Accès personnalisé à l'information : prise en compte de la dynamique utilisateur

Elie Guàrdia Sebaoun

Accès Personnalisé à l'information : Prise en Compte de la Dynamique Utilisateur

---

Elie Guàrdia Sebaoun

29 Septembre 2017

THÈSE DE DOCTORAT DE
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Elie Guàrdia Sebaoun

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Accès Personnalisé à l'information : Prise en Compte de la Dynamique Utilisateur

soutenue le 29 Septembre 2017

devant le jury composé de :

M. Patrick GALLINARI	Directeur de thèse
M. Vincent GUIGUE	Encadrant de thèse
Mme. Anne BOYER	Rapporteur
M. Emmanuel VIENNET	Rapporteur
M. Bernd AMANN	Examinateur
M. Nicolas USUNIER	Examinateur

Résumé

Au cours des deux dernières décennies, la quantité d'information disponible sur Internet a explosé, aussi bien au niveau du nombre de sites qu'au sein des sites en eux mêmes. Pour éviter de se noyer et accéder plus facilement aux informations qu'il désire, l'utilisateur a deux outils à sa disposition, les moteurs de recherche et les systèmes de recommandation. Si ces deux outils servent un même but, leurs philosophies respectives sont diamétralement opposées : d'un côté le moteur de recherche fournit des items correspondants à une requête de l'utilisateur (*i.e.* une série de mots-clefs), de l'autre le système de recommandation est proactif et utilise une représentation de l'utilisateur pour lui proposer un ensemble d'item. Nous assistons aujourd'hui à une convergence de ces deux visions. Par exemple, Google [1] utilise les clics des utilisateurs comme une forme de *feedback* pour établir des profils servant à affiner les résultats de son moteur de recherche. Les stratégies collaboratives présentent donc un double enjeu : améliorer la pertinence des réponses pour des requêtes générales et construire des profils utilisateurs pour aller vers des réponses personnalisées.

L'enjeu majeur de cette thèse réside dans l'amélioration de l'adéquation entre l'information retournée et les attentes des utilisateurs à l'aide de profils riches et efficaces. Il s'agit donc d'exploiter au maximum les retours utilisateur (qu'ils soient donnés sous la forme de clics, de notes ou encore d'avis écrits) et le contexte. En parallèle la forte croissance des appareils nomades (smartphones, tablettes) et par conséquent de l'informatique ubiquitaire nous oblige à repenser le rôle des systèmes d'accès à l'information. C'est pourquoi nous ne nous sommes pas seulement intéressés à la performance à proprement parler mais aussi à l'accompagnement de l'utilisateur dans son accès à l'information.

Durant ces travaux de thèse, nous avons choisi d'exploiter les textes écrit par les utilisateurs pour affiner leurs profils et contextualiser la recommandation. À cette fin, nous avons utilisé les avis postés sur les sites spécialisés (IMDb, RateBeer, BeerAdvocate) et les boutiques en ligne (Amazon) ainsi que les messages postés sur Twitter [2]. Dans un second temps, nous nous sommes intéressés aux problématiques

1. http ://google.com
2. http ://www.twitter.com

de modélisation de la dynamique des utilisateurs. En plus d'aider à l'amélioration des performances du système, elle permet d'apporter une forme d'explication quant aux items proposés. En effet, le système devient prédictif au sens où l'on estime le prochain item que l'utilisateur va considérer et la réponse du système prend la forme suivante : *Je pense que tu es intéressé par XXX, cependant il y a de fortes chances que ce produit te déplaise, tu devrais plutôt considérer le produit YYY*. Ainsi, nous proposons d'accompagner l'utilisateur dans son accès à l'information au lieu de contraindre le à un ensemble d'items que le système juge pertinents.

Remerciements

Dans le désordre :

Je tiens à remercier tout particulièrement Patrick Gallinari, mon directeur de thèse, sans qui rien n'aurait été possible. Je remercie aussi mon encadrant, Vincent Guigue pour son accompagnement et ses conseils au cours de ces quatre années de travail. Je les remercie tous les deux pour tous leurs retours et leur patience durant cette année de rédaction à distance.

Je remercie ma famille qui me supporte depuis déjà plus de 30 ans et ma compagne qui a su s'adapter, allant jusqu'à mettre 600km entre nous deux pour me laisser la liberté de rédiger.

Je remercie mes amis, d'ici et d'ailleurs, ainsi que les bars qui nous ont accueillis si souvent.

Je remercie l'équipe MLIA, les permanents autant que mes compagnon(ne)s thésards. Tout particulièrement les habitants du bureau 534 pour l'atmosphère de travail et la bonne ambiance au quotidien. J'ajouterais une mention spéciale pour Clément Calauzènes, pour son soutien, Nicolas Baskiotis et Aymeric Tonnelier, pour les clopes que j'ai pu leur taxer, et Ludovic Denoyer pour celles qu'il a pu me taxer.

Je remercie aussi l'ensemble du personnel administratif et technique du laboratoire, sans qui aucune recherche ne pourrait avoir lieu.

Table des matières

1 Introduction

> *People assume that time is a strict progression of cause to effect, but actually from a non-linear, non-subjective viewpoint - it's more like a big ball of wibbly wobbly... time-y wimey... stuff.*
>
> **— Steven Moffat**

1.1 Vision Générale

Au cours des deux dernières décennies, la quantité d'information disponible sur Internet a explosé, aussi bien au niveau du nombre de sites qu'au sein des sites en eux mêmes. Pour éviter de se noyer et accéder plus facilement aux informations qu'il désire, l'utilisateur a deux outils à sa disposition, les moteurs de recherche et les systèmes de recommandation.

Si ces deux outils servent un même but, leurs philosophies respectives sont diamétralement opposées. D'un côté le moteur de cherche fournit des items correspondants à une requête de l'utilisateur (*i.e.* une série de mots-clefs), de l'autre le système de recommandation est proactif et utilise une représentation de l'utilisateur pour lui proposer un ensemble d'item. Cette représentation est basée sur des données explicites (*e.g.* âge, sexe) et des données de *feedback* (*e.g.* étoiles, notes, clicks).

Nous assistons aujourd'hui à une convergence de ces deux visions. Par exemple, Google [1] utilise les clics des utilisateurs comme une forme de *feedback* pour établir des profils servant à affiner les résultats de son moteur de recherche. Les *feedbacks* récurrents deviennent même des critères d'indexation du type : *toute personne faisant une recherche "req" attend la réponse "ans"*. C'est la capacité de Google à utiliser ces *feedbacks* lui a permis au fil du temps d'asseoir encore un peu plus sa position de leader. Les stratégies collaboratives présentent donc un double enjeu : améliorer la pertinence des réponses pour des requêtes générales et construire des profils utilisateurs pour aller vers des réponses personnalisées

1. http ://google.com

Ici, nous appellerons *profil utilisateur* l'ensemble des informations qui nous permettent de définir ses goûts et préférences, que ces informations soient des interactions (*e.g.* clicks, likes, étoiles) ou des formulaires explicites. En parallèle, nous utiliserons le terme *contexte* pour décrire une définition du monde au moment de l'accès à l'information. L'explosion de l'utilisation de l'internet nomade, a multiplié les exemples d'exploitation du contexte (*e.g.* géolocalisation, date, météo). Ces deux définitions nous permettent de définir le cadre de cette thèse : l'apprentissage de profils pertinents et la prise en compte du contexte dans les stratégies de classification personnalisées.

Durant ces travaux de thèse, nous avons choisi d'exploiter les textes écrits par les utilisateurs pour affiner leurs profils. À cette fin, nous avons utilisé les avis postés sur les sites spécialisés (IMDb, RateBeer, BeerAdvocate) et les boutiques en ligne (Amazon). Ces textes, disponibles en grand volume et pour de nombreux d'utilisateurs, sont une excellente source d'informations diverses, si l'utilisateur y définit explicitement ses goûts et préférences (il n'y dit pas seulement s'il aime un item, mais aussi pourquoi), le style d'écriture et le lexique utilisés permettent aussi de le rapprocher d'autres utilisateurs qui lui ressemblent.

Pour ce qui est de la contextualisation, nous avons une fois de plus considéré l'utilisation de données textuelles. Nous avons d'abord considéré la possibilité d'extraire les émotions des textes pour caractériser l'humeur de l'utilisateur au moment de la recommandation. Nous nous sommes rabattus sur la modélisation de la dynamique des utilisateurs. Nous avons aussi essayé de caractériser le contexte d'une situation à travers les réseaux sociaux (Twitter) pour répondre à des tâches annexes, ici la prédiction de résultats au Box-Office.

La modélisation de la dynamique des utilisateurs, en plus d'aider à l'amélioration des performances du système, permet d'apporter une forme d'explicativité quant aux items proposés. En effet, le système devient prédictif au sens où l'on estime le prochain item que l'utilisateur va considérer et la réponse du système prend la forme suivante : *Je pense que tu es intéressé par XXX, cependant il y a de fortes chances que ce produit te déplaise, tu devrais plutôt considérer le produit YYY*. Ainsi, nous proposons d'accompagner l'utilisateur dans son accès à l'information au lieu de contraindre le à un set d'items que le système juge pertinents.

1.2 Enjeux

Un des enjeux majeurs pour l'amélioration de l'adéquation entre l'information retournée et les attentes des utilisateurs réside dans la mise en place de profils

riches et efficaces. Il s'agit donc d'exploiter au maximum les retours utilisateur (qu'ils soient donnés sous la forme de clics, de notes ou encore d'avis écrits) et le contexte [Abb+15].

En parallèle la forte croissance des appareils nomades (smartphones, tablettes) et par conséquent de l'informatique ubiquitaire nous oblige à repenser le rôle des systèmes d'accès à l'information. L'accent ne doit plus seulement être mis sur la performance à proprement parler mais aussi sur l'accompagnement de l'utilisateur. En d'autre termes, il semble aujourd'hui capital de diversifier les applications des systèmes de recommandation. Ces systèmes se doivent de proposer aux utilisateurs des solutions à de nouvelles problématiques comme la recommandation exploratoire [Kap+15], l'explication des résultats [Mcc+04] ou encore les questions de santé publique allant de l'accompagnement de l'utilisateur dans ses courses, l'invitant à privilégier un panier plus sain [WM15] au développement de modèles de détection des pathologies [Jel+11].

Aujourd'hui, une difficulté majeure dans le traitement de ces tâches réside dans l'évaluation. Même lorsque l'on s'intéresse au problème classique de la recommandation, il n'existe aucune méthodologie définie si ce n'est le retour fourni par les utilisateur d'un système en ligne. En effet, dans le cas hors-ligne, il est possible de considérer le problème sous deux angles différents (la prédiction de notes ou d'ensemble d'items), chacun pouvant être évalué à l'aide de diverses mesures (*e.g.* *MAE* ou *RMSE* dans le premier cas, *AUC* ou *Top-N* dans le second) suivant autant de protocoles qu'il existe d'articles traitant du sujet. Cette diversité de protocoles rend la comparaison des méthodes malaisée dans le meilleur des cas. D'un autre côté, la diversité dans l'évaluation, elle, peut être bénéfique, dans la mesure où elle sert de support à la diversifications des applications des systèmes de recommandations cités précédemment. En effet, les capacités d'exploration d'un système de recommandation ne s'évaluent pas de la même façon que son impact sur le régime alimentaire des utilisateurs, c'est pourquoi il est nécessaire de définir des mesures de référence adaptées à chacune des tâches proposées précédemment.

1.3 Contributions

Au cours de cette thèse, nous avons voulu remettre l'utilisateur au cœur du système de recommandation. Ainsi, nous proposons d'affiner la perception qu'a le système de l'utilisateur par deux moyens. Le premier est d'élargir le retour utilisateur considéré en y incorporant de nouvelles dimensions descriptives de l'utilisateur : les avis textuels. Le second consiste à caractériser la dynamique de l'utilisateur dans ses

interactions avec les items plutôt que de se focaliser sur la dynamique du contexte, comme la plupart des études le proposent.

Pour incorporer les données textuelles à l'élaboration du profil de l'utilisateur, nous avions besoin de deux choses, d'un module d'analyse de sentiments capable de classifier des textes suivant leur polarité (positive ou négative) ainsi que de la capacité d'utiliser ses résultats dans une application prédictive. Nous avons abordé ces tâches dans des travaux préliminaires [GS+13]. Dans cet article, nous proposons un module de classification de sentiments trans-média et l'utilisation de ses sorties pour la prédiction de résultats au box-office. Dans un second temps, nous avons proposé dans [Pou+14] un modèle de recommandation utilisant d'une part la polarité des textes, mais aussi les données textuelles brutes pour caractériser des similarités entre utilisateurs.

La seconde approche que nous avons explorée était la caractérisation des dynamiques utilisateur pour la recommandation. En amont de cette tâche nous avons exploré dans [GS+14] une méthode de construction d'un espace de représentation des items capable de prendre en compte ces dynamiques. L'inscription de cette thèse au sein du projet AMMICO [2] (projet visant à la mise au point d'un dispositif mobile d'assistance à la visite pour les musées) nous a permis de tester ce modèle sur des données fortement marquées par les dynamiques utilisateur : les traces utilisateur lors de leur parcours dans l'exposition Great Black Music [3]. Dès lors, nous avons pu nous intéresser aux questions de personnalisation [Guà+15], et enfin, nous avons exploré l'impact des communautés sur ces dynamiques [GS+16].

