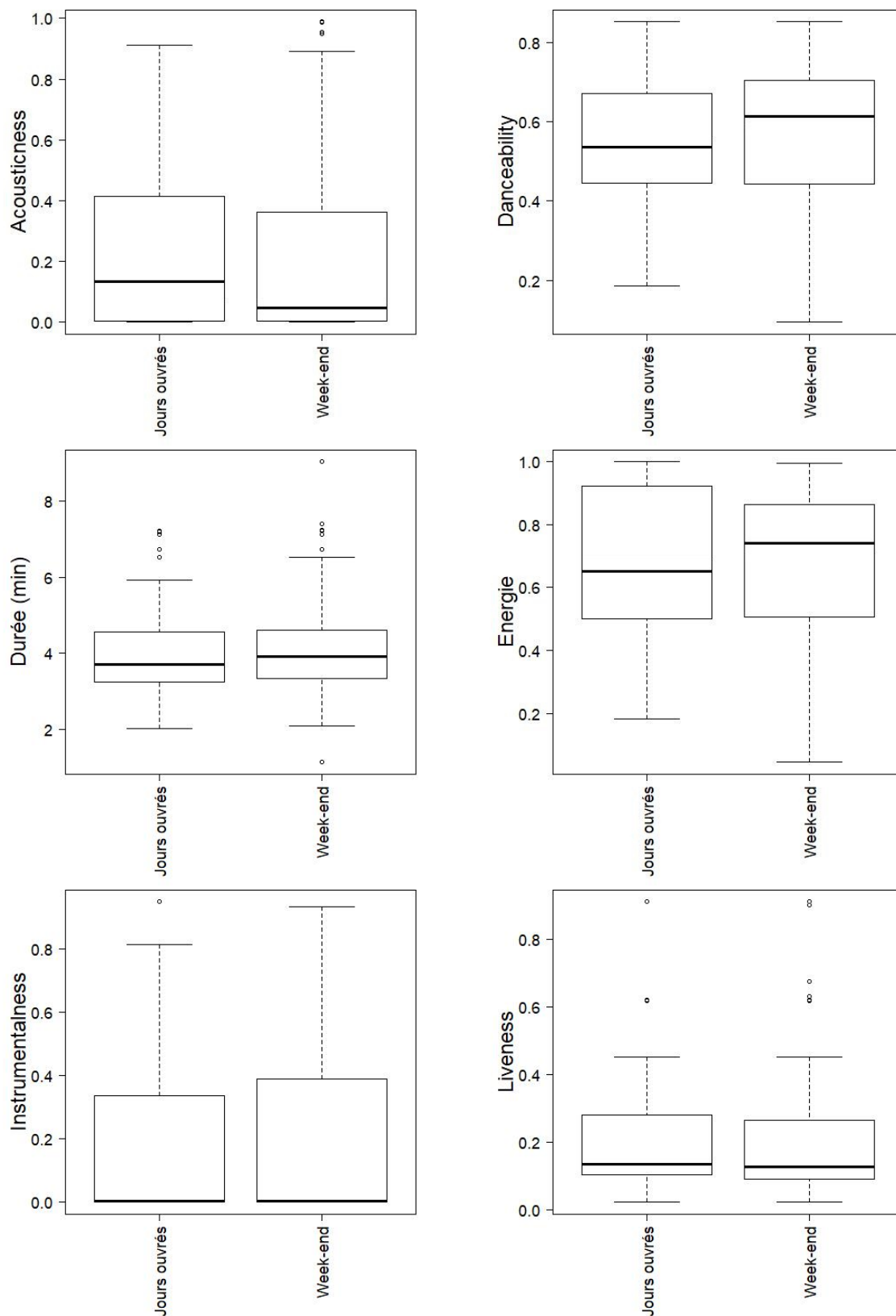
Comparaisons multiples du test de Kruskal-Wallis

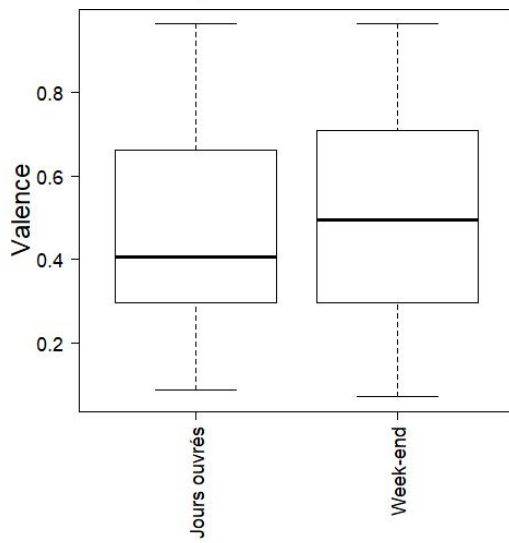
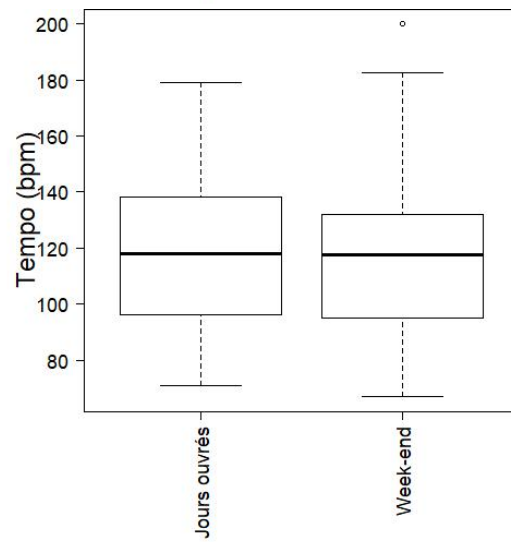
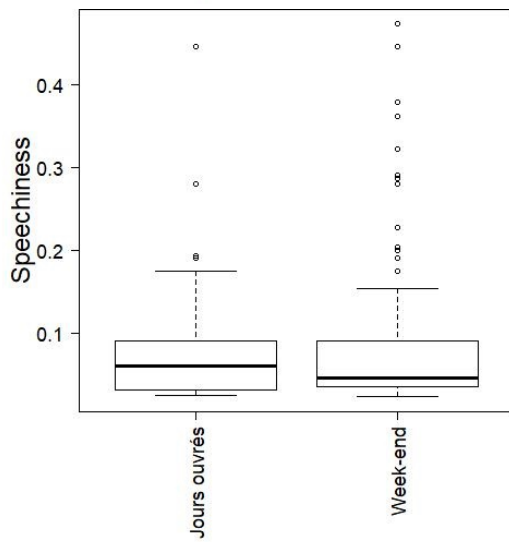
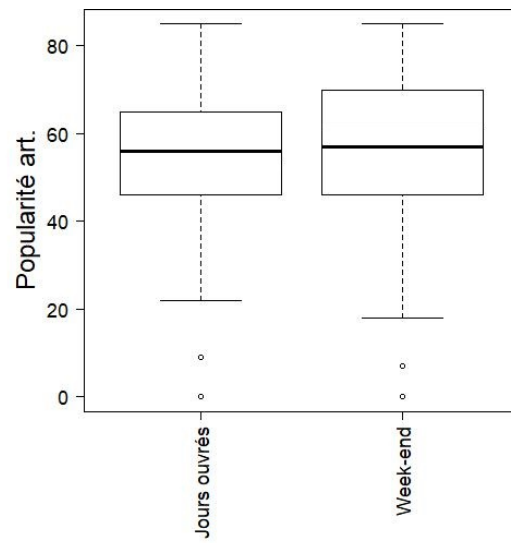
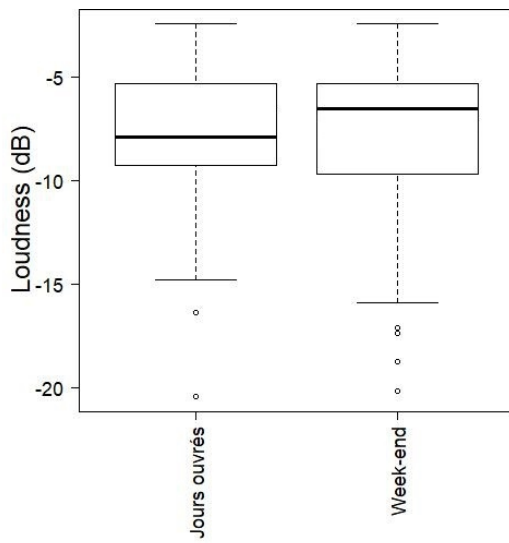
TABLE A.9 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension saison

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Automne-Été	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Automne-Hiver	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Automne-Printemps	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Été-Hiver	vrai	vrai	faux	vrai	faux	faux	vrai	faux	faux	vrai	faux
Été-Printemps	vrai	faux	faux	vrai	faux	faux	vrai	faux	faux	faux	faux
Hiver-Printemps	faux	vrai	faux	faux	faux	faux	faux	faux	faux	vrai	faux

Dimension “semaine”

Boîtes à moustaches





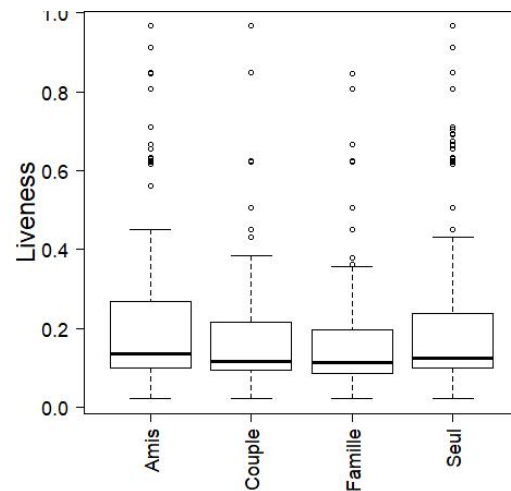
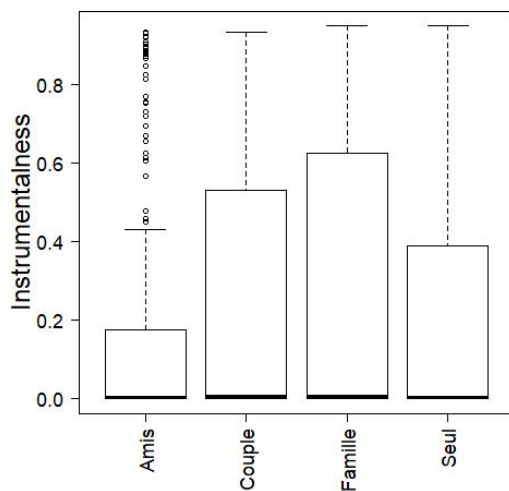
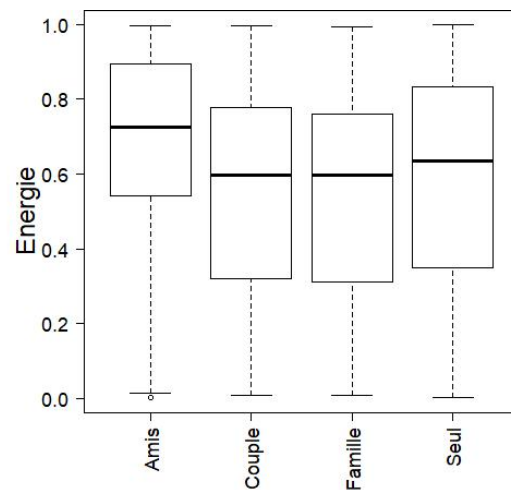
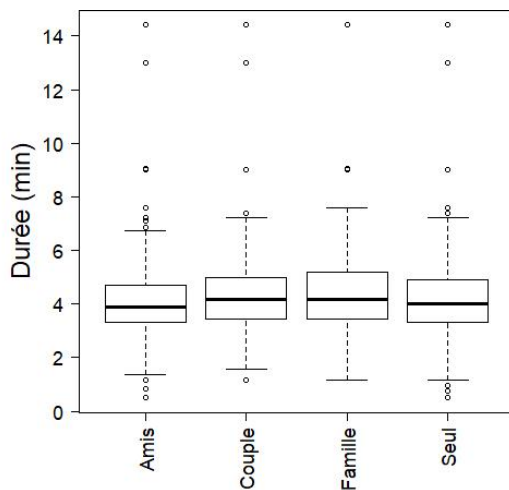
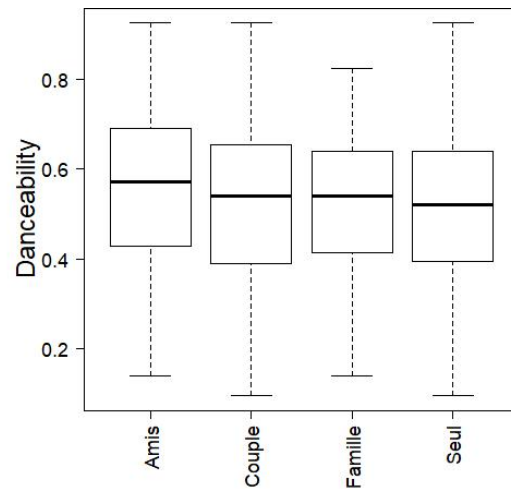
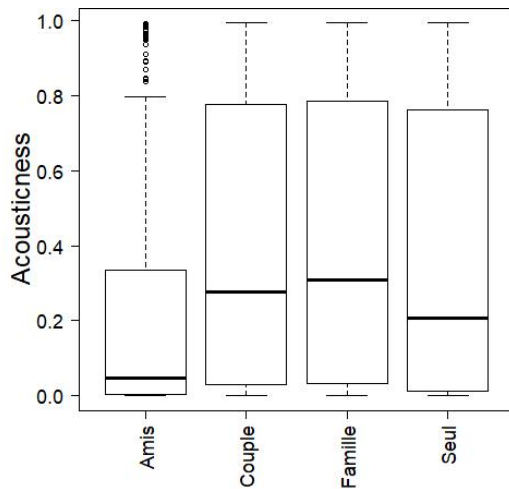
Comparaisons multiples du test de Kruskal-Wallis

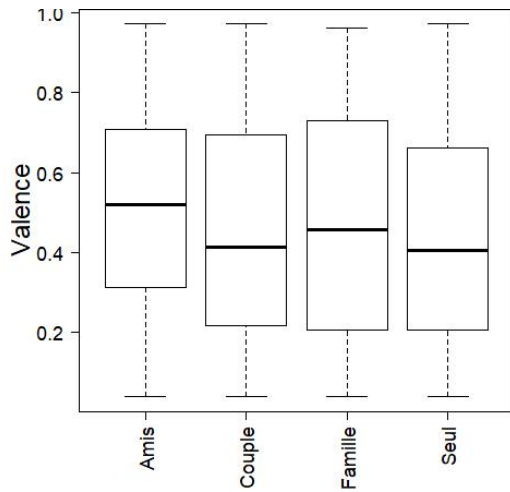
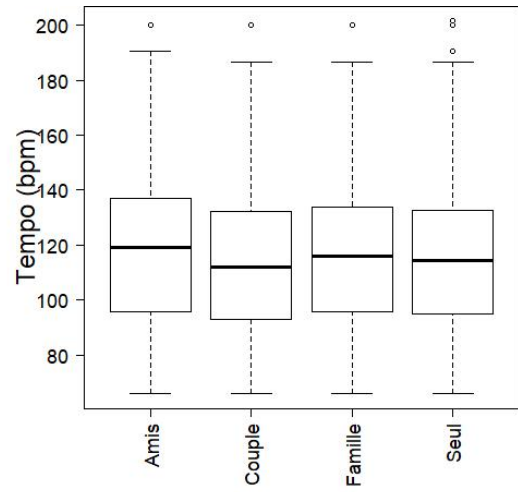
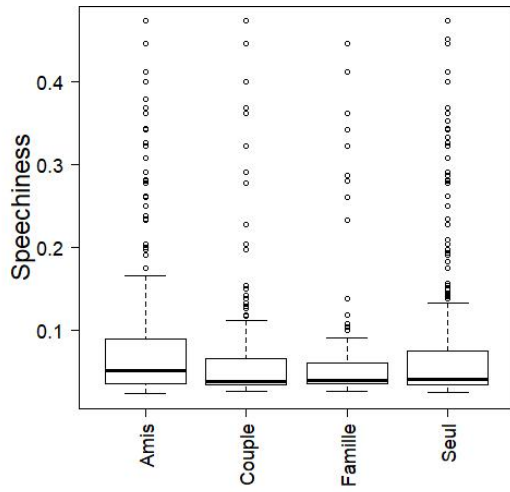
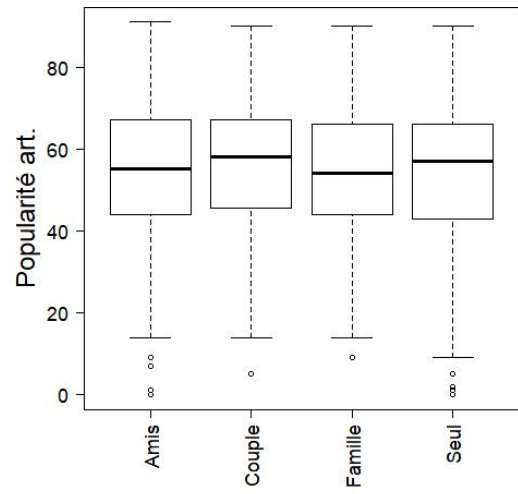
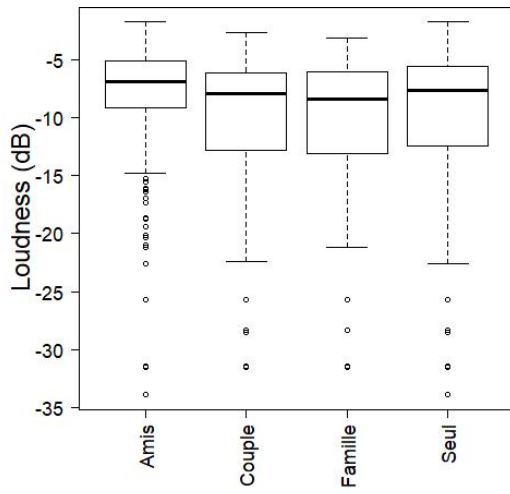
TABLE A.10 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension lieu

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Jours ouvrés-Week-end	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	faux	faux

Dimension “social”

Boîtes à moustaches





Comparaisons multiples du test de Kruskal-Wallis

TABLE A.11 – Comparaisons multiples du test de Kruskal-Wallis pour la dimension sociale

	acou.	danc.	dur.	éne.	inst.	live.	loud.	spee.	temp.	val.	pop.
Amis-Couple	vrai	faux	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Amis-Famille	vrai	faux	faux	vrai	faux	vrai	vrai	vrai	faux	faux	faux
Amis-Seul	vrai	vrai	faux	vrai	faux	faux	vrai	vrai	faux	vrai	faux
Couple-Famille	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Couple-Seul	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux
Famille-Seul	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux	faux

Bibliographie

- [A and van Schnydel R, 2002] A, U. and van Schnydel R (2002). A review of factors affecting music recommender success! In 3rd international conference on music information retrieval, pages 44–50.
- [Abbassi et al., 2009] Abbassi, Z., Amer-Yahia, S., Lakshmanan, L. V., Vassilvitskii, S., and Yu, C. (2009). Getting recommender systems to think outside the box. In Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, pages 285–288, New York, NY, USA. ACM.
- [Abowd et al., 1999] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Steggles, P. (1999). Towards a better understanding of context and context-awareness. In Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing, HUC '99, pages 304–307, London, UK, UK. Springer-Verlag.
- [Adomavicius and Kwon, 2012] Adomavicius, G. and Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. IEEE Trans. on Knowl. and Data Eng., 24(5) :896–911.
- [Adomavicius et al., 2011a] Adomavicius, G., Manouselis, N., and Kwon, Y. (2011a). Multi-Criteria Recommender Systems, pages 769–803. Springer US, Boston, MA.
- [Adomavicius et al., 2011b] Adomavicius, G., Mobasher, B., Ricci, F., and Tuzhilin, A. (2011b). Context-aware recommender systems. AI Magazine, 32(3) :67–80.
- [Adomavicius and Tuzhilin, 2001] Adomavicius, G. and Tuzhilin, A. (2001). Expert-driven validation of rule-based user models in personalization applications. Data Min. Knowl. Discov., 5(1/2) :33–58.
- [Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems : a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6) :734–749.
- [Adomavicius and Tuzhilin, 2011] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. Recommender Systems Handbook, pages 217–253.
- [Aggarwal, 2015] Aggarwal, C. C. (2015). Outlier analysis. In Data mining, pages 237–263. Springer International Publishing.
- [Aggarwal, 2016a] Aggarwal, C. C. (2016a). Content-Based Recommender Systems, pages 139–166. Springer International Publishing, Cham.
- [Aggarwal, 2016b] Aggarwal, C. C. (2016b). Evaluating Recommender Systems, pages 225–254. Springer International Publishing, Cham.
- [Agrawal et al., 2009] Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In Proceedings of the second ACM International Conference on Web Search and Data Mining, WSDM'09, Barcelona, Spain.