1.4 Plan de Thèse

Cette thèse s'articule autour de l'utilisation des données textuelles et temporelles pour l'accès à l'information. Nous commencerons par deux chapitres d'état de l'art avant de nous présenter nos contributions dans les trois chapitres suivants.

Chapitre 2 Dans ce chapitre, après une introduction au domaine de l'apprentissage de représentations, nous développerons une étude des méthodes de factorisation matricielle, centrales dans nos travaux. Après nous être intéressés au problème d'approximation de rang faible, nous passerons à une étude des factorisations matricielles positives et parcimonieuses.

2. http ://ammico.fr
3. http ://greatblackmusic.fr

Chapitre 3 Ici, nous nous intéresserons à la recommandation en elle-même. Nous commencerons donc par une définition du problème de recommandation et un survol des principales méthodes y répondant. Dans les deux sections suivantes, nous nous intéresserons à la recommandation contextualisée, d'abord textuellement, puis temporellement. Enfin dans la dernière partie, nous traiterons en détail les méthodes d'évaluations des systèmes de recommandation, en ligne comme hors ligne, ainsi que leurs limitations.

Chapitre 4 Comme expliqué précédemment, la moyenne des notes attribuées à un produit est un bon estimateur pour la recommandation. Cependant, son calcul nécessite d'avoir à disposition un nombre d'exemple significatif et cet estimateur n'est donc pas adapté au paradigme du démarrage à froid. Dans ce chapitre, nous proposons l'utilisation de tweets pour le calcul de caractéristiques de sentiments utilisables dans le cadre du démarrage à froid. Notre cadre expérimental hors ligne ne nous permettant pas d'utiliser ces caractéristiques pour la recommandation, nous proposons d'évaluer leur intérêt sur une tâche de prédiction connexe : la prédiction de résultats de films au box office.

Chapitre 5 Dans ce chapitre, nous allons aborder l'utilisation des avis utilisateur pour la contextualisation de la recommandation. Nous y présentons une méthode d'enrichissement des profils pour le filtrage collaboratif. Les données textuelles sont une mine d'informations sur les goûts et opinions des utilisateurs, nous avons fait l'hypothèse que cette information nous permettrait de caractériser plus facilement les utilisateurs et ainsi d'améliorer la pertinence des recommandation qui leur sont proposées. Dans un second temps nous présentons une méthode de génération de revues, espérant ainsi apporter une meilleure explication à la recommandation.

Chapitre 6 Ce chapitre est consacré à la contextualisation temporelle des systèmes de recommandation. Nous présentons une méthodologie innovante au croisement de la recommandation et de la prédiction de trajectoires. De plus, le modèle présenté permet d'unifier les deux tâches classiques de la recommandation, la prédiction d'items, et la prédiction de notes. Après avoir introduit les méthodes utilisées pour la construction de l'espace de représentation des items, nous développerons nos méthodes de personnalisation (à l'échelle de l'utilisateur et des communautés). Enfin, nous proposons une série d'expériences dont nous discuterons les résultats.

2 Apprentissage de Représentation et Factorisation Matricielle

> *hén oida hóti oudèn oida*
>
> — **Socrates**

2.1 Apprentissage de Représentation

Quelques soient les méthodes ou les tâches considérées, la première étape d'un système d'apprentissage automatique consiste à transposer les données dans un format pertinent et adapté à la tâche finale. Cette étape peut être séparée de la phase d'apprentissage (*e.g.* la transformation d'un texte en sac de mots avant traitement) ou intégrée à celle-ci (*e.g.*la recommandation par factorisation matricielle : les représentations des utilisateurs et des produits sont générées lors de l'apprentissage du modèle).

Parmi les nombreuses approches existantes, on distingue classiquement deux grandes familles de méthodes. La première se base sur l'extraction d'un dictionnaire de fonctions explicatives. Elle regroupe par exemple la méthode de segmentation *k-moyennes* [KR90] - qui calcule k prototypes à partir des données et répartit les données par calcul de similarité avec les prototypes - ou encore l'analyse en composantes principales (PCA) [Pea01 ; Hot33] - qui définit un dictionnaire en calculant les k axes expliquant le mieux la variance des données. À l'inverse la seconde famille se base sur un dictionnaire fixe et cherche des représentations qui, pour chaque élément, expliquent la contribution de chaque élément du dictionnaire. Cette approche est communément utilisée dans le cadre de la compression d'images [Hor+12] ou l'extraction de thèmes [Ble+03]. En effet, si le dictionnaire est construit de manière intelligente, seules un nombre restreint de composantes contribuent à chaque représentations.

Dans cette thèse, nous nous sommes concentrés sur les approches par factorisation matricielle. La factorisation matricielle se trouve au croisement de ces deux familles : celle-ci se base sur l'idée que les exemples observés sont décomposables en un nombre limité de facteurs latents et peut être apprise par minimisation d'une fonction

de coût représentant la différence entre la matrice des donnés et la matrice de reconstruction.

2.2 Apprentissage de Représentation par Factorisation Matricielle

Ces travaux de thèse s'articulent autour de la tâche de recommandation, et plus précisément autour du filtrage collaboratif. Les contributions proposées dans ce manuscrit s'articulent autour de l'apprentissage de représentation et, plus précisément, des méthodes de factorisation matricielle. C'est pourquoi, avant de traiter plus en avant de la recommandation en elle même, il nous paraissait important d'analyser le fonctionnement de ces méthodes.

La factorisation matricielle permet d'extraire depuis les données des représentations interprétables ce qui en fait, au même titre que l'analyse en composantes indépendantes [HO00 ; Ama+96], un modèle de référence pour les problèmes de séparation aveugle de sources (SAS). Par exemple, en astrophysique, des images d'une même zone géographique sont obtenues pour différentes longueurs d'onde, générant ainsi un cube de données dont deux axes indiquent les coordonnées spatiales alors que le troisième fournit l'information spectrale. Pour chaque pixel spatial du cube, l'information spectrale observée est perçue comme un mélange linéaire instantané des spectres sources. Le dictionnaire représente alors les différents spectres sources, et l'encodage leur contribution [Ber+07]. Il est intéressant de noter que de nombreuses tâches peuvent être considérées comme un problème de séparation aveugle des sources, ce qui explique la démocratisation de l'utilisation de la factorisation matricielle dans de nombreux domaines allant de la reconnaissance faciale [PZ11] à, bien entendu, la recommandation de produits par filtrage collaboratif [Kor08]. Dans le cas de la recommandation, le signal est remplacé par des notes et on cherche à les décomposer entre deux groupes de sources cachées : les profils utilisateur d'un côté et les caractéristiques des items de l'autre.

La factorisation matricielle est cependant un problème non trivial, notamment dans le cadre de données partiellement observées. En effet, le grand nombre de paramètres impliqués dans l'apprentissage de la factorisation la rendent sensibles aux plus faibles bruits dans les données et induisent une variance élevée. Pour répondre à ces limites, il est nécessaire d'ajouter des mécanismes de contrôle de la complexité comme des termes de régularisation pour éviter que le modèle sur-apprenne et ainsi garantir une bonne capacité de généralisation. On parle de sur-apprentissage (illustré en Fig.2.1) lorsque le modèle a une trop grande liberté dans le choix de ses paramètres et peut ainsi apprendre *par cœur* les données du jeu d'apprentissage. Ce phénomène

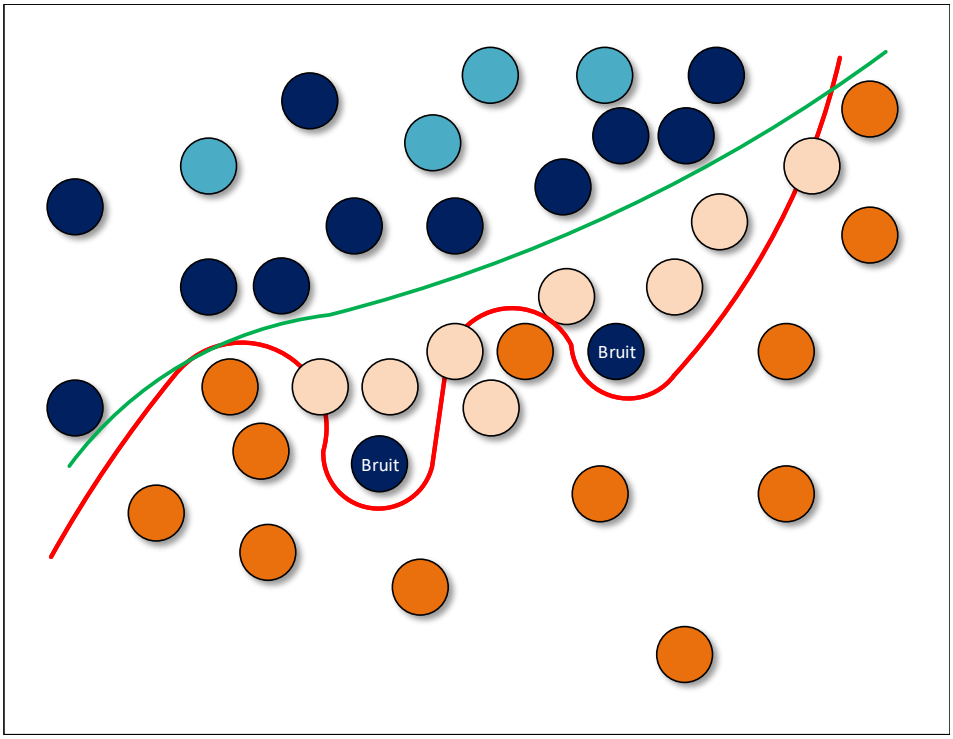


Figure 2.1: Illustration du sur-apprentissage, séparation de données en deux classes (bleu et orange). Les données disponibles à l'apprentissage sont en couleurs pleines, et les données non-observées (réservées à l'évaluation) sont en couleurs claires. Sont représentés ici deux modèles, un vert (régularisé) et un rouge (en sur-apprentissage). On remarque que le modèle rouge fait une classification parfaite des données d'apprentissage alors que le modèle vert fait deux erreurs. Cependant le modèle vert a une meilleure capacité de généralisation.

rend le modèle sensible aux plus petites fluctuations dans les données (*e.g.* présence de bruit) et induit donc de faibles capacités de généralisation.

Il existe deux manières de mettre en place ces mécanismes de contrôle, visant principalement à limiter le modèle aux seules configurations pertinentes vis à vis de la nature des données, la première consiste à contraindre le modèle lui même. Dans le cadre de la recommandation, il existe deux méthodes de référence.Nous trouverons d'une part l'approximation de rang faible, obtenue par décomposition en valeurs singulières (SVD) [GVL12] et d'autre part, la factorisation matricielle non-négative (NMF) où l'on force la valeur de chaque facteur des représentations à être positive. La seconde méthode consiste à ajouter un terme de régularisation durant la phase d'apprentissage. Par exemple la factorisation matricielle parcimonieuse [Hoy02 ; Hoy04] propose d'utiliser une régularisation $L1$ pour forcer les matrices à posséder une certaine quantité de facteurs nuls. Dans la suite de cette partie, nous allons étudier plus en détail l'approximation de rang faible et la NMF avant d'aborder le problème de la factorisation parcimonieuse.

2.3 Approximation de Rang Faible par Décomposition en Valeurs Singulières (SVD)

2.3.1 Cadre Général : Décomposition de Matrices Pleines

Soit un corpus contenant m exemples décrits par n caractéristiques continues. On représente ce corpus par la matrice de données $X \in \mathbb{R}^{n \times m}$ dont chaque colonne correspond à un exemple $x_i \in \mathbb{R}^n$. Le problème de factorisation matricielle est alors défini comme la construction de deux matrices de facteurs latents $D \in \mathbb{R}^{n \times k}$ et $H \in \mathbb{R}^{k \times m}$ telles que leur produit $\hat{X} = DH$ soit proche de X. Ainsi, trouver la factorisation D, H revient à minimiser la coût d'approximation. Si diverses fonctions de coût ont été proposées, la plus largement utilisée reste l'erreur des moindres carrés (MSE) :

$$\mathcal{L}(X, H, D) = \|X - \hat{X}\|_{Fro}^2 = \|X - DH\|_{Fro}^2 \tag{2.1}$$

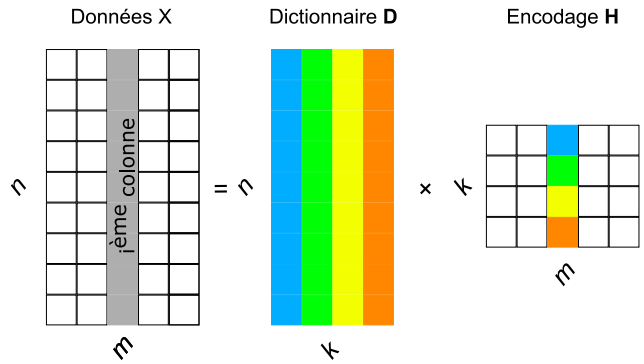


Figure 2.2: La *i*ème colonne de **X** est approximée par une combinaison linéaires des colonnes de **D**. Les coefficients de cette combinaison linéaires sont donnés par la *i*ème colonne de **H**

Ainsi, comme illustré dans la Figure.2.2 chaque exemple du corpus d'apprentissage est décomposé en une combinaison d'éléments de D (ligne) dont la pondération est stockée dans la matrice H (colonne). La matrice D est appelée le dictionnaire et contient les éléments atomiques utilisés pour la reconstruction alors que H contient les représentation latentes de chaque exemple, ce qui correspond à leur encodage.