- [Al-Shamri, 2016] Al-Shamri, M. Y. H. (2016). User profiling approaches for demographic recommender systems. *Know.-Based Syst.*, 100(C) :175–187.
- [Al-Taie, 2013] Al-Taie, M. Z. (2013). Explanations in recommender systems : Overview and research approaches. In *Proceedings of the Fifth ACM Conference on Recommender Systems, ACIT'2013*.
- [Aleksandrova et al., 2017] Aleksandrova, M., Brun, A., Boyer, A., and Chertov, O. (2017). Identifying representative users in matrix factorization-based recommender systems : application to solving the content-less new item cold-start problem. *J. Intell. Inf. Syst.*, 48(2) :365–397.
- [Ardissono et al., 2003] Ardissono, L., Goy, A., Petrone, G., Segnan, M., and Torasso, P. (2003). Intrigue : Personalized recommendation of tourist attractions for desktop and hand held devices. *Applied Artificial Intelligence*, 17 :687–714.
- [Aytekin and Karakaya, 2014] Aytekin, T. and Karakaya, M. O. (2014). Clustering-based diversity improvement in top-n recommendation. *J. Intell. Inf. Syst.*, 42(1) :1–18.
- [Azaria et al., 2013] Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., and Neta-nely, I. (2013). Movie recommender system for profit maximization. In *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13*, pages 121–128, New York, NY, USA. ACM.
- [Baizal et al., 2016] Baizal, Z. K. A., Widyantoro, D. H., and Maulidevi, N. U. (2016). Design of knowledge for conversational recommender system based on product functional requirements. In *2016 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6.
- [Balabanović and Shoham, 1997] Balabanović, M. and Shoham, Y. (1997). Fab : Content-based, collaborative recommendation. *Commun. ACM*, 40(3) :66–72.
- [Baltrunas and Amatriain, 2009] Baltrunas, L. and Amatriain, X. (2009). Towards Time-Dependant Recommendation based on Implicit Feedback. In *Context-aware Recommender Systems Workshop at Recsys09*.
- [Baltrunas et al., 2011a] Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K.-H., and Schwaiger, R. (2011a). Incarmusic : Context-aware music recommendations in a car. In Huemer, C. and Setzer, T., editors, *EC-Web*, volume 85 of *Lecture Notes in Business Information Processing*, pages 89–100. Springer.
- [Baltrunas et al., 2012] Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Comput.*, 16(5) :507–526.
- [Baltrunas et al., 2011b] Baltrunas, L., Ludwig, B., and Ricci, F. (2011b). Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, pages 301–304, New York, NY, USA. ACM.
- [Baltrunas and Ricci, 2014] Baltrunas, L. and Ricci, F. (2014). Experimental evaluation of context-dependent collaborative filtering using item splitting. *User Modeling and User-Adapted Interaction*, 24(1-2) :7–34.
- [Bamshad, 2010] Bamshad, M. (2010). Contextual user modeling for recommendation. In *Keynote at the 2nd Workshop on Context-Aware Recommender*.
- [Barthet et al., 2013] Barthet, M., Fazekas, G., and Sandler, M. (2013). Music emotion recognition : From content- to context-based models. In Aramaki, M., Barthet, M., Kronland-Martinet, R., and Ystad, S., editors, *From Sounds to Music and Emotions*, pages 228–252, Berlin, Heidelberg. Springer Berlin Heidelberg.

-
- [Bazire and Brézillon, 2005] Bazire, M. and Brézillon, P. (2005). Understanding context before using it. In Proceedings of the 5th International Conference on Modeling and Using Context, CONTEXT'05, pages 29–40, Berlin, Heidelberg. Springer-Verlag.
- [Bedi et al., 2014] Bedi, P., Agarwal, S. K., Jindal, V., and Richa (2014). Marst : Multi-agent recommender system for e-tourism using reputation based collaborative filtering. In Madaan, A., Kikuchi, S., and Bhalla, S., editors, Databases in Networked Information Systems, pages 189–201, Cham. Springer International Publishing.
- [Belkin and Croft, 1992] Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval : Two sides of the same coin ? Commun. ACM, 35(12) :29–38.
- [Bellogín et al., 2017] Bellogín, A., Castells, P., and Cantador, I. (2017). Statistical biases in information retrieval metrics for recommender systems. Information Retrieval Journal, 20(6) :606–634.
- [Beltran et al., 2012] Beltran, V., Arabshian, K., and Schulzrinne, H. (2012). Ontology-Based User-Defined Rules and Context-Aware Service Composition System, pages 139–155. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Bilgic, 2005] Bilgic, M. (2005). Explaining recommendations : Satisfaction vs. promotion. In In Proceedings of Beyond Personalization 2005, the Workshop on the Next Stage of Recommender Systems Research(IUI2005), pages 13–18.
- [Billsus and Pazzani, 2000] Billsus, D. and Pazzani, M. J. (2000). User modeling for adaptive news access. User Modeling and User-Adapted Interaction, 10(2) :147–180.
- [Birukov et al., 2005] Birukov, A., Blanzieri, E., and Giorgini, P. (2005). Implicit : An agent-based recommendation system for web search. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '05, pages 618–624, New York, NY, USA. ACM.
- [Boim et al., 2011] Boim, R., Milo, T., and Novgorodov, S. (2011). Diversification and refinement in collaborative filtering recommender. In Proceedings of the 20 ACM international conference on Information and knowledge management, CIKM'11, Glasgow, UK.
- [Bonnardel, 2009] Bonnardel, N. (2009). Activités de conception et créativité : de l'analyse des facteurs cognitifs à l'assistance aux activités de conception créatives, pages 5–22. Lavoisier.
- [Bonnin et al., 2012] Bonnin, G., Brun, A., and Boyer, A. (2012). Exploitation du skipping pour la modélisation prédictive des usages du web. Vers une meilleure prise en compte du bruit. Revue des Sciences et Technologies de l'Information - Série RIA : Revue d'Intelligence Artificielle, 26(6) :609–642.
- [Bonnin and Jannach, 2014] Bonnin, G. and Jannach, D. (2014). Automated generation of music playlists : Survey and experiments. ACM Comput. Surv., 47(2) :26 :1–26 :35.
- [Bosch et al., 2016] Bosch, A., Bogers, T., and Kunder, M. (2016). Estimating search engine index size variability : A 9-year longitudinal study. Scientometrics, 107(2) :839–856.
- [Bouneffouf et al., 2012] Bouneffouf, D., Bouzeghoub, A., and Gançarski, A. L. (2012). A contextual-bandit algorithm for mobile context-aware recommender system. In Proceedings of the 19th International Conference on Neural Information Processing - Volume Part III, ICONIP'12, pages 324–331, Berlin, Heidelberg. Springer-Verlag.
- [Bradley and Smith, 2001] Bradley, K. and Smith, B. (2001). Improving recommendation diversity. In Irish Conference on Artificial Intelligence and Cognitive Science, AICS'01, pages 85–94.

- [Braunhofer et al., 2014] Braunhofer, M., Codina, V., and Ricci, F. (2014). Switching hybrid for cold-starting context-aware recommender systems. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 349–352, New York, NY, USA. ACM.
- [Braunhofer and Ricci, 2016] Braunhofer, M. and Ricci, F. (2016). Contextual Information Elicitation in Travel Recommender Systems, pages 579–592. Springer International Publishing, Cham.
- [Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98, pages 43–52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Breese et al., 2013] Breese, J. S., Heckerman, D., and Kadie, C. M. (2013). Empirical analysis of predictive algorithms for collaborative filtering. CoRR, abs/1301.7363.
- [Bridge, 2001] Bridge, D. (2001). Product recommendation systems : A new direction. In Workshop on CBR in Electronic Commerce at The international Conference on Case-Based Reasoning, ICCBR'01, pages 79–86.
- [Brown and Chen, 1997] Brown, P. and Chen, X. (1997). Context-aware applications : from the laboratory to the marketplace. In IEEE, IEEE'97, pages 58–64.
- [Brun et al., 2009] Brun, A., Bonnin, G., and Boyer, A. (2009). History dependent Recommender Systems based on Partial Matching. In Houben, G.-J., McCalla, G., Pianesi, F., and Zancanaro, M., editors, First and Seventeenth International Conference on User Modeling, Adaptation and Personalization - UMAP 2009, volume 5535, pages 343–348, Trento, Italy. Springer Berlin / Heidelberg. The original publication is available at www.springerlink.com.
- [Burke, 2000] Burke, R. (2000). Knowledge-based recommender systems. In ENCYCLOPEDIA OF LIBRARY AND INFORMATION SYSTEMS, page 2000. Marcel Dekker.
- [Burke, 2002] Burke, R. (2002). Hybrid recommender systems : Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) :331–370.
- [Buskirk, 2017] Buskirk, E. V. (2017). Spotify, accuweather reveal how weather affects music listening.
- [Cai et al., 2007] Cai, R., Zhang, C., Wang, C., Zhang, L., and ying Ma, W. (2007). Musicsense : Contextual music recommendation using emotional allocation modeling. ACM Trans. Interact. Intell. Syst.
- [Campos et al., 2014] Campos, P. G., Díez, F., and Cantador, I. (2014). Time-aware recommender systems : a comprehensive survey and analysis of existing evaluation protocols. User Modeling and User-Adapted Interaction, 24(1) :67–119.
- [Candillier et al., 2011] Candillier, L., Chevalier, M., Dudognon, D., and Mothe, J. (2011). Diversity in recommender systems : Bridging the gap between users and systems. In 4th International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies and Services (CENTRIC'2011), Barcelona, Spain.
- [Candillier et al., 2013] Candillier, L., Chevalier, M., Dudognon, D., and Mothe, J. (2013). Multiple similarities for diversity in recommender systems.
- [Candillier et al., 2007] Candillier, L., Meyer, F., and Boullé, M. (2007). Comparing state-of-the-art collaborative filtering systems. In Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM '07, pages 548–562, Berlin, Heidelberg. Springer-Verlag.

-
- [Castagnos, 2008] Castagnos, S. (2008). Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d’interactions sociales au sein de systèmes temps réel de recherche et d’accès à l’information. Theses, Université Nancy II.
- [Castagnos et al., 2013a] Castagnos, S., Brun, A., and Boyer, A. (2013a). Utilité et perception de la diversité dans les systèmes de recommandation. In CORIA 2013 - Conférence en Recherche d’Informations et Applications - 10th French Information Retrieval Conference, Neuchâtel, Suisse, April 3-5, 2013., pages 237–252.
- [Castagnos et al., 2013b] Castagnos, S., Brun, A., and Boyer, A. (2013b). When Diversity Is Needed... But Not Expected! In International Conference on Advances in Information Mining and Management, pages 44–50, Lisbon, Portugal. IARIA XPS Press.
- [Castagnos et al., 2014] Castagnos, S., Brun, A., and Boyer, A. (2014). La diversité : entre besoin et méfiance dans les systèmes de recommandation. In Information Interaction Intelligence, I3 ’14.
- [Castagnos et al., 2009] Castagnos, S., Jones, N., and Pu, P. (2009). Recommenders’ influence on buyers’ decision process. In Proceedings of the Third ACM Conference on Recommender Systems, RecSys ’09, pages 361–364, New York, NY, USA. ACM.
- [Castagnos et al., 2010] Castagnos, S., Jones, N., and Pu, P. (2010). Eye-tracking product recommenders’ usage. In Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys ’10, pages 29–36, New York, NY, USA. ACM.
- [Castagnos et al., 2015] Castagnos, S., L ’huillier, A., and Boyer, A. (2015). Toward a robust diversity-based model to detect changes of context. In 27th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2015), Vietri sul Mare, Italy.
- [Celma, 2008] Celma, Òscar. (2008). Music Recommendation and Discovery in the Long Tail. PhD thesis, Universitat Pompeu Fabra, Barcelona.
- [Cena et al., 2006] Cena, F., Console, L., Gena, C., Goy, A., Levi, G., Modeo, S., and Torre, I. (2006). Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. AI Commun., 19(4) :369–384.
- [Chee et al., 2001] Chee, S. H. S., Han, J., and Wang, K. (2001). Rectree : An efficient collaborative filtering method. In Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery, DaWaK ’01, pages 141–151, London, UK, UK. Springer-Verlag.
- [Chen and Chen, 2014a] Chen, G. and Chen, L. (2014a). Recommendation based on contextual opinions. In Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., and Houben, G.-J., editors, User Modeling, Adaptation, and Personalization, volume 8538 of UMAP ’14, pages 61–73. Springer International Publishing.
- [Chen and Chen, 2014b] Chen, G. and Chen, L. (2014b). Recommendation Based on Contextual Opinions, pages 61–73. Springer International Publishing, Cham.
- [Chen et al., 2013] Chen, L., de Gemmis, M., Felfernig, A., Lops, P., Ricci, F., and Semeraro, G. (2013). Human decision making and recommender systems. ACM Trans. Interact. Intell. Syst., 3(3) :17 :1–17 :7.
- [Chen and Pu, 2012] Chen, L. and Pu, P. (2012). Critiquing-based recommenders : survey and emerging trends. User Model. User-Adapt. Interact., 22(1-2) :125–150.