Comme nous l'avons dit précédemment, la résolution du problème de factorisation matricielle implique un très grand nombre de variables, c'est pourquoi il est nécessaire de limiter la capacité des modèles à sur-apprendre. Une solution consiste à

borner le rang de la matrice de reconstruction $\hat{X} = DH$ par une valeur r. Cette méthode est appelée l'approximation de rang faible. On la définit comme il suit :

$$\min_{H,D} \mathcal{L}(X, H, D) \quad s.t. \;\; rang(\hat{X}) \leq r \tag{2.2}$$

La SVD est la méthode la plus couramment utilisée, notamment en recommandation, pour répondre au problème d'approximation de faible rang présenté en Eq. 2.2. Soit $\sigma \in \mathbb{R}^+$ un scalaire positif et $u \in \mathbb{R}^n$ et $v \in \mathbb{R}^m$ deux vecteurs. u et v sont respectivement appelés les vecteurs singuliers gauche et droit associés à la valeur singulière σ de $X \in \mathbb{R}^{n \times m}$ s'ils vérifient les égalités suivantes :

$$Xv = \sigma u \text{ et } X^t u = \sigma v \tag{2.3}$$

Une décomposition complète de la matrice X en valeurs singulières est un ensemble de triplets (u, v, σ) vérifiant les propriétés suivantes. Soit U (resp. V) la matrice telle qu'à chaque colonne lui corresponde un vecteur singulier gauche (resp. droit) distinct. Soient $\{\sigma_i\}$ les valeurs singulières de X ; on définit alors la matrice $\Sigma \in \mathbb{R}^{m \times n}$ telle que $\Sigma_{i,i} = \sigma_i$ et $\forall i \neq j, \Sigma_{i,j} = 0$. Par souci de clarté dans la suite du manuscrit, nous dirons, par abus de langage, que Σ est diagonale. On a alors :

$$X = U\Sigma V^t, \;\; UU^t = Id, \;\; VV^t = Id \tag{2.4}$$

Il est important de noter que toute matrice réelle accepte une décomposition en valeurs singulières positives. En effet, les valeurs singulières de X sont en fait les valeurs propres de la matrice XX^t. Or XX^t est par définition symétrique semi-définie positive et donc diagonalisable.

Les valeurs singulières de X étant positives, on peut définir un opérateur $\sqrt{\cdot}$ tel que $\forall (i,j) \;\; (\sqrt{\Sigma})_{ij} = \sqrt{\Sigma_{ij}}$. Σ étant diagonale, on a $\sqrt{\Sigma}^2 = \Sigma$. En revenant à la définition de la factorisation matricielle, on peut définir la SVD comme il suit :

$$\begin{aligned} X = DH = U\Sigma V^t, \\ D = U\sqrt{\Sigma}, \;\; H = \sqrt{\Sigma}V^t \end{aligned} \tag{2.5}$$

Lors d'une décomposition en valeurs singulières positives, telle que décrite ci-dessus, il est possible de limiter le nombre de valeurs singulières que l'on veut utiliser pour la factorisation pour obtenir une décomposition de rang faible. En effet, choisir le nombre de valeurs singulières utilisées revient exactement à régler le rang de la matrice de reconstruction $\hat{X}$. soit $\Sigma^{(r)} \in \mathbb{R}^{r \times r}$ la matrice diagonale contenant les

r plus grandes valeurs singulières de $\hat{X}$ rangées par ordre décroissant. soient $U^{(r)}$ et $V^{(r)}$ les matrices contenant les vecteurs singuliers gauche et droite associés. On définit ainsi une SVD tronquées comme il suit :

$$\begin{aligned} X \approx \hat{X} = DH = U^{(r)}\Sigma^{(r)}(V^{(r)})^t, \\ D = U^{(r)}\sqrt{\Sigma^{(r)}}, \quad H = \sqrt{\Sigma^{(r)}}(V^{(r)})^t \end{aligned} \tag{2.6}$$

D'après le théorème d'Eckart-Young, la décomposition décrite en Eq. 2.6 est celle qui permet la meilleure reconstruction $\hat{X}$ de X au sens des moindres carrés [EY36]. Toutefois, il est intéressant de remarquer qu'il existe un lien étroit entre la dimension de l'espace de représentation et le rang de $\hat{X}$. En effet, k est une borne supérieure de $rang(\hat{X})$; on pourra alors chercher à régler k de diverses manières, comme à l'aide d'information expert (*e.g.* le nombre de domaines considérés) ou encore d'un critère de reconstruction sur les données d'apprentissage. Quelle que soit la méthode utilisée, il est important de garder à l'esprit qu'une valeur faible de k, au delà d'alléger les calculs, vise avant tout permettre à chaque dimension de n'encoder que l'information utile à la différenciation des utilisateurs et des items.

2.3.2 Décomposition en Valeurs Singulières et Données Partiellement Observées

Contrairement au cadre théorique de la factorisation matricielle, dans celui de la recommandation, X n'est que partiellement observée. Il est donc nécessaire d'adapter la fonction de coût. Le critère des moindres carrés (MSE) correspondant au cadre habituel (nommément des notes entre un et cinq), c'est celui qui est communément utilisé [Kor+09]. Il définit fonction de coût $\mathcal{L}$ suivante :

$$\mathcal{L} = \frac{1}{nm} \|X - DH\|_F^2 \tag{2.7}$$

Cependant, cette fonction de coût prend en compte les données non-observées de X. La solution la plus simple à ce problème consiste à ne considérer que les couples (u, i) observés. La formulation de 2.7 devient alors :

Soit m le nombre d'observations dans l'ensemble d'apprentissage,

$$\mathcal{L} = \frac{1}{m} \sum_{(u,i) \in App} \left(X_{(u,i)} - d_u \cdot h_i \right)^2 \tag{2.8}$$

Cette formulation a l'avantage de permettre un apprentissage par application direct de l'algorithme de descente de gradient stochastique (SGD) que l'on préférera dans ce cas aux approches multiplicatives décrites en Section 2.2 qui nécessiteraient l'utilisation d'un filtre pour occulter les parties non observées de la matrice. De plus, au lieu de nécessiter le stockage de X de dimension $n \times m$, cette méthode se suffit de m triplets de la forme (u, i, X_{ui}). Il est alors facile de mélanger aléatoirement l'ensemble d'apprentissage en vue de la SGD.

2.4 Factorisation Matricielle Non-Négative

Ici, nous allons nous intéresser à une seconde forme de contraintes de régularisation. Au lieu de considérer le rang de la matrice de reconstruction, nous allons nous concentrer sur la structure interne de cette matrice. Dans une tâche de recommandation, les données sont souvent exprimées sous la forme de notes positives, ainsi, la matrice d'exemples est elle même non-négative [1]. Intuitivement, dans un souci d'interprétabilité, il paraît judicieux d'étendre cette non-négativité aux matrices de représentation. On cherche à définir les éléments du dictionnaire D comme des concepts respectant la forme des données pour en faciliter l'explication. La matrice H représente les combinaisons des concepts du dictionnaire pour la reconstruction des exemples ; ainsi, lui permettre de prendre des valeurs négatifs pourrait avoir des effets étranges, voire néfastes : un concept à valence négative pourrait alors contrecarrer l'effet d'un autre, ce qui réduirait considérablement la compréhension que l'on a de la décomposition.

La contrainte de non-négativité agit comme un fort paramètre de régularisation et s'utilise dans de nombreux domaines comme la segmentation de documents [Xu+03], la génération automatique de résumés [Par08], la reconnaissance d'images [Zaf+06 ; Ben+06] et, bien entendu la recommandation [MS10]. Sa popularité est aussi en grande partie due à sa facilité d'apprentissage. Dans [LS01], les auteurs proposent un algorithme d'apprentissage efficace permettant de passer outre la non convexité du problème. Remarquant que le problème en D devient convexe pour H fixe et inversement, les auteurs proposent un apprentissage alterné des deux matrices, H_{t+1} est calculé à partir de H_t en considérant D et X fixes puis il en va de même pour D_{t+1}, en considérant H et X fixes :

$$H_{p+1} = H_p \frac{D^t X}{D^t D H_p} \text{ et } D_{p+1} = D_p \frac{X H^t}{D_p H H^t} \tag{2.9}$$

1. *Stricto sensu*, la notion de positivité en français, contrairement à l'anglais, inclus le 0. Ainsi nous devrions parler de Factorisation Matricielle Positive. Cependant par souci de clarté et d'alignement sur le vocabulaire anglo-saxon, nous utiliserons dans ce manuscrit la formulation *non-négative*.

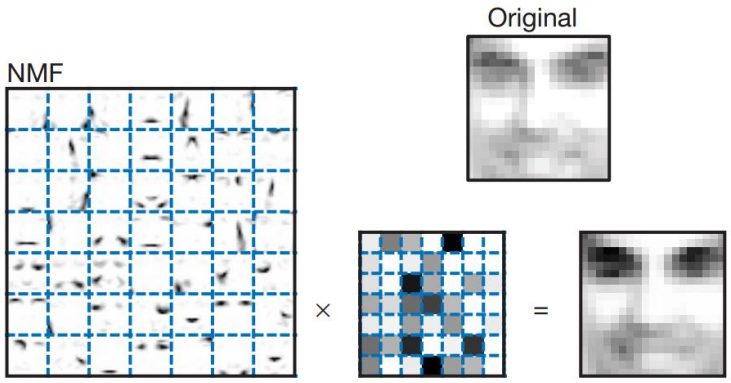


Figure 2.3: Illustration de l'application de la NMF à la reconnaissance faciale tirée de [LS99].La matrice **NMF** représente le dictionnaire **D** (cf. Fig.2.2) contenant les éléments de base. La matrice à droite du signe $\times$ représente l'encodage **H**. Enfin la matrice à droite du signe $=$ est la reconstruction de l'original par combinaison linéaire de **D** et **H**.

Ces règles de mise à jour ont l'avantage d'être aisément dérivables et prouvées convergentes. L'algorithme d'apprentissage (disponible en Algorithme 1) proposé dans [LS01] est simple à implémenter vu que les pas de gradients sont calculés automatiquement dans la dérivation. Cependant, dans [Lin07], les auteurs soutiennent qu'une méthode de gradient projeté peut converger plus vite vers une solution satisfaisante.

Data : X, H_0, D_0

Result : H, D

$H \leftarrow H_0, D \leftarrow D_0$;

while *convergence non atteinte* **do**

$H \leftarrow H \frac{D^t X}{DHH^t}$

$D \leftarrow D \frac{XH^t}{D^t DH}$

end

Algorithme 1 : Mise à jour multiplicative utilisant le coût de Frobenius pour la NMF

Dans le cadre d'une matrice X partiellement observée, nous nous heurtons au même problème que dans la Sec. 2.3.2. S'il est possible de contourner le problème en appliquant un masque de filtrage à X, il est ici aussi préférable de redéfinir la fonction de

coût telle que présentée dans Eq. 2.8. L'algorithme d'apprentissage devient alors :

Données : $App = \left\{x_{(u,i)}, u, i\right\}, H_0, D_0, \lambda$
Résultat : H, D
$H \leftarrow H_0, D \leftarrow D_0$;
tant que *convergence non atteinte* **faire**
  Mélanger App
  pour $(x_{(u,i)} \in App$ **faire**
    Soit d_u la u-ème ligne de D et h_i la i-ème ligne de H
    Calcul de la prédiction $\hat{x}_{(u,i)} = d_u \cdot h_i$. Soit $\delta \leftarrow \hat{x}_{(u,i)} - x_{(u,i)}$
    $d'_u \leftarrow max(0, d_u - \mu(\lambda d_u + \delta h_i))$
    $h'_i \leftarrow max(0, h_i - \mu(\lambda h_i + \delta d_u))$
    $d_u \leftarrow d'_u$ et $h_i \leftarrow h'_i$
  fin
fin

Algorithme 2 : Mise à jour par descente de gradient stochastique pour la NMF pour les matrices partiellement observées.

2.5 Factorisation Matricielle Non-Négative Parcimonieuse

Il est bien entendu possible d'ajouter d'autres contraintes à celle de non négativité, comme, par exemple, la parcimonie qui consiste à limiter le nombre d'entrées non nulles dans la matrice de reconstruction. L'utilisation de ce type de contraintes dans le cadre de la factorisation matricielle provient d'une autre branche de l'apprentissage de représentations : l'apprentissage de dictionnaires, domaine dans lequel l'utilisation de la parcimonie est très largement répandue. L'idée sous-jacente est qu'une grande parcimonie implique une forte compression des représentations, même si le nombre de dimensions de l'espace des représentations reste élevé, vu que seulement un petit nombre d'entre elles sont actives en même temps [Lee+07]. Cette contraintes est définie de la façon suivante :

$$\min_{H,D} \mathcal{L}(X, H, D) + \lambda_H \|H\|_1 \quad s.t. \quad \forall i, \|d_i\|_2 = 1 \tag{2.10}$$

Dans l'équation 2.10, on devrait utiliser $\|H\|_0$ le nombre de valeurs non nulles dans H au lieu de $\|H\|_1$, cependant, $\|H\|_0$ n'étant pas dérivable on lui préfère habituellement $\|H\|_1$. La contrainte sur la norme euclidienne des colonnes d_i de D sert à limiter le nombre de solutions équivalentes atteignables (en multipliant D par un réel supérieur à 1 et en divisant H par la même valeur).

Dans [Hoy02], les auteurs proposent d'incorporer la contrainte de parcimonie aux règles de mise à jour multiplicatives décrites dans l'Algorithme 1. Pour ce faire, ils préconisent l'utilisation d'un gradient projeté pour le dictionnaire D, comme décrit dans l'Algorithme 3.

Données : $X, H_0, D_0, \lambda_H, \mu$
Résultat : H, D
$H \leftarrow H_0, D \leftarrow D_0$;
tant que *convergence non atteinte* **faire**
  $H \leftarrow H \cdot \frac{D^t X}{\lambda_H + DHH^t}$
  $D \leftarrow max(0, D + \mu(X - DH)H^t)$
  $\forall i, d_i \leftarrow \frac{d_i}{\|d_i\|_2}$
fin

Algorithme 3 : Algorithme d'apprentissage de la NMF parcimonieuse [Hoy02]

La parcimonie des matrices D et H peut être interprétée de diverses façons. Dans [Hoy04], les auteurs nous proposent la vision suivante. La parcimonie des lignes de H signifie que chaque concept (*i.e.* les colonnes de D) ne doit être impliqué que dans la reconstruction d'un nombre restreint d'exemples (*i.e.* les colonnes de X). En d'autre termes, les concepts de D sont supposément discriminants. La parcimonie au niveau des lignes de D implique que chaque caractéristique d'entrée doit être utilisée aussi peu que possible dans la définition des différents concepts. Ce qui revient à dire que les caractéristiques d'entrée sont supposément discriminantes. La parcimonie sur les colonnes de H implique que chaque exemple de X ne doit être construit qu'avec un nombre réduit de concepts pertinents, c'est la vision classique de la parcimonie. Enfin, la parcimonie sur les colonnes de D implique que chaque concept doit utiliser aussi peu de caractéristiques d'entrée que possible ; on retrouve ici la capacité de la NMF à définir des représentations *part-based*.

3 Systèmes de Recommandation

> *On m'a dit le plus grand bien de vos harengs pomme à l'huile.*
>
> **— Jean Dujardin**

3.1 Cadre général de la Recommandation

Dans ce chapitre, nous allons nous intéresser à la tâche de recommandation, définie comme il suit : *étant donné un ensemble d'items, on appelle recommandation, la suggestion à un utilisateur d'un sous-ensemble de ces items contenant ceux qu'il trouvera les plus pertinents* [SG11 ; Bre+98 ; MH04]. La littérature foisonne de systèmes de recommandation se différenciant les uns des autres par le type de données utilisé aussi bien que par les méthodes employées pour la génération des recommandations. Le but du système de recommandation est de calculer, pour chaque utilisateur, un score associé à chaque item. Suivant la méthode d'évaluation, ce score servira à définir un ordre de pertinence sur les items ou directement à estimer la note qui leur est associé. Dans un cas comme dans l'autre, on peut définir le système de recommandation comme suit :

Soit U l'ensemble des utilisateurs et I l'ensemble des items et R un ensemble (discret ou continu) totalement ordonné. Soit F la fonction de prédiction de score :

$$F \ : \ U \times I \rightarrow R \tag{3.1}$$

Suivant le paradigme considéré, R peut avoir deux interprétations : soit il représente une échelle de préférence, permettant ainsi d'ordonner les items du plus pertinent au moins pertinent. Soit R représente directement l'appétence d'un utilisateur pour un item, sans chercher à garder l'idée d'ordre. On peut alors définir deux tâches distinctes : la prédiction d'items, et la prédiction de notes.

La prédiction d'items consiste à proposer à l'utilisateur un ensemble d'items adaptés à ses goûts et besoins [Sar+01]. Ce problème peut être formalisé de la façon suivante :

étant donné une liste d'items vu par un utilisateur, quels items va-t-il consulter dans le futur ? Cette tâche peut être évaluée à l'aide de mesures TopK [SG11].

La prédiction de notes [Res+94 ; Ben+07 ; Kor08 ; AT05] se base sur l'idée suivante : si un modèle est capable d'évaluer correctement les notes que les utilisateurs mettraient aux items, alors les profils qu'il propose sont une bonne approximation des goûts des utilisateurs. En 2009, le succès du challenge Netflix [1] a permis la démocratisation de cette formulation du problème.

Un système de recommandation vise donc à extraire autant d'information que possible pour la caractérisation des similarités entre les utilisateurs et les items [Kor+09 ; Bur07]. On distinguera trois principales familles de méthodes : celles basées sur le contenu (*content based)*, les connaissances expertes ou enfin le filtrage collaboratif. L'analyse démographique et les connaissances expertes s'appuient sur des données extérieures pour proposer une recommandation. Dans la suite de cette section, nous commencerons par introduire les notions d'évaluation des systèmes de recommandation, puis nous nous intéresseront aux modèles *content based*, au filtrage collaboratif et enfin à la contextualisation de la recommandation.

L'évaluation d'un système de recommandation peut être effectuée suivant deux paradigmes généraux, l'évaluation en ligne (*online*) ou hors ligne (*offline*) Seulement, il n'est pas toujours évident, de faire tester nos systèmes par des utilisateurs, c'est pourquoi on différencie deux cadres d'évaluation distinct, l'un en ligne, ou *online* et l'autre hors ligne, ou *offline*[SG11]. Dans le cas en ligne de vrais utilisateurs interagissent avec le système et testent un ou plusieurs modèles. En général, on effectue une comparaison empirique de la satisfaction de différentes populations d'utilisateur (chaque population étant rattachée à un modèle) à l'aide d'un test A/B [Koh+09]. L'évaluation en ligne apporte ainsi de précieuses informations sur les préférences des utilisateurs ainsi que que leur degré de satisfaction après l'utilisation du système de recommandation. Cependant, leur utilisation n'est pas toujours possible dans la mesure où leur déploiement est complexe et onéreux.

L'évaluation hors ligne, elle utilise des données précédemment enregistrées pour l'évaluation du système. Cette évaluation est effectuée en testant si la recommandation fournie par le système correspond aux intérêts décrits par les utilisateurs. Grâce à la mise à disposition de jeux de données, l'évaluation hors ligne est peu coûteuse et garantit une reproductibilité des expériences dans un environnement d'évaluation fixe. C'est pour toutes ces raisons que la plupart des travaux académiques se sont concentrés sur l'utilisation de méthodes d'évaluation hors ligne. Tout protocole d'évaluation comporte deux composantes fondamentales : la mesure d'évaluation,

1. http ://www.netflixprize.com/

qui définit ce que l'on évalue, et la méthodologie, qui définit comment on évalue. Dans le cas des systèmes de recommandation, si certaines mesures sont largement acceptées et utilisées [Her+04 ; GS09], il n'existe cependant pas de consensus sur la méthodologie [Bel+11].

De manière générale, on apprend le modèle sur les données utilisateur disponibles et on évalue sa capacité à proposer de *bonnes*[2] recommandations sur des données utilisateur additionnelles. Dans un cas *offline*, il est donc nécessaire de simuler les actions de l'utilisateur post-recommandation. Ceci est usuellement réalisé en divisant le jeu de données en deux : un ensemble d'apprentissage App (correspondant à l'historique des utilisateurs) et un ensemble de test Te (simulant leur réaction face à la recommandation). On peut aussi ajouter un troisième ensemble, dit de validation, servant à fixer les éventuels hyper-paramètres du modèle.

Un système de recommandation, comme tout système prédictif est sensible au sur-apprentissage. Ici, il se traduira par une sur-valuation des couples utilisateur-produit marginaux lors de l'apprentissage. Par exemple, la rencontre d'un fervent admirateur de Truffaut, Cocteau et Buñuel fan de Vin Diesel ne devra pas inciter le système à rapprocher Belle de Jour de Triple X. Nous évoquerons par la suite les méthodes utilisées pour lutter contre ce phénomène.

Dans la suite, on considère que si un utilisateur u a une note dans l'ensemble d'apprentissage (resp. de test), alors u fait parti de l'ensemble des utilisateurs d'apprentissage U_{App} (resp. de test U_{Te}). De la même façon, on dénote App_u (resp. Te_u) l'ensemble d'apprentissage (resp. de test) de l'utilisateur u. Enfin, on appelle $\mathcal{R}_u$ l'ensemble des notes collectées pour u (et $\mathcal{R}$ la matrice positive contenant l'ensemble des notes pour l'ensemble des utilisateurs).

3.2 Méthodes de Recommandation

Dans cette section, nous allons présenter différentes méthodes de recommandation. Après un aperçu des méthodes basées sur le contenu, nous nous intéresserons au filtrage collaboratif, et plus précisément aux méthodes de factorisation matricielle présentées dans le chapitre précédent. Ces méthodes permettent, à l'aide d'une représentation latente, de compresser l'information comportementale des utilisateurs vis à vis des produits et inversement. Ainsi, en expliquant pourquoi des utilisateurs (ou des items) se ressemblent, elles permettent une prédiction de notes précise.

2. au sens de la mesure considérée

3.2.1 Recommandation Basée sur le Contenu

Dans les méthodes *content based*, pour un utilisateur u et un item i, la pertinence $c(u, i)$ est calculée par similarité entre i et l'ensemble des items $j \in I_u$, où I_u est l'ensemble des items déjà vus par u. Par exemple, si u veut acheter un livre, le système cherchera et lui proposera ceux qui ont le plus de similarités avec ceux qu'il a acheté précédemment (langue, genre, auteur, sujet, etc.).

Ces méthodes trouvent leur origine dans les algorithmes de recherche d'information [BYRN99 ; Sal89] et de filtrage d'information [BC92]. Les grandes avancées de ces domaines au cours des années 1990 et l'explosion de la quantité d'information textuelle disponible a permis le profilage des items et, par ce biais celui des utilisateurs. Le profil est habituellement calculé par extraction de mots-clefs depuis une description textuelle de l'item [PB97 ; BYRN99]. On construit alors un espace vectoriel dans lequel chaque mot-clef est associé à une dimension. La représentation d'un item est alors calculée à l'aide d'une méthode de pondération telle que TF-IDF (*term frequency/inverse document frequency*) [Sal89].

Ainsi, on peut donc définir les caractéristiques d'un item i comme un vecteur de la forme $\psi_i = (w_{1i}, \ldots, w_{ki})$ où w_{k_j} représente l'importance du mot-clef k_j pour l'item i.

Une fois les profils des items défini, on définit la représentation ϕ_u de l'utilisateur u à l'aide des profils des items qu'il a vu précédemment. Cette agrégation peut être faite de diverses façons, on pensera à des méthodes par moyenne des vecteurs item [BS97], retour de pertinence [Roc71] ou encore à l'aide d'un classifieur Bayesien [PB97]. On peut alors calculer la pertinence $c(u, i)$ d'un item i pour l'utilisateur u par un calcul de similarité. Dans la littérature, la méthode la plus utilisée est la similarité *cosine* [PB97 ; Sal89]

$$c(u, i) = cos(\phi_u, \psi_i) = \frac{\phi_u \cdot \psi_i}{|\phi_u| \times |\psi_i|} \tag{3.2}$$

Ces méthodes présentent deux avantages principaux. Tout d'abord, elles permettent d'ajouter facilement de nouveaux items : le système n'étant pas basé sur des interactions entre les utilisateurs et les items, un produit peut être recommandé, même si personne ne l'a encore consulté (cas du démarrage à froid). De plus, elles sont transparentes à l'utilisation : il est en effet possible d'expliquer les résultats du système de recommandation en listant explicitement les caractéristiques en commun entre le profil de l'utilisateur et celui de l'item considéré.

Cependant, ces méthodes connaissent trois limitations majeures [BS97]. D'une part, elles ne peuvent considérer que les caractéristiques explicitement associées à un item. De plus, si deux items sont représentés par les mêmes caractéristiques, ils ne sont pas différentiables aux yeux du système. Enfin, ces méthodes se basant sur des similarités entre les items précédemment vus et le reste du corpus, elles ne peuvent effectuer que des tâches d'exploitation, rendant l'exploration impossible. En d'autres termes, si un utilisateur n'a vu que des films d'action, le système sera incapable de proposer un film d'un autre genre, aussi bon soit il.