- [Chen and Joachims, 2016] Chen, S. and Joachims, T. (2016). Predicting matchups and preferences in context. In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pages 775–784, New York, NY, USA. ACM.
- [Chen et al., 2016] Chen, Y.-S., Cheng, C.-H., Chen, D.-R., and Lai, C.-H. (2016). A mood- and situation-based model for developing intuitive pop music recommendation systems. Expert Sys : J. Knowl. Eng., 33(1) :77–91.
- [Chung et al., 2013] Chung, Y., Jung, H.-W., Kim, J., and Lee, J.-H. (2013). Personalized expert-based recommender system : Training c-svm for personalized expert identification. In Perner, P., editor, Machine Learning and Data Mining in Pattern Recognition, pages 434–441, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Clarke et al., 2008] Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 659–666, New York, NY, USA. ACM.
- [Cleverdon and Kean, 1968] Cleverdon, C. and Kean, M. (1968). Factors determining the performance of indexing systems. Aslib Cranfield Research Project, Cranfield, England.
- [Çoba and Zanker, 2017] Çoba, L. and Zanker, M. (2017). Replication and reproduction in recommender systems research - evidence from a case-study with the rrecsys library. In Benferhat, S., Tabia, K., and Ali, M., editors, Advances in Artificial Intelligence : From Theory to Practice, pages 305–314, Cham. Springer International Publishing.
- [Codina et al., 2016] Codina, V., Ricci, F., and Ceccaroni, L. (2016). Distributional semantic pre-filtering in context-aware recommender systems. User Modeling and User-Adapted Interaction, 26(1) :1–32.
- [Cosley et al., 2003] Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J. (2003). Is seeing believing? : How recommender system interfaces affect users' opinions. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03, pages 585–592, New York, NY, USA. ACM.
- [Cramer et al., 2008] Cramer, H., Evers, V., Ramlal, S., van Someren, M., Rutledge, L., Stash, N., Aroyo, L., and Wielinga, B. (2008). The effects of transparency on trust in and acceptance of a content-based art recommender. User Modeling and User-Adapted Interaction, 18(5) :455.
- [Cranor, 2005] Cranor, L. F. (2005). Hey, that's personal! In Ardissono, L., Bruna, P., and Mitrovic, A., editors, User Modeling 2005, volume 3538 of Lecture Notes in Computer Science, pages 4–4. Springer Berlin Heidelberg.
- [Dascalu et al., 2015] Dascalu, M.-I., Bodea, C.-N., Moldoveanu, A., Mohora, A., Lytras, M., and de Pablos, P. O. (2015). A recommender agent based on learning styles for better virtual collaborative learning experiences. Computers in Human Behavior, 45 :243 – 253.
- [Davis, 1989] Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. In MIS Quarterly 13, ICCBR'01, pages 319–340.
- [de Borba et al., 2017] de Borba, E. J., Gasparini, I., and Lichtnow, D. (2017). Time-aware recommender systems : A systematic mapping. In Kurosu, M., editor, Human-Computer Interaction. Interaction Contexts, pages 464–479, Cham. Springer International Publishing.
- [de Gemmis et al., 2015] de Gemmis, M., Lops, P., Musto, C., Narducci, F., and Semeraro, G. (2015). Semantics-Aware Content-Based Recommender Systems, pages 119–159. Springer US, Boston, MA.

-
- [Devendorf and Höllerer, 2012] Devendorf, L. and Höllerer, T. (2012). Topiclens : An interactive recommender system based on topical and social connections.
- [Dey et al., 2001] Dey, A. K., Abowd, G. D., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum.-Comput. Interact., 16(2) :97–166.
- [Di Noia et al., 2014] Di Noia, T., Ostuni, V. C., Rosati, J., Tomeo, P., and Di Sciascio, E. (2014). An analysis of users’ propensity toward diversity in recommendations. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14, pages 285–288, New York, NY, USA. ACM.
- [Dias and Fonseca, 2013] Dias, R. and Fonseca, M. J. (2013). Improving music recommendation in session-based collaborative filtering by using temporal context. In Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, ICTAI ’13, pages 783–788, Washington, DC, USA. IEEE Computer Society.
- [Diefenbach et al., 2016] Diefenbach, D., Lherisson, P.-R., Muhlenbach, F., and Maret, P. (2016). Computing the Semantic Relatedness of Music Genres using Semantic Web Data. In Semantics 2016, Leipzig, Germany.
- [Dourish, 2004] Dourish, P. (2004). What we talk about when we talk about context. Personal Ubiquitous Comput., 8(1) :19–30.
- [Ekstrand et al., 2014] Ekstrand, M. D., Harper, F. M., Willemsen, M. C., and Konstan, J. A. (2014). User perception of differences in recommender algorithms. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14, pages 161–168, New York, NY, USA. ACM.
- [Ekstrand et al., 2015] Ekstrand, M. D., Kluver, D., Harper, F. M., and Konstan, J. A. (2015). Letting users choose recommender algorithms : An experimental study. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys ’15, pages 11–18, New York, NY, USA. ACM.
- [Ekstrand et al., 2011] Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. Found. Trends Hum.-Comput. Interact., 4(2) :81–173.
- [Esslimani et al., 2009] Esslimani, I., Brun, A., and Boyer, A. (2009). A collaborative filtering approach combining clustering and navigational based correlations. In Filipe, J. and Cordeiro, J., editors, WEBIST, pages 364–369. INSTICC Press.
- [Fernández-Tobías et al., 2013] Fernández-Tobías, I., Campos, P. G., Cantador, I., and Díez, F. (2013). A Contextual Modeling Approach for Model-Based Recommender Systems, pages 42–51. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Fischer, 2001] Fischer, G. (2001). User modeling in human–computer interaction. User Modeling and User-Adapted Interaction, 11(1) :65–86.
- [Fleder and Hosanagar, 2007] Fleder, D. M. and Hosanagar, K. (2007). Recommender systems and their impact on sales diversity. In Proceedings of the 8th ACM Conference on Electronic Commerce, EC ’07, pages 192–199, New York, NY, USA. ACM.
- [Foltz and Dumais, 1992] Foltz, P. W. and Dumais, S. T. (1992). Personalized information delivery : An analysis of information filtering methods. Commun. ACM, 35(12) :51–60.
- [Foulonneau et al., 2014] Foulonneau, M., Groues, V., and Naudet, Y. (2014). Recommandeurs et diversité : Exploitation de la longue traîne et diversité des listes de recommandations. In Chartron, G., Saleh, I., and Kembellec, G., editors, Les systèmes de recommandation, chapter 4. Hermès, <http://www.editions-hermes.fr/>.

- [Gabrilovich and Markovitch, 2007] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI'07, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Gadanho and Lhuillier, 2007] Gadanho, S. C. and Lhuillier, N. (2007). Addressing uncertainty in implicit preferences. In Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys '07, pages 97–104, New York, NY, USA. ACM.
- [Gai and Qiang, 2016] Gai, L. and Qiang, C. (2016). Exploiting Explicit and Implicit Feedback for Personalized Ranking, pages 769–803. Hindawi.
- [Gayo-Avello, 2009] Gayo-Avello, D. (2009). A survey on session detection methods in query logs and a proposal for future evaluation. Inf. Sci., 179(12) :1822–1843.
- [Ghazanfar and Prügel-Bennett, 2014] Ghazanfar, M. A. and Prügel-Bennett, A. (2014). Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. Expert Systems with Applications, 41(7) :3261 – 3275.
- [Gillhofer and Schedl, 2015] Gillhofer, M. and Schedl, M. (2015). Iron Maiden While Jogging, Debussy for Dinner?, pages 380–391. Springer International Publishing, Cham.
- [Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Commun. ACM, 35(12) :61–70.
- [Gope and Jain, 2017] Gope, J. and Jain, S. K. (2017). A survey on solving cold start problem in recommender systems. In 2017 International Conference on Computing, Communication and Automation (ICCCA), pages 133–138.
- [Gras et al., 2016] Gras, B., Brun, A., and Boyer, A. (2016). Identifying grey sheep users in collaborative filtering : A distribution-based technique. In Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP '16, pages 17–26, New York, NY, USA. ACM.
- [GroupLens, 2014] GroupLens (Retrieved May 26, 2014). Movielens dataset.
- [Grčar et al., 2006] Grčar, M., Mladenič, D., Fortuna, B., and Grobelnik, M. (2006). Data sparsity issues in the collaborative filtering framework. In Proceedings of the 7th International Conference on Knowledge Discovery on the Web : Advances in Web Mining and Web Usage Analysis, WebKDD'05, pages 58–76, Berlin, Heidelberg. Springer-Verlag.
- [Gunawardana and Shani, 2015] Gunawardana, A. and Shani, G. (2015). Evaluating Recommender Systems, pages 265–308. Springer US, Boston, MA.
- [Haddad et al., 2015] Haddad, M. R., Zghal, H. B., Ziou, D., and Ghézala, H. B. (2015). Un modèle de recommandation contextuel pour la prédiction des intérêts des consommateurs sur le web. In IC 2015 : 26es Journées francophones d'Ingénierie des Connaissances (Proceedings of the 26th French Knowledge Engineering Conference), Rennes, France, July 1-3, 2015.
- [Han et al., 2010] Han, B.-J., Rho, S., Jun, S., and Hwang, E. (2010). Music emotion classification and context-based music recommendation. Multimedia Tools Appl., 47(3) :433–460.
- [Hariri et al., 2012] Hariri, N., Mobasher, B., and Burke, R. (2012). Context-aware music recommendation based on latent topic sequential patterns. In Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pages 131–138, New York, NY, USA. ACM.
- [Hariri et al., 2013] Hariri, N., Mobasher, B., and Burke, R. (2013). Query-driven context aware recommendation. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 9–16, New York, NY, USA. ACM.