3.2.2 Recommandation par Filtrage Collaboratif

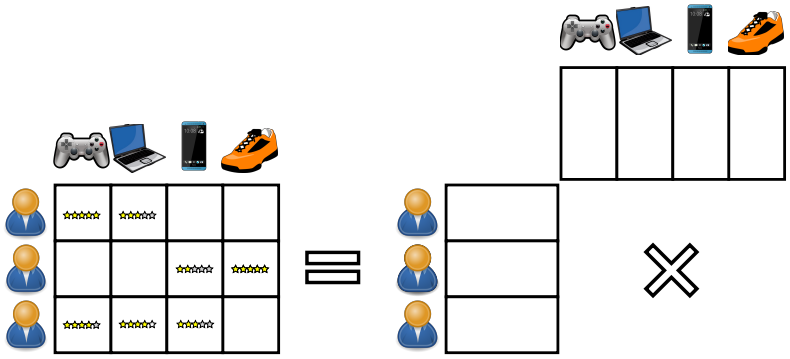


Figure 3.1: Les notes émises par les utilisateurs sont stockées dans la matrice de notes M. Dans cette matrice, chaque ligne correspond aux notes émises par l'utilisateur correspondant. De la même façon, chaque colonne correspond aux notes reçues par l'item associé. M est alors décomposée un un produit de deux facteurs, l'un contenant les représentations latentes des utilisateurs, l'autre celle des items.

Contrairement aux méthodes *content based*, les méthodes de recommandation par filtrage collaboratif exploitent le *feedback*, *i.e.* les interactions entre les utilisateurs et les items. Il est possible de distinguer deux grandes familles d'approches, les méthodes à mémoire (aussi appelées méthodes à heuristiques) et les méthodes à modèles [AT05]. Les méthodes à mémoire appliquent des heuristiques à l'intégralité des notes de l'ensemble d'apprentissage pour effectuer leur prédiction. Par exemple, dans [Her+99], le voisinage d'un utilisateur u est déterminé par la similarité entre ses notes et celles de ses voisins. Ainsi, pour prédire la note que u donnera à un item inconnu, on consulte les notes données à cet item par les voisins de u. Cette thèse portant sur les méthodes à modèles, on utilisera par la suite indifféremment les termes de filtrage collaboratif et de méthodes à modèles.

La famille de méthodes à modèle la plus largement utilisée est celle des méthodes de factorisation matricielle. ici, la factorisation matricielle ne vise plus à décoder un signal à l'aide d'un dictionnaire mais à expliquer la note mise par un utilisateur à un produit par leurs profils latents. Ainsi, nous proposons de redéfinir les notations comme il suit. Soit la matrice $\mathcal{R}$ contenant les notes émises par les utilisateurs.

$\mathcal{R} \in \mathbb{R}^{n_U \times n_I}$ est construite en indexant les n_U utilisateurs et les n_I items du système et en attribuant à chaque cellule $\mathcal{R}_{ui}$ la note r_{ui} que l'utilisateur u a donné à l'item i. On peut alors extraire des représentations latentes des utilisateurs et des items en cherchant une factorisation de la matrice $\mathcal{R}$. Soit d la dimension de l'espace des représentations, on définit alors deux matrices latentes $\Phi_u \in \mathbb{R}^{n_U \times d}$ - contenant les profils des utilisateurs - et $\Psi_i \in \mathbb{R}^{n_I \times d}$ - contenant les profils des items - telles que :

$$\mathcal{R} \approx \Phi_u \Psi_i^t \tag{3.3}$$

On définit alors le terme $fm(u,i)$ associé à la factorisation matricielle de la façon suivante :

Soient $\phi_u = \Phi_U[u,:], \quad u \in U$ et $\psi_i = \Psi_I[i,:], \quad i \in I$,

$$fm(u,i) = \phi_u \cdot \psi_i \tag{3.4}$$

Par définition, la tâche de prédiction de notes implique de se placer dans le cadre de la factorisation de matrices partiellement observées et l'on préférera donc une approche stochastique à une approche en bloc. Avec le système de notation adapté à la recommandation, la fonction de coût $\mathcal{L}$ de l'équation Eq.2.8 définie en Sec. 2.4 s'écrit alors :

Soit m le nombre d'observations dans l'ensemble d'apprentissage App,

$$\mathcal{L} = \frac{1}{m} \sum_{(u,i) \in App} \left(\mathcal{R}_{(u,i)} - \phi_u \cdot \psi_i \right)^2 \tag{3.5}$$

Sous cette forme, les méthodes de factorisation matricielles sont peu stables et nécessitent de nombreux ajustements pour fonctionner efficacement. C'est pourquoi il est commun de normaliser les données en ajoutant au terme de factorisation matricielle des termes de biais [Kor+09]. Ces biais peuvent être vus comme un *prior*. La NMF ne cherche plus à apprendre les notes, mais les décalages de chaque couple utilisateur-produit par rapport à une moyenne. On dénombre trois biais communément utilisés, un terme général, un terme centré sur les utilisateurs, et enfin un terme centré sur les produis.

Biais Général Le premier terme est un biais général. Il représente la note moyenne émise, indépendamment du produit ou de l'utilisateur. Il est calculé sur l'ensemble

d'apprentissage (App) de la façon suivante : soit m le nombre de notes dans la base d'apprentissage,

$$g_0 = \frac{1}{m} \sum_{(u,i) \in App} r_{ui} \tag{3.6}$$

L'étape d'inférence s'effectue en renvoyant la valeur de g_0 comme estimation de la note, et ce quelque soit le couple (u, i) considéré. Ce biais donne une première estimation robuste du comportement général sur l'ensemble des données, son intérêt reste cependant limité par son manque de personnalisation.

Biais Utilisateur Le second terme est le biais utilisateur. Il consiste en la moyenne des notes émises par chaque utilisateur. Il est donc calculé sur l'ensemble d'apprentissage de la façon suivante : soit m_u le nombre d'items notés par l'utilisateur u,

$$\forall u \in U, \quad g_1(u) = \frac{1}{m_u} \sum_{(u,i) \in App} r_{ui} \tag{3.7}$$

Ici, l'inférence devient personnalisée : elle consiste, pour un utilisateur u, à renvoyer la valeur de $g_1(u)$, quelque soit l'item considéré. Cette méthode a beau être moins brutale que la précédente, elle en reste tout du moins peu efficace. En effet, un utilisateur va pouvoir donner des notes positives ou négatives aux produits selon s'il les a ou non aimés. *De facto*, on se trouve confronté à une variance élevée sur les notes données par un utilisateur à différents items.

Biais Item Le troisième et dernier terme est le biais item. A l'instar du biais utilisateur, il est calculé en effectuant la moyenne des notes émises pour chaque item. Il est donc calculé sur l'ensemble d'apprentissage de la façon suivante : soit m_i le nombre d'utilisateurs ayant noté i,

$$\forall u \in U, \quad g_2(i) = \frac{1}{m_i} \sum_{(u,i) \in App} r_{ui} \tag{3.8}$$

Si tous les goûts sont dans la natures, certains sont plus courants que d'autres : les utilisateurs ont tendance à s'accorder sur la qualité d'un produit, donnant des notes proches. Ainsi, même s'il ne prend pas en compte l'avis de l'utilisateur, ce modèle constitue un très bon modèle de référence.

Formulation de la Factorisation Matricielle avec Biais L'apprentissage des représentations des utilisateurs et des items s'effectue en appliquant une descente de gradient à la fonction de coût décrite en Eq. 3.9. On remarque l'apparition de paramètres de régularisation λ dans cette formulation. Leur présence vise à lutter contre le phéno-

mène de surapprentissage induit par la complexification du modèle d'estimation des notes.

$$\mathcal{L} = \frac{1}{m} \sum_{(u,i) \in App} (\phi_u \cdot \psi_i) + g_0 + g_1(u) + g_2(i) + \lambda_U \|\phi_u\|_F^2 + \lambda_I \|\psi_i\|_F^2 \tag{3.9}$$

l'Algorithme 4 ci-dessous illustre la méthode d'apprentissage par descente de gradient stochastique. En pratique, le pas d'apprentissage μ n'est pas fixe mais décroît au cours du temps (il est multiplié, à la fin de chaque époque par un facteur inférieur à 1). De plus, Φ et Ψ ont chacun leurs propres pas d'apprentissage μ_u et μ_i et paramètres de régularisation λ_u et λ_i, optimisés sur l'ensemble de validation.

Données : $App = \{(r_{u,i}, u, i)\}, \Phi_{U0}, \Psi_{I0}, \lambda, \mu$
Résultat : Φ_U, Ψ_I
$\Phi_U \leftarrow \Phi_{U0}, \Psi_I \leftarrow \Psi_{I0}$;
tant que *Convergence non atteinte* **faire**
  Mélanger App;
  pour $(r_{ui}, u, i) \in App$ **faire**
    ϕ_u la u^{eme} ligne de Φ_U, ψ_i la i^{eme} ligne de Ψ_I;
    $\hat{r}_{ui} \leftarrow \phi_u \cdot \psi_i + g_0 + g_1(u) + g_2(i)$;
    $\delta \leftarrow \hat{r}_{ui} - r_{ui}$;
    $\phi_u \leftarrow \phi_u - \mu \cdot (\lambda \phi_u + \delta \psi_i))$;
    $\psi_i \leftarrow \psi_i - \mu \cdot (\lambda \psi_i + \delta \phi_u))$;
  fin
fin

Algorithme 4 : Descente de gradient stochastique pour la factorisation de matrice de notes partiellement observées.

Factorisation Matricielle et Prédiction d'Items Si elle est, par sa nature, plus adaptée à la prédiction de notes, il est possible de l'adapter la factorisation matricielle à la tâche de prédiction d'items [JT12]. Ce modèle propose d'appliquer un critère d'ordonnancement correspondant à la fonction de coût suivante décrite en Eq.3.10. Cette approche *pairwise* permet de rapprocher les représentations items que les utilisateurs ont noté de manière similaire tout en éloignant ceux dont les notes sont trop éloignées.

$$\mathcal{L} = \sum_{u \in U} \sum_{i \in I} k_{ui} \sum_{j \in I \setminus i} s_j \left((\hat{r}_{ui} - \hat{r}_{uj}) - (r_{ui} - r_{uj}) \right)^2 \tag{3.10}$$

$\hat{r}_{ui}$ est calculé à l'aide de la méthode classique de factorisation matricielle.

k_{ui} représente la pondération attribuée à chaque paire *user-item* par le système. Cette pondération permet de ne considérer que certains types de paires comme, par exemple, le *feedback* positif. Ici, en fixant $k_{ui} = 0$ si $r_{ui} = 0$, et $k_{ui} = 1$ sinon, on permet au modèle de limiter l'apprentissage de ses paramètres aux seuls retours utilisateurs réels. Le paramètre s_j représente une pondération sur les items, en d'autres termes, ce paramètre représente l'importance associée à l'item j par le système. En pratique, les auteurs proposent, soit de pénaliser les items vus par trop d'utilisateurs en fixant $s_i = |U_i|$ où U_i représente l'ensemble des utilisateurs ayant noté i, soit de fixer $s_i = 1$ pour tous les items, rendant ce paramètre transparent.

L'inférence est ensuite effectuée en récupérant les items ayant le meilleur score (calculé avec la formule de factorisation matricielle classique).

3.2.3 Recommandation Contextualisée

Le contexte est un concept aux multiples facettes qui a été largement étudié dans de multiples champs disciplinaires [Ado+11 ; Hus+12]. Dans [Dey01], l'auteur le décrit comme *toute information pouvant être utilisée pour caractériser l'état d'une entité*. Ici, l'entité peut être un utilisateur, un item ou leur interaction. En d'autre termes, le contexte peut représenter aussi bien la géolocalisation de l'utilisateur que l'appareil qu'il utilise, la météo, la date ou des concepts plus abstraits comme son humeur. Le contexte n'est pas facile à extraire, ni à exploiter. Par exemple, si un utilisateur adore la cuisine italienne, lui proposer un restaurant à Naples alors qu'il est à Paris n'est peut être pas la meilleure idée. Au delà du manque d'intérêt, ce genre de recommandation *hors sujet* peut entamer la confiance qu'a l'utilisateur pour le système de recommandation et affecter sa propension à l'achat [Gor+11].

Ainsi on parlera de système de recommandation contextualisée, ou CARS (*Context-Aware Recommender Systems*). On se restreindra ici aux cas où le contexte est explicitement connu [Dou04], qu'il soit qualitatif (*e.g.* le pays où se trouve l'utilisateur) ou quantifiable (*e.g.* les horodatages associés aux diverses interactions utilisateur-item - click, ajout panier, achat, etc.). La définition du système de recommandation proposée en Eq.3.1 devient alors : soit $C = [C^1, \ldots, C^n]$ l'ensemble des dimensions contextuelles.

$$F \ : \ U \times I \times C \rightarrow R \tag{3.11}$$

Dans [Ado+11], les auteurs différencient trois types de CARS suivant le moment où intervient le contexte : le pré-filtrage contextuel, le post-filtrage contextuel et les systèmes à modélisation contextuelle.

Le pré-filtrage contextuel consiste à filtrer les profils utilisateurs en fonction du contexte en amont de l'estimation des recommandations. À l'inverse, le post-filtrage contextuel, consiste à filtrer les résultats du système de recommandation en fonction du contexte de l'utilisateur. Dans un cas comme dans l'autre, l'estimation des propositions peut être effectuée à l'aide de modèles de recommandation non contextualisée.

Contrairement aux deux familles précédentes, les systèmes à modélisation contextuelle intègrent directement les informations contextuelles au calcul des prédictions. C'est à ces méthodes que nous nous intéresserons.

3.3 Contextualisation Temporelle

La prise en compte du temps est primordiale pour la recommandation, en effet il affecte aussi bien les utilisateurs (nos goûts changent avec le temps) que les items (les modes passent) et, bien évidemment, le contexte, qui est, par définition ancré dans l'instant de la recommandation. C'est la prise en compte de ces dynamiques, sur le court comme le long terme, qui permettent l'amélioration, non seulement des performances, mais aussi de la compréhension que nous avons des résultats.

3.3.1 Différents Contextes Temporels

Les informations temporelles (*i.e.* les attributs contextuels associés au temps tels que l'heure, le jour, le mois, etc.) présentent l'avantage d'être faciles à collecter. En effet, la plupart des systèmes proposent l'horodatage des interactions (*e.g.* note, *like*, *check-in*, avis) entre les utilisateurs et les items.

Si l'on considère la définition du temps donnée dans le dictionnaire Larousse [3], on comprend que le contexte temporel peut être vu de diverses façons ; on peut le voir comme un ordre dans une séquence d'événements (*e.g.* l'enchaînement des jours de la semaine ou une séquence d'achats) ou encore la durée d'un événement (*e.g.* une piste audio, un film ou un trajet).

Dimension polyvalente, le temps peut être vu comme un phénomène linéaire (dans le cas d'un ordonnancement) ou périodique (la répétition des jours de la semaine), continu ou discret (découpage en unités de temps), il peut même se voir affublé d'une structure hiérarchique (les heures d'une journée, les journée d'une semaine, etc.). Il est ainsi possible d'utiliser un large panel d'implémentations contextuelles du temps.

3. http ://www.larousse.fr/encyclopedie/divers/temps/96458

On pourra par exemple considérer l'horodatage des interactions utilisateur-item comme une variable continue pour modéliser un effet d'oubli (plus un événement est lointain, moins il importe), utiliser la périodicité des saisons pour affiner une recommandation touristique et ainsi éviter de proposer du ski en plein mois d'août ou encore tenter de caractériser la dynamique des utilisateurs (*i.e.* l'évolution au cours du temps de la perception que le système a de chaque utilisateur).

Ces informations temporelles faisant elles-mêmes partie du contexte général, les systèmes de recommandation temporels (TARS) sont donc une sous-famille de CARS. Ainsi, si l'on considère T l'ensemble des information temporelles utilisées par le système, la formulation générale donnée en Eq.3.11 devient :

$$F \ : \ U \times I \times T \rightarrow R \tag{3.12}$$

Si en général les TARS se contentent d'utiliser les horodatages associés au moment où l'utilisateur a noté un item, on peut considérer d'autres contextualisations temporelles comme l'horodatage de l'achat, de l'ajout de l'item au catalogue du système ou encore de l'inscription de l'utilisateur.

3.3.2 Systèmes de Recommandation et Contexte Temporel

Les premiers essais de prise en compte du temps dans la recommandation datent de 2001 : dans [AT01] les auteurs proposent l'utilisation de représentations multidimensionnelles (dont le temps) pour les utilisateurs et les items. Dans [Zim+01], la tâche de filtrage collaboratif se voit associée à un ordonnancement des données et est traitée comme un problème de séries temporelles.

Il faut ensuite attendre quelques années avant que le sujet revienne au goût du jour, notamment grâce à [AT05 ; AT08] et [Kor09a] dont l'apport prépondérant a influencé le domaine pendant de nombreuses années. Par la suite, nous ferons la distinction entre trois types de contextualisation temporelle : les méthodes continues où le temps s'apparente à un horodatage, les méthodes catégorielles, où le temps est représentée par un ensemble discret (*e.g.* le jour de la semaine) et enfin les méthodes adaptatives qui ajustent dynamiquement certaines de leurs paramètres en fonction du contexte temporel.

Contextualisation Temporelle Continue Dans le cas de la contextualisation temporelle continue, le temps est représenté comme une variable continue et l'heure de la recommandation est explicitement utilisé dans le calcul de la prédiction. On peut citer deux articles majeurs basés sur la factorisation matricielle [Ren12 ; Kor09a].

Koren ajoute à la définition de la factorisation matricielle présentée en 2.7 deux biais dynamiques ainsi qu'une représentation dynamique de l'utilisateur :

$$F(u,i,t) = \mu + b_u(t) + b_i(t) + \phi_u(t) \cdot \psi_i \tag{3.13}$$

Avec μ le biais général, $b_u(t)$, et $b_i(t)$ les biais utilisateur et item, tous deux dépendant du temps. Ce modèle prend pour hypothèse que les goûts de l'utilisateur évoluent au fil du temps, la représentation latente de l'utilisateur $\phi_u(t)$ devient donc fonction du temps. Ainsi un utilisateur a une représentation par pas de temps t. Dans [Kor09a], seule la faible durée de l'expérience (un mois) a permis aux auteurs de limiter l'explosion du nombre de paramètres : en la granularité temporelle est placée à l'échelle de la journée, ils ont pu limiter le nombre de représentations par utilisateurs à 30. De plus, la faible durée de l'expérience ne laissant pas supposer d'un effet de mode, la représentation des items reste la même au cours du temps.

Dans [Xio+10], les auteurs préconisent l'utilisation d'une approche Bayesienne pour la factorisation de tenseur de probabilités. Ainsi à chaque pas de temps est associé un vecteur de représentation w. La matrice de notes $\mathcal{R}$ de dimensions $|U| \times |I|$ devient donc un tenseur de dimensions $|U| \times |I| \times |W|$, ce qui permet de modéliser les utilisateurs, les items et le temps suivent un modèle de vecteurs latents probabilistes, calculé par factorisation de tenseur. La prédiction de note s'effectue donc comme il suit :

$$F(u,i,t) = \sum_{j=0}^{d} \phi_{u,j}\psi_{i,j}w_{t,j} \tag{3.14}$$

où $\phi_{\cdot,j}, \psi_{\cdot,j}, w_{\cdot,j}$ représentent la j^{eme} coordonnée des vecteurs respectifs. En factorisant toute l'information temporelle dans w, cette formulation évite la multiplication des vecteurs associés aux utilisateurs/items tels que présentés dans [Kor09a]. Dans cette approche la pondération dans l'espace latent apporte une caractérisation à valeur *sémantique* aux interactions *user-item*.

Récemment, [Zha+14a] ont proposé une autre approche bayésienne : ils utilisent des filtres de Kalman pour représenter les transitions au cours du temps entre différentes représentations latentes de l'utilisateur.

Dans [Koe+11], les auteurs implémentent des facteurs de *session* d'écoute pour caractériser le comportement de l'utilisateur dans le cadre de la recommandation musicale. Une session est constituée comme une séquence de morceaux écoutés à moins de 5 heures d'intervalle. Les auteurs ont mis en évidence une forte inertie de notation au sein d'une session.

Contextualisation Temporelle Catégorielle Contrairement aux modèles continus, les modèles catégoriels, tels que présentés dans [Ado+11], utilisent une représenta-

tion discrète du temps. Dans [Oku+06], les auteurs proposent une méthode pour prendre en considérations diverses dimensions contextuelles discrètes dans le calcul d'une machine à vecteurs de support (SVM) pour la recommandation. À chaque utilisateur correspond un hyperplan de l'espace de représentation des items séparant les *bons* et les *mauvais* produits.

Comme illustré dans Fig.3.2, une dimension de l'espace de représentation des items représente, soit une caractéristique des items (*e.g. équipement, service, spécialité pour un restaurant*), soit une dimension contextuelle (la date, par exemple). L'aspect collaboratif est obtenu en appliquant une mesure de similarité entre les utilisateurs à la recommandation. Cette similarité est calculée sur le ratio d'items en commun notés de la même manière.

Cette approche est une réponse élégante aux problèmes de division de l'ensemble d'apprentissage. En effet, le regroupement des périodes par catégorie permet de contourner le problème de multiplication des axes contextuels.

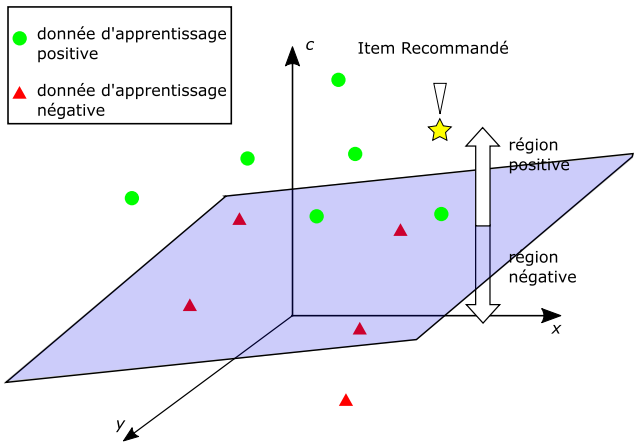


Figure 3.2: Les axes x et y correspondent aux caractéristiques des items. l'axe c lui représente le contexte (par souci de clarté,la dimension du contexte vaut 1). Le SVM permet ici l'apprentissage du plan de séparation (en mauve) entre les régions positives et négatives de l'espace. Ainsi le système est à même de recommender les items de la région positive.

Contextualisation Temporelle Adaptative Contrairement aux modèles précédents, ces modèles ne s'intéressent pas au contexte temporel au moment de la recommandation mais à l'ordre dans lesquels les interactions ont eu lieu. C'est le modèle que nous avons privilégié dans ces travaux de thèse.

Parmi les publications majeures dans ce domaine, on pourra citer les travaux de [Kar11] dans lesquels il incorpore l'information d'ordonnancement à la factorisation

matricielle. Les représentations des items sont ici dépendantes des horodatages, comme présenté dans Eq.3.15.

$$F(u,i,t) = \sum_{j=1}^{d} \phi_{u,j}\psi^{s}_{i,j}\psi^{s-1}_{i,a}\psi^{s-2}_{i,b}\cdots\psi^{s-N}_{i,N} \tag{3.15}$$

$\psi^{s}_{:,i}$ dénote la représentation de l'item i apprise en prenant en compte l'information disponible au temps s et $a, b, \cdots, N$ les items consultés aux temps $s-1, s-2, \cdots, s-N$. En plus de ce modèle multiplicatif, les auteurs ont aussi proposé un modèle additif où la prédiction de note est calculée par somme des facteurs $\phi_{u,j}\psi^{s}_{i,j}, \phi_{u,j}\psi^{s-1}_{i,j}, \cdots, \phi_{u,j}\psi^{s-N}_{i,j}$. Dans un cas comme dans l'autre la complexité spatiale du modèle est gérée manuellement en fixant N l'horizon de mémoire.

Dans [Jah+10], les auteurs proposent de découper les données en segments en fonction de leur horodatage. Pour chaque segment, il s'agit d'apprendre un modèle indépendant du temps. Ces modèles sont alors pondérés et combinés, ce qui rend le processus dépendant du temps.

De manière assez similaire, [ML13a] proposent un modèle à niveaux d'expérience, modélisant l'évolution des connaissances des utilisateurs dans un domaine. On définit pour chaque utilisateur une fonction croissante qui à chaque note horodatée associe une valeur d'expérience et à chaque niveau d'expérience correspond un problème de factorisation matricielle. Cette méthode peut être vue comme une chaîne de Markov à états cachés gauche-droite dont chaque état serait une factorisation matricielle. L'apprentissage est alors effectué de manière alternée :

— pour une répartition donnée des niveaux d'expérience, on apprend les paramètres des factorisations matricielles.

— à factorisations matricielles fixes, on optimise la répartition des niveaux d'expérience e (cf. Fig.3.3)

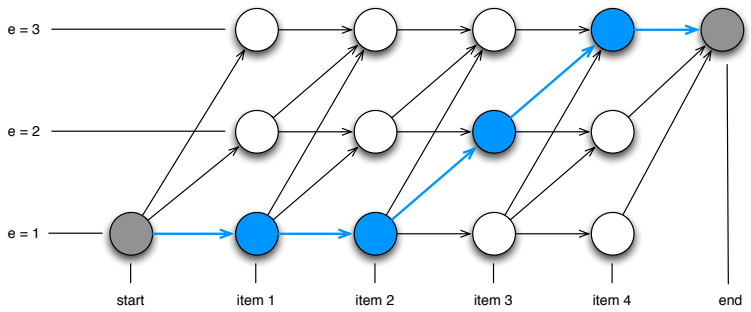


Figure 3.3: Représentation gauche-droite du problème d'optimisation des niveaux d'expérience e sur trois niveaux pour une trace de quatre items.

3.4 Recommandation et Données Textuelles

L'utilisation des données textuelles dans la recommandation est presque aussi vieille que la recommandation elle-même. En effet, comme nous l'avons vu dans la section précédente, la plupart des méthodes *content based* sont basées sur des sacs de mots ou sur l'extraction de mots-clefs pour caractériser les items.

L'utilisation conjointe des notes et de la sémantique des items (ici, le texte) est une thématique de recherche relativement récente [Dia+16]. Entre 2007 et 2011, les modèles graphiques basés sur le *topic-modeling* ont explosé, donnant lieu à de nombreuses publications comme [WB11] dans laquelle les auteurs proposent d'utiliser une allocation de Dirichlet latente pour extraire des thèmes et enrichir leur système de filtrage collaboratif. Cependant, on trouve peu d'articles dans la littérature qui proposent d'utiliser de données textuelles directement dans le système de recommandation. [Gan+09] fut un des premiers modèles hybrides proposés. Il se base sur un étiquetage manuel des phrases des avis (suivant leur catégorie et leur polarité) pour l'identification de caractéristiques des items pertinentes pour l'utilisateur. Les catégories et polarités sont ensuite utilisées pour apprendre différents classifieurs ensuite combinés pour la prédiction de notes, améliorant les performances du système tout en apportant de l'explicativité aux résultats. Dans [ML13b], en plus d'automatiser l'extraction des catégories à l'aide d'une allocation de Dirichlet latente, les auteurs améliorent l'exploitation du texte en proposant de projeter les notes et les données textuelles dans le même espace de représentation.