-
- [Hariri et al., 2014] Hariri, N., Mobasher, B., and Burke, R. (2014). Context adaptation in interactive recommender systems. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 41–48, New York, NY, USA. ACM.
- [Hariri et al., 2015] Hariri, N., Mobasher, B., and Burke, R. (2015). Adapting to user preference changes in interactive recommendation. In Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, pages 4268–4274. AAAI Press.
- [Harper et al., 2015] Harper, F. M., Xu, F., Kaur, H., Condiff, K., Chang, S., and Terveen, L. (2015). Putting users in control of their recommendations. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 3–10, New York, NY, USA. ACM.
- [Hasan et al., 2014] Hasan, M., Kashyap, A., Hristidis, V., and Tsotras, V. (2014). User effort minimization through adaptive diversification. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, pages 203–212, New York, NY, USA. ACM.
- [Haydar et al., 2012] Haydar, C., Boyer, A., and Roussanaly, A. (2012). Hybridising collaborative filtering and trust-aware recommender systems. In 8th International Conference on Web Information Systems and Technologies -WEBIST'2012, Porto, Portugal. INSTICC.
- [Haydar et al., 2015] Haydar, C., Boyer, A., and Roussanaly, A. (2015). Time-aware trust model for recommender systems. International Symposium on Web AlGorithms. Poster.
- [Herlocker et al., 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99, pages 230–237, New York, NY, USA. ACM.
- [Herlocker et al., 2000] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW '00, pages 241–250, New York, NY, USA. ACM.
- [Herlocker et al., 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst., 22(1) :5–53.
- [Hidasi et al., 2015] Hidasi, B., Karatzoglou, A., Baltrunas, L., and Tikk, D. (2015). Session-based recommendations with recurrent neural networks. CoRR, abs/1511.06939.
- [Hidasi and Tikk, 2012] Hidasi, B. and Tikk, D. (2012). Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback, pages 67–82. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Hofmann, 2004] Hofmann, T. (2004). Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst., 22(1) :89–115.
- [Hosseinzadeh Aghdam et al., 2015] Hosseinzadeh Aghdam, M., Hariri, N., Mobasher, B., and Burke, R. (2015). Adapting recommendations to contextual changes using hierarchical hidden markov models. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 241–244, New York, NY, USA. ACM.
- [Hu and Pu, 2011] Hu, R. and Pu, P. (2011). Enhancing recommendation diversity with organization interfaces. In Proceedings of the 16th International Conference on Intelligent User Interfaces, IUI '11, pages 347–350, New York, NY, USA. ACM.
- [Hu et al., 2008] Hu, Y., Koren, Y., and Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM '08, pages 263–272, Washington, DC, USA. IEEE Computer Society.

- [Häubl and Murray, 2003] Häubl, G. and Murray, K. B. (2003). Preference construction and persistence in digital marketplaces : The role of electronic recommendation agents. Journal of Consumer Psychology, 13(1) :75 – 91. Consumers in Cyberspace.
- [Iaquinta et al., 2008] Iaquinta, L., d. Gemmis, M., Lops, P., Semeraro, G., Filannino, M., and Molino, P. (2008). Introducing serendipity in a content-based recommender system. In 2008 Eighth International Conference on Hybrid Intelligent Systems, pages 168–173.
- [J. McCarthy, 1969] J. McCarthy, P. H. (1969). Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence, 4 :463–502.
- [Jahrer et al., 2010] Jahrer, M., Tscher, A., and Legenstein, R. (2010). Combining predictions for accurate recommender systems. In ACM International Conference on Knowledge Discovery and Data Mining, SIGKDD '10, pages 693–702, Washington, DC, USA.
- [Jannach and Adomavicius, 2016] Jannach, D. and Adomavicius, G. (2016). Recommendations with a purpose. In Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, pages 7–10, New York, NY, USA. ACM.
- [Jannach et al., 2014] Jannach, D., Kamehkhosh, I., and Bonnin, G. (2014). Analyzing the characteristics of shared playlists for music recommendation. In RSWeb@RecSys.
- [Jannach et al., 2013] Jannach, D., Lerche, L., Gedikli, F., and Bonnin, G. (2013). What Recommenders Recommend – An Analysis of Accuracy, Popularity, and Sales Diversity Effects, pages 25–37. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Jannach et al., 2015] Jannach, D., Lerche, L., and Kamehkhosh, I. (2015). Beyond "hitting the hits" : Generating coherent music playlist continuations with the right tracks. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 187–194, New York, NY, USA. ACM.
- [Jannach et al., 2017] Jannach, D., Naveed, S., and Jugovac, M. (2017). User control in recommender systems : Overview and interaction challenges. In Bridge, D. and Stuckenschmidt, H., editors, E-Commerce and Web Technologies, pages 21–33, Cham. Springer International Publishing.
- [Jannach et al., 2010] Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). Recommender Systems : An Introduction. Cambridge University Press, New York, NY, USA, 1st edition.
- [Javari and Jalili, 2015] Javari, A. and Jalili, M. (2015). A probabilistic model to resolve diversity-accuracy challenge of recommendation systems. CoRR, abs/1501.01996.
- [Jawaheer et al., 2010] Jawaheer, G., Szomszor, M., and Kostkova, P. (2010). Comparison of implicit and explicit feedback from an online music recommendation service. In Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems, HetRec '10, pages 47–51, New York, NY, USA. ACM.
- [Jawaheer et al., 2014] Jawaheer, G., Weller, P., and Kostkova, P. (2014). Modeling user preferences in recommender systems : A classification framework for explicit and implicit user feedback. ACM Trans. Interact. Intell. Syst., 4(2) :8 :1–8 :26.
- [Jiang and He, 2016] Jiang, C. and He, Y. (2016). Smart-dj : Context-aware personalization for music recommendation on smartphones. In 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), pages 133–140.
- [Jones, 2010] Jones, N. (2010). User Perceived Qualities and Acceptance of Recommender Systems. PhD thesis, IC, Lausanne.

-
- [Jones et al., 2011a] Jones, N., Brun, A., and Boyer, A. (2011a). Initial perspectives from preferences expressed through comparisons. In HCI International 2011 - Posters' Extended Abstracts - International Conference, HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part I, pages 33–37.
- [Jones et al., 2011b] Jones, N., Brun, A., Boyer, A., and Hamad, A. (2011b). An Exploratory Work in Using Comparisons Instead of Ratings, pages 184–195. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Jones and Pu, 2007] Jones, N. and Pu, P. (2007). User technology adoption issues in recommender systems. In Proceedings of the 2007 Networking and Electronic Commerce Research Conference, page 379–394, Riva del Garda, Italy.
- [Jr., 2013] Jr., J. F. (2013). The occupational conditioning of preferential auditory tempo : a contribution toward an empirical theory of aesthetics. In The journal of Social Psychology, pages 121–129.
- [Juniarta et al., 2018] Juniarta, N., Codocedo, V., Couceiro, M., and Napoli, A. (2018). Bi-clustering based on FCA and partition pattern structures for recommendation systems. In Proceedings of the 6th International Workshop "What can FCA do for Artificial Intelligence" ? co-located with International Joint Conference on Artificial Intelligence and European Conference on Artificial Intelligence (IJCAI/ECAI 2018), Stockholm, Sweden, July 13, 2018., pages 105–116.
- [Kamehkhosh et al., 2016] Kamehkhosh, I., Jannach, D., and Lerche, L. (2016). Personalized next-track music recommendation with multi-dimensional long-term preference signals. In UMAP.
- [Kaminskas and Ricci, 2012] Kaminskas, M. and Ricci, F. (2012). Contextual music information retrieval and recommendation : State of the art and challenges. Computer Science Review, 6(2–3) :89 – 119.
- [Kaminskas et al., 2013] Kaminskas, M., Ricci, F., and Schedl, M. (2013). Location-aware music recommendation using auto-tagging and hybrid matching. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 17–24, New York, NY, USA. ACM.
- [Kapoor et al., 2015] Kapoor, K., Kumar, V., Terveen, L., Konstan, J. A., and Schrater, P. (2015). "i like to explore sometimes" : Adapting to dynamic user novelty preferences". In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 19–26, New York, NY, USA. ACM.
- [Karatzoglou et al., 2010] Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation : N-dimensional tensor factorization for context-aware collaborative filtering. In Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, pages 79–86, New York, NY, USA. ACM.
- [Khrouf and Troncy, 2013] Khrouf, H. and Troncy, R. (2013). Hybrid event recommendation using linked data and user diversity. In Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13, pages 185–192, New York, NY, USA. ACM.
- [Killick and Eckley, 2014] Killick, R. and Eckley, I. (2014). changepoint : An r package for changepoint analysis. Journal of Statistical Software, Articles, 58(3) :1–19.
- [Knees and Schedl, 2013] Knees, P. and Schedl, M. (2013). A survey of music similarity and recommendation from music context data. ACM Trans. Multimedia Comput. Commun. Appl., 10(1) :2 :1–2 :21.

- [Knijnenburg and Kobsa, 2013] Knijnenburg, B. P. and Kobsa, A. (2013). Making decisions about privacy : Information disclosure in context-aware recommender systems. ACM Trans. Interact. Intell. Syst., 3(3) :20 :1–20 :23.
- [Knijnenburg et al., 2013] Knijnenburg, B. P., Kobsa, A., and Jin, H. (2013). Dimensionality of information disclosure behavior. International Journal of Human-Computer Studies, 71(12) :1144 – 1162.
- [Konstan and Riedl, 2012] Konstan, J. A. and Riedl, J. (2012). Recommended for you. IEEE Spectrum, 49(10) :54–61.
- [Koren, 2010] Koren, Y. (2010). Factor in the neighbors : Scalable and accurate collaborative filtering. ACM Trans. Knowl. Discov. Data, 4(1) :1 :1–1 :24.
- [Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8) :30–37.
- [Kotkov et al., 2016] Kotkov, D., Wang, S., and Veijalainen, J. (2016). A survey of serendipity in recommender systems. Knowledge-Based Systems, 111 :180 – 192.
- [Kramár and Bieliková, 2014] Kramár, T. and Bieliková, M. (2014). Context of seasonality in web search. In de Rijke, M., Kenter, T., de Vries, A., Zhai, C., de Jong, F., Radinsky, K., and Hofmann, K., editors, Advances in Information Retrieval, volume 8416 of Lecture Notes in Computer Science, pages 644–649. Springer International Publishing.
- [Kunaver and Požrl, 2017] Kunaver, M. and Požrl, T. (2017). Diversity in recommender systems – a survey. Knowledge-Based Systems, 123 :154 – 162.
- [Kwon and Kim, 2010] Kwon, J. and Kim, S. (2010). Friend recommendation method using physical and social context.
- [Lacerda, 2017] Lacerda, A. (2017). Multi-objective ranked bandits for recommender systems. Neurocomputing, 246 :12 – 24. Brazilian Conference on Intelligent Systems 2015.
- [Lathia, 2010] Lathia, N. (2010). Evaluating Collaborative Filtering Over Time. PhD thesis, University of London, Department of Computer Science, University College London.
- [Lathia et al., 2010] Lathia, N., Hailes, S., Capra, L., and Amatriain, X. (2010). Temporal diversity in recommender systems. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, pages 210–217, New York, NY, USA. ACM.
- [Lawrence, 2000] Lawrence, S. (2000). Context in web search. IEEE Data Engineering Bulletin, 23 :25–32.
- [Leake and Scherle, 2001] Leake, D. and Scherle, R. (2001). Towards context-based search engine selection. In Proceedings of the 6th International conference on Intelligent user interfaces, pages 109–112, New York, NY, USA. ACM.
- [Lee et al., 2010] Lee, D., Park, S., Kahng, M., Lee, S., and Lee, S.-g. (2010). Exploiting contextual information from event logs for personalized recommendation. In Lee, R., editor, Computer and Information Science 2010, volume 317 of Studies in Computational Intelligence, pages 121–139. Springer Berlin Heidelberg.
- [Lee and Price, 2016] Lee, J. H. and Price, R. (2016). User experience with commercial music services : An empirical exploration. Journal of the Association for Information Science and Technology, 67(4) :800–811.
- [Lerche et al., 2016] Lerche, L., Jannach, D., and Ludewig, M. (2016). On the value of reminders within e-commerce recommendations. In Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP '16, pages 27–35, New York, NY, USA. ACM.