Dans [Zha+14b], les auteurs proposent d'extraire des mots-clefs des avis pour apporter de l'explicabilité à la recommandation. Ces mots-clés sont supposés fortement polarisés et donc permettre à l'utilisateur de comprendre pourquoi il aimera ou non un item.

Cette section est volontairement limitée car nous détaillons des contributions originales dans ce domaine dans le Chapitre 5.

3.5 Évaluation de la Recommandation

L'évaluation des systèmes de recommandation est délicate et dépend fortement de notre cadre de développement : en ligne ou hors ligne. Nous commençons par lister les modes d'évaluation généraux dans les deux cas avant d'approfondir les stratégies hors-lignes, qui sont plus accessible dans un cadre académique.

3.5.1 Évaluation hors ligne et en ligne

Comme dit précédemment, il existe deux grandes familles de méthodes d'évaluation, les méthodes en ligne (*user-based evaluation*) et les méthodes hors ligne (*system-based evaluation*) [Her+04 ; GS09 ; SG11]. Dans chacun des cas, le système de recommandation est appris à l'aide d'information sur les utilisateurs (préférences, horodatage, données démographiques) et les items (descriptions, attributs, catégories). On propose une recommandation aux utilisateurs et on enregistre leurs retours. On applique alors une métrique d'évaluation à ces retours. Cette métrique vise à évaluer une ou plusieurs propriétés du système de recommandation (précision, diversité, exploration, prédiction de notes ou encore satisfaction de l'utilisateur).

Dans le cas en ligne, les utilisateurs utilisent différents réglages du système à évaluer et, parfois, remplissent un questionnaire évaluant leur expérience après avoir reçu leurs recommandations. L'évaluation est alors effectuée soit par test A/B, c'est à dire en enregistrant et comparant les comportements des utilisateurs (*e.g.* notes, activité) suivant les différents réglages du système [Koh+09], soit en comparant les perceptions des utilisateurs récupérées grâce aux questionnaires [Kni+12], soit par combinaison de ces deux méthodes.

Dans le cas hors ligne, on utilise un jeu de données de test pour simuler les comportement qu'auraient eu les utilisateurs s'ils avaient utilisé le système de recommandation. La plupart du temps, les résultats sont évalués en comparant les notes réelles et celles prédites par le système pour chaque triplet utilisateur-item-note présent dans le jeu de test [Her+04 ; SG11]. En effet, il est impossible de savoir si un utilisateur a opté pour un produit parce que celui-ci était celui qui lui correspondait le plus ou simplement parce que le système de recommandation en place lors de la récolte des données en aura privilégié l'affichage.

L'évaluation en ligne est préférable dans la mesure où elle permet de tenir compte de l'expérience de l'utilisateur lors de son interaction avec le système de recommandation[Kni+12 ; KR12]. De plus, CREMONESI et al. [Cre+11] ont mis en avant les disparités entre les métriques d'évaluation hors ligne et la perception qu'avaient les utilisateurs de la qualité d'un système de recommandation. Divers paramètres semblent jouer sur ces disparités comme la sélection des données [Cre+11], les différences d'interface utilisateur [Cos+03], ou encore l'état d'esprit de l'utilisateur [Kni+12]. Enfin, les données peuvent être intrinsèquement biaisées par un système de recommandation préalablement installé sur le site considéré.

L'évaluation hors ligne, elle, ne nécessite que l'implémentation du modèle de recommandation à tester. C'est pourquoi, dans l'optique d'une mise en production,

il est commun de commencer par tester un modèle hors ligne avant de passer à une évaluation en ligne. On limite ainsi les risques de perte et le coût général de l'évaluation s'en trouve réduit [Koh+09 ; SG11]. cependant, il est important de noter que si elle permet d'obtenir une estimation du comportement des utilisateurs, celle-ci n'est pas exacte. En effet, elle sera fortement biaisée, que ce soit par la popularité des produits, ou par le système de recommandation utilisé lors de la récupération des données.

Dans le cas des TARS, il existe peu de travaux traitant de l'évaluation en ligne. Ceci est probablement dû au fait que le but premier de ces méthodes est l'amélioration de la précision de la recommandation. Or, quand il n'est pas toujours évident de disposer d'un système opérationnel en ligne, la précision peut être aisément mesurée à l'aide de méthodes hors ligne. Par exemple, ADOMAVICIUS et TUZHILIN [AT05] et PARK et al. [Par+07] ont engagé des utilisateurs dans le but de créer des datasets, mais ont évalué leurs modèles hors ligne. On pourra toute fois citer deux articles traitant de l'évaluation en ligne des TARS [Wen+09 ; Oku+06].

Enfin, comme nous l'avons vu précédemment, il existe une dernière différence fondamentale entre les systèmes d'évaluation en ligne et hors lignes. Les systèmes en ligne utilisent toutes les données disponibles au moment de la recommandation alors que les systèmes hors ligne se basent sur un découpage arbitraire des données. Nous allons maintenant nous intéresser à ces stratégies de découpage.

3.5.2 Méthodologies d'Évaluation hors-ligne

Si certaines étapes de l'évaluation d'un système de recommandation sont communes à toutes les méthodes, d'autres sont sujettes à des variations dans leur implémentation, ce qui mène à des différences méthodologiques significatives. Le processus de répartition des données entre apprentissage et test est par exemple une source majeure de déviation des méthodologies. Cet effet est d'autant plus flagrant quand on considère l'horodatage des notes, comme c'est le cas pour les TARS [GS09]. On trouve ainsi dans la littérature pléthore d'implémentations de la méthode *hold-out* [Dud+00], méthode largement utilisée pour le découpage des données entre apprentissage et test [GS09].

Lors de l'élaboration d'une méthodologie d'évaluation, une des premières question à se poser est de savoir si les notes doivent être préalablement séparées suivant un critère d'ordonnancement ou non. On pourra par exemple vouloir considérer un ordonnancement temporel des notes où toutes les notes seront préalablement triées suivant leur horodatage. Ensuite, on pourra considérer que l'ensemble d'apprentissage contiendra toutes les notes émises avant une date donnée alors que l'ensemble

de test contiendra le reste des notes [Pan+09]. À l'opposé, on pourra décider de s'affranchir de la temporalité et tirer les exemples de test au hasard [Sto07]. Il existe bien entendu bien des nuances entre ces deux extrêmes, comme par exemple considérer un pourcentage fixe de la ligne temporelle de chaque utilisateur pour la séparation, comme c'était le cas pour la compétition Netflix [Ben+07].

De plus, il est important de noter que l'ordonnancement des notes peut avoir différentes granularités. En effet, le découpage peut être effectué sur l'ensemble de la matrice de notes M [Kar11] ou être créé de manière indépendante pour chaque utilisateur [Koe+11].

Le nombre de notes considérées pour le test est une autre source de divergence de méthodologies au sein de la communauté des TARS. On pourra par exemple utiliser le schéma classique proportionnel (*e.g.* 80% des données sont utilisées pour l'apprentissage et les 20 derniers pourcents sont utilisés pour l'évaluation) [Sin+10], sélectionner arbitrairement un nombre fixe de notes par utilisateur [Din+06] ou enfin, baser le découpage sur une date absolue, comme expliqué précédemment [Lat+09].

En plus des problèmes de répartition des données se pose celui de la validation croisée. Ces méthodes visent à améliorer la capacité de généralisation de l'évaluation sur des jeux de données indépendants. Dans les méthodes les plus populaire, on pourra citer celles de ré-échantillonnage comme *X-fold cross validation*. Cette technique, qui consiste en divers ré-échantillonnages successifs des données, permet de moyenner les résultats sur divers jeux de test tous extraits de M [AT05].

Le besoin de répondre à des tâches de recommandations aussi diverses que spécifiques est une troisième source de différenciations entre les méthodologies. Par exemple pour s'atteler au problème du *Top-N*, il est nécessaire de définir précisément un ensemble d'items cible que le système de recommandation doit ordonner. BELLOGÍN et al. [Bel+11] décrivent dans leur article différentes approches pour générer ledit ensemble. Ils proposent par exemple de ne considérer que les items dont l'intérêt peut être quantifié [AT05], ou encore de mélanger des items pertinents (*i.e.* dont les notes sont élevées) avec des items dont l'intérêt reste à démontrer (*i.e.* sans notes) [Cre+10].

Toujours dans le cadre de la tâche de recommandation en *Top-N*, il existe de nombreuses façons de considérer le concept de pertinence des items. On pourra considérer que tous les items notés par l'utilisateur sont pertinents à ses yeux (on ne considère alors pas le principe de déception associé à une mauvaise note) ou encore décider retirer certains items, considérant à posteriori qu'ils ne présentaient que peu d'intérêt aux yeux de l'utilisateur (*e.g.* un produit ayant reçu la note minimale ou

une chanson écoutée partiellement ou même une seule fois). Il est aussi possible rapprocher le traitement de ces informations des problématiques d'*implicit feedback* [PA11].

3.5.3 Métriques d'Évaluation

Au fil des publications, de nombreuses méthodes d'évaluations ont été proposées pour évaluer et comparer les algorithmes de recommandation ; chacune de ces méthodes se focalisant sur l'estimation d'une propriété des recommandations générées par le système [GS09]. Dans la littérature, la plupart des méthodes publiées se focalisent sur la précision de la prédiction de notes. On pourra citer deux métriques se focalisant sur la capacité du système à prédire efficacement la note associée à un couple utilisateur-item : l'erreur moyenne absolue (MAE) et l'erreur-type (RMSE), définies comme suit :

Soit $\mathcal{T} = \{(r_{u,i}, u, i)\}$ un ensemble de données observées et $\hat{r}_{(u,i)}$ la prédiction du modèle étudié pour le couple (u, i),

$$\text{MAE} = \sum_{(u,i)\in\mathcal{T}} \frac{\left|r_{(u,i)} - \hat{r}_{(u,i)}\right|}{n} \quad \text{RMSE} = \left(\frac{\sum_{(u,i)\in\mathcal{T}} \left(r_{(u,i)} - \hat{r}_{(u,i)}\right)^2}{n}\right)^{\frac{1}{2}} \tag{3.16}$$

Depuis quelques années, l'utilisation de la prédiction de notes fait polémiques et certains préconisent l'utilisation de métriques de précision d'ordonnancement. En effet, le but d'un système de recommandation reste de fournir un nombre limité d'items qui, supposément, plairont à l'utilisateur [KR12]. En général, ces méthodes consistent en l'ordonnancement des items suivant leurs notes prédites. Le système recommande alors meilleurs N items (*Top-N*). Les métriques de précision d'ordonnancement mesurent alors la quantité d'items pertinents contenue dans la recommandation [Her+04]. Ces métriques sont souvent issues des protocoles d'évaluation de recherche d'information, comme le rappel et la précision [BYRN99] ou encore de l'apprentissage statistique, comme la fonction d'efficacité du récepteur (*courbe ROC*) ou encore l'aire sous la courbe ROC (*AUC*) [Lin+03]. La mesure ROC est définie sous la forme d'une courbe qui donne le taux de vrais positifs (fraction des positifs qui sont effectivement détectés) en fonction du taux de faux positifs (fraction des négatifs qui sont détectés (incorrectement)) :

— À (0, 0) le classificateur déclare toujours *négatif* : il n'y a aucun faux positif, mais également aucun vrai positif. Les proportions de vrais et faux négatifs dépendent de la population sous-jacente.

— À (1, 1) le classificateur déclare toujours *positif* il n'y a aucun vrai négatif, mais également aucun faux négatif. Les proportions de vrais et faux positifs dépendent de la population sous-jacente.
— Un classificateur aléatoire tracera une droite allant de (0, 0) à (1, 1).
— À (0, 1) le classificateur n'a aucun faux positif ni aucun faux négatif, et est par conséquent parfaitement exact, ne se trompant jamais.
— À (1, 0) le classificateur n'a aucun vrai négatif ni aucun vrai positif, et est par conséquent parfaitement inexact, se trompant toujours. Il suffit d'inverser sa prédiction pour en faire un classificateur parfaitement exact.

Il est important de garder en mémoire que les métriques de précision de la prédiction de notes et celles de précision d'ordonnancement sont utilisées pour évaluer deux tâches différentes, nommément la prédiction de notes et la recommandation *Top-N*.

Récemment, de nouvelles méthodes ont exploré d'autres chemins que l'évaluation de la précision (aussi bien de la prédiction de notes que de l'ordonnancement). Dans ces articles, la lumière est mise sur d'autres aspects de la recommandation, comme son explicabilité [Pou+14] ou la propension du modèle à l'inclusion de nouveaux items et à la diversification de la recommandation [VC11]. Ces nouvelles facettes peuvent être quantifiées à l'aide de métriques comme *Self Information* [Zho+10] et *Intra List Similarity* [Zie+05]. Les métriques s'attelant à la nouveauté visent à capturer la capacité du système à recommander (aussi bien au niveau de l'utilisateur que de la communauté en général) des items jusque là inconnus. Les métriques mesurant la diversité vont quant à elles quantifier la similarité entre les items recommandés par le système.