-
- [L’Huillier et al., 2014] L’Huillier, A., Castagnos, S., and Boyer, A. (2014). Understanding Usages by Modeling Diversity over Time. volume 1181 of UMAP 2014 Extended Proceedings, Aalborg, Denmark.
- [L’Huillier et al., 2016a] L’Huillier, A., Castagnos, S., and Boyer, A. (2016a). Modéliser la diversité au cours du temps pour détecter le contexte dans un service de musique en ligne. Revue des Sciences et Technologies de l’Information.
- [L’Huillier et al., 2016b] L’Huillier, A., Castagnos, S., and Boyer, A. (2016b). The new challenges when modeling context through diversity over time in recommender systems. In Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, UMAP ’16, pages 341–344, New York, NY, USA. ACM.
- [L’Huillier et al., 2017] L’Huillier, A., Castagnos, S., and Boyer, A. (2017). Are item attributes a good alternative to context elicitation in recommender systems? In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, UMAP ’17, pages 371–372, New York, NY, USA. ACM.
- [Li et al., 2010] Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th International Conference on World Wide Web, WWW ’10, pages 661–670, New York, NY, USA. ACM.
- [Li et al., 2015] Li, X., Wang, H., and Yan, X. (2015). Accurate Recommendation Based on Opinion Mining, pages 399–408. Springer International Publishing, Cham.
- [Liang et al., 2014] Liang, S., Ren, Z., and de Rijke, M. (2014). Personalized search result diversification via structured learning. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14, pages 751–760, New York, NY, USA. ACM.
- [Lin et al., 2018] Lin, B., Zampetti, F., Di Penta, M., Bavota, G., Lanza, M., and Oliveto, R. (2018). Sentiment analysis for software engineering : How far can we go? In Proceedings of the 40th International Conference on Software Engineering, ICSE ’18, page 94–104.
- [Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations : Item-to-item collaborative filtering. IEEE Internet Computing, 7(1) :76–80.
- [Liu et al., 2010] Liu, N. N., Xiang, E. W., Zhao, M., and Yang, Q. (2010). Unifying explicit and implicit feedback for collaborative filtering. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM ’10, pages 1445–1448, New York, NY, USA. ACM.
- [Lops et al., 2011] Lops, P., de Gemmis, M., and Semeraro, G. (2011). Content-based Recommender Systems : State of the Art and Trends, pages 73–105. Springer US, Boston, MA.
- [Louëdec et al., 2015] Louëdec, J., Chevalier, M., Garivier, A., and Mothe, J. (Feb. 2015). Algorithmes de bandits pour la recommandation à tirages multiples. Document numérique (Hermès), 18 :59–79.
- [Louëdec et al., 2015] Louëdec, J., Chevalier, M., Garivier, A., and Mothe, J. (2015). Systèmes de recommandations : algorithmes de bandits et évaluation expérimentale. In 47emes Journées de Statistique de la SFdS (JDS 2015), pages pp. 1–8, Lille, France.
- [Ludewig and Jannach, 2018] Ludewig, M. and Jannach, D. (2018). Evaluation of session-based recommendation algorithms. CoRR, abs/1803.09587.

- [Maksai et al., 2015] Maksai, A., Garcin, F., and Faltings, B. (2015). Predicting online performance of news recommender systems through richer evaluation metrics. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 179–186, New York, NY, USA. ACM.
- [Mallipeddi et al., 2011] Mallipeddi, R., Suganthan, P., Pan, Q., and Tasgetiren, M. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing, 11(2) :1679 – 1696. The Impact of Soft Computing for the Progress of Artificial Intelligence.
- [Mantilla and Víctor, 2013] Mantilla, César A., T.-P. and Víctor, Fabregat, R. (2013). Context- and Social-Aware User Profiling for Audiovisual Recommender Systems, pages 569–577. Springer International Publishing, Cham.
- [Marshall et al., 2015] Marshall, T. C., Lefringhausen, K., and Ferenczi, N. (2015). The big five, self-esteem, and narcissism as predictors of the topics people write about in facebook status updates. Personality and Individual Differences, 85 :35 – 40.
- [Massa and Avesani, 2007] Massa, P. and Avesani, P. (2007). Trust-aware recommender systems. In Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys '07, pages 17–24, New York, NY, USA. ACM.
- [McGinty and Smyth, 2003] McGinty, L. and Smyth, B. (2003). On the role of diversity in conversational recommender systems. In Proceedings of the 5th International Conference on Case-based Reasoning : Research and Development, ICCBR'03, pages 276–290, Berlin, Heidelberg. Springer-Verlag.
- [McNee et al., 2006] McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough : How accuracy metrics have hurt recommender systems. In CHI '06 Extended Abstracts on Human Factors in Computing Systems, CHI EA '06, pages 1097–1101, New York, NY, USA. ACM.
- [McSherry, 2002] McSherry, D. (2002). Diversity-conscious retrieval. In Proceedings of the 6th European Conference on Advances in Case-Based Reasoning, pages 219–233, London, UK.
- [Middleton et al., 2009] Middleton, S., Roure, D., and Shadbolt, N. (2009). Ontology-based recommender systems. In Staab, S. and Studer, R., editors, Handbook on Ontologies, International Handbooks on Information Systems, pages 779–796. Springer Berlin Heidelberg.
- [Moling et al., 2012] Moling, O., Baltrunas, L., and Ricci, F. (2012). Optimal radio channel recommendations with explicit and implicit feedback. In Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pages 75–82, New York, NY, USA. ACM.
- [Moore et al., 2013] Moore, J. L., Chen, S., Turnbull, D., and Joachims, T. (2013). Taste over time : The temporal dynamics of user preferences. In de Souza Britto Jr., A., Gouyon, F., and Dixon, S., editors, ISMIR, pages 401–406.
- [Moukas, 1997] Moukas, A. (1997). Amalthea information discovery and filtering using a multi-agent evolving ecosystem. Applied Artificial Intelligence, 11(5) :437–457.
- [Murakami et al., 2008] Murakami, T., Mori, K., and Orihara, R. (2008). Metrics for evaluating the serendipity of recommendation lists. In Proceedings of the 2007 Conference on New Frontiers in Artificial Intelligence, JSAI'07, pages 40–46, Berlin, Heidelberg. Springer-Verlag.
- [Muralidhar et al., 2015] Muralidhar, N., Rangwala, H., and Han, E. H. S. (2015). Recommending temporally relevant news content from implicit feedback data. In 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), pages 689–696.

-
- [Nart and Tasso, 2014] Nart, D. D. and Tasso, C. (2014). A personalized concept-driven recommender system for scientific libraries. Procedia Computer Science, 38 :84 – 91. 10th Italian Research Conference on Digital Libraries, IRCDL 2014.
- [Nguyen et al., 2006] Nguyen, A.-T., Denos, N., and Berrut, C. (2006). Modèle d’espaces de communautés basé sur la théorie des ensembles d’approximation dans un système de filtrage hybride. In COnférence en Recherche Information et Applications (CORIA), pages 303–314, Lyon, France.
- [Nguyen et al., 2014] Nguyen, T. T., Hui, P.-M., Harper, F. M., Terveen, L., and Konstan, J. A. (2014). Exploring the filter bubble : The effect of using recommender systems on content diversity. In Proceedings of the 23rd International Conference on World Wide Web, WWW ’14, pages 677–686, New York, NY, USA. ACM.
- [Nikolaenko et al., 2013] Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., and Boneh, D. (2013). Privacy-preserving matrix factorization. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS ’13, pages 801–812, New York, NY, USA. ACM.
- [Odic et al., 2011] Odic, A., Tkalcic, M., Košir, A., and Tasic, J. F. (2011). A relevant context in a movie recommender system : Users’ opinion vs. statistical detection. In In : Proc. of the 4th Workshop on Context-Aware Recommender Systems (2011).
- [Odic et al., 2013] Odic, A., Tkalcic, M., Tasic, J. F., and Kosir, A. (2013). Predicting and detecting the relevant contextual information in a movie-recommender system. Interacting with Computers, 25(1) :74–90.
- [Oku et al., 2006] Oku, K., Nakajima, J., and Uemura, S. (2006). Context-aware svm for context-dependent information recommendation. In Proceedings of the 7th International Conference on Mobile Data Management, page 109.
- [Ono et al., 2009] Ono, C., Takishima, Y., Motomura, Y., and Asoh, H. (2009). Context-aware preference model based on a study of difference between real and supposed situation data. In Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization : Formerly UM and AH, UMAP ’09, pages 102–113, Berlin, Heidelberg. Springer-Verlag.
- [Onuma et al., 2009] Onuma, K., Tong, H., and Faloutsos, C. (2009). Tangent : A novel, ’surprise me’, recommendation algorithm. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’09, pages 657–666, New York, NY, USA. ACM.
- [Palmisano et al., 2008] Palmisano, C., Tuzhilin, A., and Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. IEEE Trans. on Knowl. and Data Eng., 20(11) :1535–1549.
- [Pálovics et al., 2014] Pálovics, R., Benczúr, A. A., Kocsis, L., Kiss, T., and Frigó, E. (2014). Exploiting temporal influence in online recommendation. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14, pages 273–280, New York, NY, USA. ACM.
- [Panniello and Gorgoglione, 2011] Panniello, U. and Gorgoglione, M. (2011). Context-aware recommender systems : A comparison of three approaches. In DART@ AI* IA.
- [Panniello et al., 2009] Panniello, U., Gorgoglione, M., and Palmisano, C. (2009). Comparing pre-filtering and post-filtering approach in a collaborative contextual recommender system : an application to e-commerce. In International Conference on Electronic Commerce and Web Technologies, pages 348–359. Springer Berlin Heidelberg.

- [Panniello et al., 2014] Panniello, U., Tuzhilin, A., and Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. User Modeling and User-Adapted Interaction, 24(1) :35–65.
- [Paramythis et al., 2010] Paramythis, A., Weibelzahl, S., and Masthoff, J. (2010). Layered evaluation of interactive adaptive systems : Framework and formative methods. User Modeling and User-Adapted Interaction, 20(5) :383–453.
- [Park et al., 2006] Park, H.-S., Yoo, J.-O., and Cho, S.-B. (2006). A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In Proceedings of the Third International Conference on Fuzzy Systems and Knowledge Discovery, FSKD'06, pages 970–979, Berlin, Heidelberg. Springer-Verlag.
- [Park et al., 2011] Park, S. E., Lee, S., and goo Lee, S. (2011). Session-based collaborative filtering for predicting the next song. In Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on, pages 353–358.
- [Park and Tuzhilin, 2008] Park, Y.-J. and Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, pages 11–18, New York, NY, USA. ACM.
- [Pazzani and Billsus, 2007] Pazzani, M. J. and Billsus, D. (2007). Content-Based Recommendation Systems, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Pichl et al., 2016] Pichl, M., Zangerle, E., and Specht, G. (2016). Understanding playlist creation on music streaming platforms. In 2016 IEEE International Symposium on Multimedia (ISM), pages 475–480.
- [Porter, 1997] Porter, M. F. (1997). Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [Pu and Chen., 2007] Pu, P. and Chen., L. (2007). Trust-inspiring explanation interfaces for recommender systems. Knowledge-Based Systems, 20(6) :542 – 556. Special Issue On Intelligent User Interfaces.
- [Pu et al., 2009] Pu, P., Zhou, M., and Castagnos, S. (2009). Critiquing recommenders for public taste products. In Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09, pages 249–252, New York, NY, USA. ACM.
- [Puntheeranurak and Chaiwitooanukool, 2011] Puntheeranurak, S. and Chaiwitooanukool, T. (2011). An item-based collaborative filtering method using item-based hybrid similarity. In 2011 IEEE 2nd International Conference on Software Engineering and Service Science, pages 469–472.
- [Radlinski et al., 2009] Radlinski, F., Bennett, P. N., Carterette, B., and Joachims, T. (2009). Redundancy, diversity and interdependent document relevance. SIGIR Forum, 43(2) :46–52.
- [Radlinski and Dumais, 2006] Radlinski, F. and Dumais, S. (2006). Improving personalized web search using result diversification. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, pages 691–692, New York, NY, USA. ACM.
- [Ravi and Vairavasundaram, 2016] Ravi, L. and Vairavasundaram, S. (2016). A collaborative location based travel recommendation system through enhanced rating prediction for the group of users. Intell. Neuroscience, 2016 :7–.
- [Razali and Geraghty, 2001] Razali, N. M. and Geraghty, J. (2001). Genetic algorithm performance with different selection strategies in solving tsp.

-
- [Reilly et al., 2004] Reilly, J., McCarthy, K., McGinty, L., and Smyth, B. (2004). Dynamic critiquing. In Funk, P. and Calero, P. A. G., editors, Proceedings of the 7th European Conference on Case-Based Reasoning, pages 763–777. Springer-Verlag.
- [Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens : An open architecture for collaborative filtering of netnews. In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW '94, pages 175–186, New York, NY, USA. ACM.
- [Resnick and Varian, 1997] Resnick, P. and Varian, H. R. (1997). Recommender systems. Commun. ACM, 40(3) :56–58.
- [Ribeiro et al., 2014] Ribeiro, M. T., Ziviani, N., Moura, E. S. D., Hata, I., Lacerda, A., and Veloso, A. (2014). Multiobjective pareto-efficient approaches for recommender systems. ACM Trans. Intell. Syst. Technol., 5(4) :53 :1–53 :20.
- [Ricci et al., 2011] Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to Recommender Systems Handbook, pages 1–35. Springer US, Boston, MA.
- [Ricci et al., 2015] Ricci, F., Rokach, L., and Shapira, B. (2015). Recommender Systems : Introduction and Challenges, pages 1–34. Springer US, Boston, MA.
- [Ricci et al., 2010] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). Recommender Systems Handbook. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition.
- [Rodriguez et al., 2012] Rodriguez, M., Posse, C., and Zhang, E. (2012). Multiple objective optimization in recommender systems. In Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12, pages 11–18, New York, NY, USA. ACM.
- [Rosner and Kostek, 2018] Rosner, A. and Kostek, B. (2018). Automatic music genre classification based on musical instrument track separation. Journal of Intelligent Information Systems, 50(2) :363–384.
- [S. Zufryden et al., 1993] S. Zufryden, F., H. Pedrick, J., and Sankaralingam, A. (1993). Zapping and its impact on brand purchase behavior. 33 :58–66.
- [Said, 2010] Said, A. (2010). Identifying and utilizing contextual data in hybrid recommender systems. In Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, pages 365–368, New York, NY, USA. ACM.
- [Said and Bellogín, 2014] Said, A. and Bellogín, A. (2014). Comparative recommender system evaluation : Benchmarking recommendation frameworks. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 129–136, New York, NY, USA. ACM.
- [Said and Bellogín, 2015] Said, A. and Bellogín, A. (2015). Replicable evaluation of recommender systems. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 363–364, New York, NY, USA. ACM.
- [Said et al., 2012] Said, A., Jain, B. J., Kille, B., and Albayrak, S. (2012). Increasing diversity through furthest neighbor-based recommendation. In Proceedings of the WSDM'12 Workshop on Diversity in Document Retrieval (DDR'12).
- [Said et al., 2011] Said, A., Luca, E. W. D., and Albayrak, S. (2011). Inferring contextual user profiles - improving recommender performance. In Proceedings of the 3rd RecSys Workshop on Context-Aware Recommender Systems.
- [Sarwar et al., 2000] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In Proceedings of the 2Nd ACM Conference on Electronic Commerce, EC '00, pages 158–167, New York, NY, USA. ACM.

- [Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th International Conference on World Wide Web, WWW '01, pages 285–295, New York, NY, USA. ACM.
- [Sarwar et al., 2002] Sarwar, B.M. and Karypis, G., Konstan, J., and Riedl, J. (2002). Recommender systems for large-scale e-commerce : Scalable neighborhood formation using clustering. In The 5th Int. Conf. on Computer and Information Technology.
- [Schafer et al., 2002] Schafer, J., Konstan, J., and Riedl, J. (2002). Meta-recommendations systems : user-controlled integration of diverse recommendations. In International Conference on Information And Knowledge Management, pages 43–51.
- [Schafer et al., 1999] Schafer, J. B., Konstan, J., and Riedl, J. (1999). Recommender systems in e-commerce. In Proceedings of the 1st ACM Conference on Electronic Commerce, EC '99, pages 158–166, New York, NY, USA. ACM.
- [Schedl, 2016] Schedl, M. (2016). The lfm-1b dataset for music retrieval and recommendation. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICMR '16, pages 103–110, New York, NY, USA. ACM.
- [Schedl et al., 2015] Schedl, M., Knees, P., McFee, B., Bogdanov, D., and Kaminskas, M. (2015). Music Recommender Systems, pages 453–492. Springer US, Boston, MA.
- [Schedl et al., 2017] Schedl, M., Zamani, H., Chen, C., Deldjoo, Y., and Elahi, M. (2017). Current challenges and visions in music recommender systems research. CoRR, abs/1710.03208.
- [Schein et al., 2002] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02, pages 253–260, New York, NY, USA. ACM.
- [Schickel-Zuber and Faltings, 2006] Schickel-Zuber, V. and Faltings, B. (2006). Overcoming incomplete user models in recommendation systems via an ontology. In Proc. of the 7th Int. Conf. on Knowledge Discovery on the Web, WebKDD'05, pages 39–57, Berlin. Springer.
- [Schubert, 2003] Schubert, E. (2003). Update of the hevner adjective checklist. Perceptual and Motor Skills, 96(3_suppl) :1117–1122.
- [Schwartz, 2005] Schwartz, B. (2005). The Paradox of Choice : Why More Is Less. Harper Perennial.
- [Sezgin et al., 2012] Sezgin, M. C., Günsel, B., and Kurt, G. K. (2012). Perceptual audio features for emotion detection. EURASIP Journal on Audio, Speech, and Music Processing, 2012(1) :16.
- [Shakirova, 2017] Shakirova, E. (2017). Collaborative filtering for music recommender system. In 2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIconRus), pages 548–550.
- [Shani and Gunawardana, 2011] Shani, G. and Gunawardana, A. (2011). Evaluating Recommendation Systems, pages 257–297. Springer US, Boston, MA.
- [Shani et al., 2005] Shani, G., Heckerman, D., and Brafman, R. I. (2005). An mdp-based recommender system. J. Mach. Learn. Res., 6 :1265–1295.
- [Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering : Algorithms for automating “word of mouth”. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '95, pages 210–217, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

-
- [Sharma and Murty, 2011] Sharma, G. and Murty, M. N. (2011). Mining sentiments from songs using latent dirichlet allocation. In Proceedings of the 10th International Conference on Advances in Intelligent Data Analysis X, IDA'11, pages 328–339, Berlin, Heidelberg. Springer-Verlag.
- [Sharma and Ray, 2016] Sharma, R. and Ray, S. (2016). Explanations in recommender systems : An overview. Int. J. Bus. Inf. Syst., 23(2) :248–262.
- [Shen et al., 2015] Shen, J., Brdiczka, O., and Liu, J. (2015). A study of facebook behavior : What does it tell about your neuroticism and extraversion ? Computers in Human Behavior, 45 :32 – 38.
- [Shi and Zhang, 2017] Shi, X. and Zhang, X. (2017). Chance left constraint model for tsp and its gaso algorithm. In Li, X. and Xu, X., editors, Proceedings of the Fourth International Forum on Decision Sciences, pages 387–397, Singapore. Springer Singapore.
- [Sieg et al., 2007] Sieg, A., Mabasher, B., and Burke, R. (2007). Representing context in web search with ontological user profiles. In Proceedings of the 6th International Conference on Modeling and Using Context.
- [Simpson, 2014] Simpson, C. (2014). Amazon will sell you things before you know you want to buy them. The Wire.
- [Smyth and McClave, 2001] Smyth, B. and McClave, P. (2001). Similarity vs. diversity. In Proceedings of the 4th International Conference on Case-Based Reasoning : Case-Based Reasoning Research and Development, ICCBR '01, pages 347–361, London, UK, UK. Springer-Verlag.
- [Sonam and Puneet, 2014] Sonam, K. and Puneet, G. (2014). An efficient solution of travelling salesman problem using genetic algorithm. In International Journal of Advanced Research in Computer Science and Software Engineering - Volume 4, Issue 5.
- [Song et al., 2012] Song, Y., Dixon, S., and Pearce, M. (2012). A survey of music recommendation systems and future perspectives. In The 9th International Symposium on Computer Music Modeling and Retrieval (CMMR).
- [Su and Khoshgoftaar, 2009] Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Adv. in Artif. Intell., 2009 :4 :2–4 :2.
- [Suchman, 1987] Suchman, L. A. (1987). Plans and Situated Actions : The Problem of Human-machine Communication. Cambridge University Press, New York, NY, USA.
- [Tang and Wang, 2015] Tang, Q. and Wang, J. (2015). Privacy-preserving context-aware recommender systems : Analysis and new solutions. In European Symposium on Research in Computer Security, pages 101–119. Springer.
- [Tarus et al., 2017] Tarus, J. K., Niu, Z., and Yousif, A. (2017). A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. Future Generation Computer Systems, 72 :37 – 48.
- [Taylor, 2000] Taylor, W. (2000). Change-point analysis : A powerful new tool for detecting changes.
- [thenextweb.com, 2012] thenextweb.com (2012). Remember netflix's \$1m algorithm contest ? well, here's why it didn't use the winning entry.
- [thenextweb.com, 2017] thenextweb.com (2017). Netflix is changing its star ratings to a thumbs up/down system.

- [Ticha, 2015] Ticha, S. B. (2015). Recommandation personnalisée hybride. (Hybrid personalized recommendation). PhD thesis, University of Lorraine, Nancy, France.
- [Tintarev et al., 2013] Tintarev, N., Dennis, M., and Masthoff, J. (2013). Adapting recommendation diversity to openness to experience : A study of human behaviour. In Carberry, S., Weibelzahl, S., Micarelli, A., and Semeraro, G., editors, User Modeling, Adaptation, and Personalization, pages 190–202, Berlin, Heidelberg. Springer Berlin Heidelberg.
- [Tintarev and Masthoff, 2007] Tintarev, N. and Masthoff, J. (2007). A survey of explanations in recommender systems. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, ICDEW '07, pages 801–810, Washington, DC, USA. IEEE Computer Society.
- [Umanets et al., 2014] Umanets, A., Ferreira, A., and Leite, N. (2014). Guideme – a tourist guide with a recommender system and social interaction. Procedia Technology, 17 :407 – 414. Conference on Electronics, Telecommunications and Computers – CETC 2013.
- [Unger, 2015] Unger, M. (2015). Latent context-aware recommender systems. In Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pages 383–386, New York, NY, USA. ACM.
- [van den Oord et al., 2013] van den Oord, A., Dieleman, S., and Schrauwen, B. (2013). Deep content-based music recommendation. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 26, pages 2643–2651. Curran Associates, Inc.
- [Vargas and Castells, 2011] Vargas, S. and Castells, P. (2011). Rank and relevance in novelty and diversity metrics for recommender systems. In Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, pages 109–116, New York, NY, USA. ACM.
- [Vargas and Castells, 2014] Vargas, S. and Castells, P. (2014). Improving sales diversity by recommending users to items. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 145–152, New York, NY, USA. ACM.
- [Verbert et al., 2012] Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I., and Duval, E. (2012). Context-aware recommender systems for learning : A survey and future challenges. IEEE Trans. Learn. Technol., 5(4) :318–335.
- [Wang et al., 2016] Wang, D., Deng, S., Liu, S., and Xu, G. (2016). Improving music recommendation using distributed representation. In Proceedings of the 25th International Conference Companion on World Wide Web, WWW '16 Companion, pages 125–126, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.
- [Wang et al., 2015] Wang, H., Wang, N., and Yeung, D.-Y. (2015). Collaborative deep learning for recommender systems. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pages 1235–1244, New York, NY, USA. ACM.
- [Webster, 1980] Webster, N. (1980). Webster’s new twentieth century dictionary of the english language.
- [Wei et al., 2012] Wei, S., Ye, N., Zhang, S., Huang, X., and Zhu, J. (2012). Collaborative filtering recommendation algorithm based on item clustering and global similarity. In 2012 Fifth International Conference on Business Intelligence and Financial Engineering, pages 69–72.
- [Woerndl and Schlichter, 2007] Woerndl, W. and Schlichter, J. (2007). Introducing context into recommender systems. In Proceedings of the AAAI workshop on recommender systems in e-commerce, Vancouver, pages 22–23.

-
- [Wu et al., 2015] Wu, H., Yue, K., Liu, X., Pei, Y., and Li, B. (2015). Context-aware recommendation via graph-based contextual modeling and postfiltering. International Journal of Distributed Sensor Networks, 11(8) :613612.
- [Yang and Wu, 2009] Yang, Y. J. and Wu, C. (2009). An attribute-based ant colony system for adaptive learning object recommendation. Expert Systems with Applications, 36(2, Part 2) :3034 – 3047.
- [Zeng et al., 2016] Zeng, J., Li, F., Liu, H., Wen, J., and Hirokawa, S. (2016). A restaurant recommender system based on user preference and location in mobile environment. In 2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), pages 55–60.
- [Zhang and Hurley, 2008] Zhang, M. and Hurley, N. (2008). Avoiding monotony : Improving the diversity of recommendation lists. In Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08, pages 123–130, New York, NY, USA. ACM.
- [Zhang et al., 2017] Zhang, S., Yao, L., and Sun, A. (2017). Deep learning based recommender system : A survey and new perspectives. CoRR, abs/1707.07435.
- [Zheleva et al., 2010] Zheleva, E., Guiver, J., Mendes Rodrigues, E., and Milić-Frayling, N. (2010). Statistical models of music-listening sessions in social media. In Proceedings of the 19th International Conference on World Wide Web, WWW '10, pages 1019–1028, New York, NY, USA. ACM.
- [Zheng, 2016] Zheng, Y. (2016). Tutorial at acm sac 2016 : Context in recommender systems.
- [Zheng et al., 2014a] Zheng, Y., Burke, R., and Mobasher, B. (2014a). Splitting approaches for context-aware recommendation : An empirical study. In Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC '14, pages 274–279, New York, NY, USA. ACM.
- [Zheng et al., 2014b] Zheng, Y., Mobasher, B., and Burke, R. (2014b). Cslim : Contextual slim recommendation algorithms. In Proceedings of the 8th ACM Conference on Recommender Systems, RecSys '14, pages 301–304, New York, NY, USA. ACM.
- [Zheng et al., 2015a] Zheng, Y., Mobasher, B., and Burke, R. (2015a). Integrating Context Similarity with Sparse Linear Recommendation Model, pages 370–376. Springer International Publishing, Cham.
- [Zheng et al., 2015b] Zheng, Y., Mobasher, B., and Burke, R. (2015b). Similarity-based context-aware recommendation. In Wang, J., Cellary, W., Wang, D., Wang, H., Chen, S.-C., Li, T., and Zhang, Y., editors, Web Information Systems Engineering – WISE 2015, pages 431–447, Cham. Springer International Publishing.
- [Zhou et al., 2010a] Zhou, H., Hermans, T., Karandikar, A. V., and Rehg, J. M. (2010a). Movie genre classification via scene categorization. In Proceedings of the 18th ACM International Conference on Multimedia, MM '10, pages 747–750, New York, NY, USA. ACM.
- [Zhou et al., 2010b] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. (2010b). Solving the apparent diversity-accuracy dilemma of recommender systems. Proceedings of the National Academy of Sciences, 107(10) :4511–4515.
- [Zhou and Nadaf, 2017] Zhou, Y. and Nadaf, A. (2017). Embedded collaborative filtering for "cold start" prediction. CoRR, abs/1704.02552.
- [Ziegler et al., 2005] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In Proceedings of the 14th International Conference on World Wide Web, WWW '05, pages 22–32, New York, NY, USA. ACM.

- [Zimmermann et al., 2007] Zimmermann, A., Lorenz, A., and Oppermann, R. (2007). An operational definition of context. In Proceedings of the 6th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT'07, pages 558–571, Berlin, Heidelberg. Springer-Verlag.
- [Zou et al., 2016] Zou, X., Gonzales, M., and Saeedi, S. (2016). A context-aware recommendation system using smartphone sensors. In 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pages 1–6.

Résumé

Les systèmes de recommandation se sont imposés comme étant des outils indispensables face à une quantité de données qui ne cesse chaque jour de croître depuis l'avènement d'Internet. Leur objectif est de proposer aux utilisateurs des items susceptibles de les intéresser sans que ces derniers n'aient besoin d'agir pour les obtenir. Après s'être majoritairement focalisés sur la précision de la prédiction d'intérêt, ces systèmes ont évolué pour prendre en compte d'autres critères dans leur processus de recommandation, tels que les facteurs humains inhérents à la prise de décision, afin d'améliorer la qualité et l'utilité des recommandations. Cependant, la prise en compte de certains facteurs humains tels que la diversité et le contexte demeure critiquable. Alors que le contexte des utilisateurs est inféré sur la base d'informations collectées à l'insu de leur vie privée, la prise en compte de la diversité est quant à elle réduite à une dimension qu'un système se doit de maximiser. Or, certains travaux récents démontrent que la diversité correspond à un besoin évoluant dynamiquement au cours du temps, et dont la proportion à insuffler dans les recommandations est dépendante de la tâche effectuée (*i.e* du contexte). Partant du postulat inverse selon lequel l'analyse de l'évolution de la diversité au cours du temps permet de définir le contexte de l'utilisateur, nous proposons dans ce manuscrit une nouvelle approche de modélisation contextuelle basée sur la diversité. En effet, nous soutenons qu'une variation de diversité remarquable peut être la conséquence d'un changement de contexte et qu'il faut alors adapter la stratégie de recommandation en conséquence. Nous présentons la première approche de la littérature permettant de modéliser en temps réel l'évolution de la diversité, ainsi qu'une nouvelle famille de contextes dits implicites n'exploitant aucune donnée sensible. La possibilité de remplacer les contextes traditionnels (explicites) par les contextes implicites est confirmée de plusieurs manières. Premièrement, nous démontrons sur deux corpus issus d'applications réelles qu'il existe un fort recouvrement entre les changements de contextes explicites et les changements de contextes implicites. Deuxièmement, une étude utilisateur impliquant de nombreux participants nous permet de démontrer l'existence de liens entre les contextes explicites et les caractéristiques des items consultés dans ces derniers. Fort de ces constats et du potentiel offert par nos modèles, nous présentons également plusieurs approches de recommandation et de prise en compte des besoins des utilisateurs.

Mots-clés: Systèmes de recommandation, diversité, contexte, vie privée

Abstract

Recommender Systems (RS) have become essential tools to deal with an endless increasing amount of data available on the Internet. Their goal is to provide items that may interest users before they have to find them by themselves. After being exclusively focused on the precision of users' interests prediction task, RS had to evolve by taking into account other criteria like human factors involved in the decision-making process while computing recommendations, so as to improve their quality and usefulness of recommendations. Nevertheless, the way some human factors, such as context and diversity needs, are managed remains open to criticism. While context-aware recommendations relies on exploiting data that are collected without any consideration for users' privacy, diversity has been coming down to a dimension which has to be maximized. However recent studies demonstrate that diversity corresponds to a need which evolves dynamically over time. In addition, the optimal amount of diversity to provide in the recommendations depends on the on-going task of users (*i.e* their contexts). Thereby, we argue that analyzing the evolution of diversity over time would be a promising way to define a user's context, under the condition that context is now defined by item attributes. Indeed, we support the idea that a sudden variation of diversity can reflect a change of user's context which requires to adapt the recommendation strategy. We present in this manuscript the first approach to model the evolution of diversity over time and a new kind of context, called "implicit contexts", that are respectful of privacy (in opposition to explicit contexts). We confirm the benefits of implicit contexts compared to explicit contexts from several points of view. As a first step, using two large music streaming datasets we demonstrate that explicit and implicit context changes are highly correlated. As a second step, a user study involving many participants allowed us to demonstrate the links between the explicit contexts and the characteristics of the items consulted in the meantime. Based on these observations and the advantages offered by our models, we also present several approaches to provide privacy-preserving context-aware recommendations and to take into account user's needs.

Keywords: Recommender Systems, diversity, context, privacy