3.6 Conclusion

Comme nous avons pu le voir au cours de cet état de l'art, le domaine de la recommandation a connu de nombreuses évolutions au cours de la dernière décennie et ce aussi bien au niveau des méthodes mises en œuvre que de la définition même de la tâche. ce phénomène évolutif a été encore accéléré ces dernières années par l'apparition de l'internet nomade (données de géolocalisation) et l'inter-connectivité des applications. Cette thèse s'est déroulée à une époque charnière du domaine, durant laquelle la prise en compte du contexte est devenue incontournable et de nouveaux paradigmes ont vu le jour, répondant à une volonté d'ajouter une plus-value à la recommandation. On citera par exemple l'apparition des *healthy recommender systems* ou des modèles exploratoires [Kap+15].

les travaux qui vont vous être présentés dans la suite de ce manuscrit s'inscrivent dans une volonté d'accompagnement de l'utilisateur et d'explicabilité des résultats.

4 Extraction et Exploitation de Marqueurs d'Opinion

> *Je ne suis ni pour, ni contre, bien au contraire.*
>
> **— Coluche**

4.1 Introduction

Dans ce chapitre, nous étudions les traitements usuels associés aux données textuelles, notamment sur les aspects touchant à l'analyse d'opinion. Nous présentons une méthode d'extraction de caractéristiques textuelles à partir de données récupérées sur Twitter [1] ainsi qu'une méthode de transfert explicite trans-media depuis les revues d'utilisateurs vers les tweets.

En recueillant les messages associés aux produits et en lançant une analyse de sentiments sur ces corpus, nous définissons ainsi des marqueurs de sentiment (positif/négatif) [Pan+02] ou d'objectivité [Liu10] permettant de modéliser efficacement l'avis des utilisateurs sur les items. Pour valider l'intérêt de ces caractéristiques, nous proposons dans un second temps de les appliquer à la prédiction de résultats au box office [AH10 ; Del+07].

Extraction de Caractéristiques Ayant décidé de nous placer dans un paradigme d'apprentissage supervisé, il était nécessaire d'avoir des jeux de données étiquetées pour apprendre notre classifieur. Or, Twitter a beau être une source intarissable de données textuelles, elles ne sont cependant pas étiquetées et le coût prohibitif de l'étiquetage manuel d'une quantité de tweets suffisante à l'apprentissage d'un modèle d'analyse de sentiments complique grandement son application aux données de micro-blogging. En parallèle, il existe sur le web de nombreuses sources de données textuelles étiquetées : les avis postés par les utilisateurs sur les sites spécialisés (*e.g.* films : IMDb [2], bières : RateBeer [3]) et les boutiques en ligne (*e.g.* Amazon [4]). Une

1. http ://www.twitter.com
2. http ://www.imdb.com
3. http ://www.ratebeer.com
4. http ://www.amazon.com

solution consiste donc à apprendre nos modèles sur des avis d'utilisateurs pour ensuite les appliquer à Twitter. Toutefois, le problème de transfert est déjà reconnu comme complexe à l'échelle des domaines (*e.g.* du cinéma vers l'électroménager) [Bli+06]. Or, il se complexifie encore lorsque l'on veut passer d'un medium à l'autre. Il existe des différences structurelles majeures entre les avis utilisateur et les tweets : les avis sont des textes écrits plutôt correctement, pouvant exprimer plusieurs sentiments (lister des avantages et des inconvénients, par exemple) alors que les tweets sont des messages courts (moins de 140 caractères), fortement polarisés, contenant de nombreuses abréviations et emojis [PP10].

Dans [MS12], les auteurs ont montré la pertinence de l'utilisation des avis utilisateurs pour la classification de tweets, ainsi que l'importance du volume de données d'apprentissage. Dans la suite, nous proposons d'utiliser diverses sources annotées, nommément Amazon [Bli+06 ; JL07], DBLP [Maa+11] et TripAdvisor [Wan+07] pour apprendre notre modèle de transfert que nous évaluons ensuite sur un jeu de données Twitter manuellement étiqueté[Che+12]. Nous avons considéré différentes méthodes de transfert en commençant par la plus simple, un transfert direct sans adaptation, avant de nous intéresser à des méthodes de transfert explicite [Bli+06 ; DI09].

Évaluation des Caractéristiques A l'aide du jeu de données Twitter proposé dans [Che+12], contenant l'ensemble des tweets anglophones associés à 32 films sur une période de six mois , nous proposons de mesurer l'impact de différentes caractéristiques de sentiments sur la précision de la prédiction du nombre d'entrées (exprimé en terme de chiffre d'affaire). Concrètement, nous utilisons les caractéristiques de sentiment extraites précédemment et optimisons un problème de régression. Si cette problématique n'est pas une tâche de recommandation, il s'agit néanmoins de quantifier l'apport des données textuelles sur une tâche de prédiction d'opinion de plus haut niveau. Nous démontrons ainsi l'intérêt que présente l'ajout de cette nouvelle source d'information aux méthodes volumétriques classiques (basées sur le comptage de tweets et d'articles de blogs).

La suite de ce chapitre s'articule comme suit : la section 4.2 abordera les modèles que nous avons proposé pour le transfert trans-média, la section 4.3 traitera du modèle de prédiction. Enfin, nous conclurons dans la section 4.4.

4.2 Classification Supervisée d'Opinion

Cette section se divise en trois parties, nous commencerons par décrire le modèle de base que nous avons choisi pour l'analyse de sentiments puis les méthodes de transfert que nous avons mises en place. Enfin, nous aborderons leur évaluation.

4.2.1 Modèle de base

La classification trans-media est une tâche complexe. Ceci est principalement du aux différences lexicales majeures entre Twitter et les revues d'utilisateurs. Il est donc primordial de privilégier les capacités de généralisation du modèle plutôt que sa spécialisation, c'est pourquoi dans ce chapitre, les données textuelles sont représentées à l'aide d'unigrammes (sac de mots classique), couplés à un codage présentiel [Pan+02]. Ainsi, chaque document devient un vecteur $\mathbf{x} \in \mathbb{R}^d$ où $x_j \in \{0, 1\}$ et d représente la taille du dictionnaire. De plus, nous limitons notre dictionnaire aux 5000 mots les plus fréquents, comme préconisé dans [Bli+06] (on a donc $d = 5000$). Les notes sont codées par des étiquettes $y \in \{-1, 1\}$: les notes 0 et 1 sont associées à une polarité négative, représentée par $y = -1$ et les notes 4 et 5 sont associées à une polarité positive, représentée par $y = 1$. les documents associés à une note de 3 étant considérés comme ambigus (d'un utilisateur à l'autre, ils peuvent être considérés comme positifs, négatifs ou neutres), ils sont supprimés.

La première approche que nous avons proposée s'inscrit dans la lignée des travaux de [MS12]. Elle consiste en l'apprentissage d'un classifieur de sentiment classique, mais appris sur de gros corpus étiquetés, recouvrant divers domaines. Nous cherchons par ce biais à maximiser les chances de découvrir de nouvelles expressions et ainsi pallier au problèmes d'écarts sémantiques [Bes+11].

Pour leur efficacité et leur capacité de passage à l'échelle, nous avons privilégié l'utilisation des machines à vecteurs de support linéaires (SVM) [Joa02]. Le classifieur est représenté par un vecteur $\mathbf{w} \in \mathbb{R}^d$, avec $d = 5000$ et la fonction de décision, permettant de définir la polarité d'un document $\mathbf{x}$, est définie comme suit :

$$f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle = \sum_{j=1}^{d} x_j w_j \tag{4.1}$$

Si l'on considère un corpus de N documents étiquetés, le SVM est alors appris par optimisation du problème suivant :

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - f(\mathbf{x}_i) y_i) \tag{4.2}$$

Il est intéressant de noter que l'utilisation d'un SVM linéaire avec un codage binaire des documents apporte une certaine explicativité aux résultats. En effet, chaque coefficient w_j du vecteur w ainsi appris est associé à un mot j du dictionnaire. Ainsi la valeur de w_j représente la polarité générale de ce mot : si elle est positive (resp. négative) alors chaque document qui contient ce mot se verra déplacé vers la classe 1 (resp. -1) et plus cette valeur sera grande, plus ce mot aura d'influence sur la classification des documents.

4.2.2 Transfert

Pour améliorer les capacités de d'extraction de notre modèle sur le corpus twitter, nous proposons de lui apposer des méthodes de transfert explicite. Nous en avons retenu deux, *Structural Correspondence Learning* [Bli+06], ou SCL, et *Frustratingly Easy Domain Adaptation* [DI09], ou FEDA, qui présente l'avantage d'être rapide et de passer facilement à l'échelle. Ces méthodes sont habituellement utilisées pour le transfert trans-domaine (*e.g* apprendre sur un corpus de films pour classifier de l'électroménager), mais ici, nous les utiliserons dans un contexte trans-media, c'est à dire que nous allons apprendre un modèle sur des revues d'utilisateurs et l'adapter à des tweets.

Structural Correspondence Learning (SCL) [Bli+06] SCL est un processus en trois étapes : tout d'abord, on extrait np mots pivots, c'est à dire des mots qui gardent supposément le même comportement dans les domaines source et cible. Ensuite, on apprend, np classifieurs $\mathbf{w} \in \mathbb{R}^d$ (avec d la taille du dictionnaire) visant à prédire la présence ou non d'un pivot dans un document. Enfin, les vecteurs $\mathbf{w}$ sont concaténés dans une matrice de taille $np \times d$. On applique une méthode de factorisation matricielle à cette matrice (SVD) pour construire une matrice de projection $\Theta \in \mathbb{R}^{h \times d}$ avec $h < np$. Chaque document est alors décrit par son sac de mots originel et

les caractéristiques de transfert obtenues par projections de la représentation du document sur Θ. L'algorithme d'apprentissage est décrit dans Algo.5

Données : Données étiquetées dans la source $\{(x_t, y, t)|t \in [1, T]\}$
Données non étiquetées dans la source et la cible $\{x_j\}$
Résultat : Fonction de prédiction $f : X \to Y$
Choisir np pivots.
Créer np problèmes de prédiction, $p_l(x), x \in [1, m]$
pour $l \in [1, m]$ **faire**
$\quad w_l = \underset{w}{\text{argmin}} \left(\sum_j L(w \cdot x_j, p_l(x_j))\right)$
fin
$W = [w_1, \cdots, w_m]$
$\left[U, D, V^T\right] = SVD(W)$
$\Theta = U^T_{1:h,:}$
Apprendre un prédicteur f sur l'ensemble : $\left\{\left(\left[\begin{pmatrix} x_t \\ \Theta x_t \end{pmatrix}\right], y_t\right)_{t=1}^{T}\right\}$

Algorithme 5 : Algorithme SCL.

Ici, nous avons sélectionné comme pivots les mots les plus fréquents dans les données sources et le corpus de validation [San11]. Comme nous cherchons à nous concentrer sur les marqueurs de sentiments, nous limitons donc la recherche des mots pivots aux mots appartenant au champ lexical des sentiments [HL04]. Au cours de la phase d'expérimentation, nous avons fixé $np = 100$, ainsi, nous obtenons 100 nouvelles caractéristiques pour décrire chaque documents.

Frustratingly Easy Domain Adaptation (FEDA) [DI09] Cette méthode se base sur l'idée d'augmenter la représentation des documents en y ajoutant $n + 1$ fois la représentation du dictionnaire, une pour chacune des n sources et une pour la cible (on en ajoute donc $n+2$, en comptant la représentation de base). En d'autres termes, si j'ai 4 sources de données pour mon apprentissage et un jeu de données cible, j'obtiendrais donc un dictionnaire 6 fois plus grand.

Ici, nous utiliserons trois réplications du dictionnaire, une générale, une pour les revues d'utilisateurs (*i.e* notre *source*) et une pour les tweets (*i.e.* notre *cible*). La représentation d'une revue $\mathbf{x}$ devient donc $\mathbf{x}_e = [\mathbf{x}\ \mathbf{x}\ 0]$ alors qu'un tweet est représenté par $\mathbf{x}_e = [\mathbf{x}\ 0\ \mathbf{x}]$. Cette approche, tout en restant très efficace est facile à implémenter. Son intérêt réside dans sa capacité d'équilibrage automatique : même si le nombre de retours utilisateur est beaucoup plus élevé que le nombre de tweets, l'algorithme reste capable d'extraire efficacement l'information discriminantes des deux sources d'apprentissage. Cependant, il est important de noter qu'elle demande

d'avoir une certaine quantité de données étiquetées dans le domaine cible (ici, les tweets).

Une fois les représentations apprises, on utilise un SVM pour apprendre le classifieur sur les représentations $\{(\mathbf{x}_{e,i}, y_i)\}_{i=1,\dots,N}$. L'inférence étant effectuée sur des Tweets, les éléments sont donc tous de la forme $\mathbf{x}_e = [\mathbf{x}\ 0\ \mathbf{x}]$.

4.2.3 Évaluation

Comme expliqué précédemment, dans ce chapitre nous nous intéressons à l'adaptation trans-media, c'est à dire l'adaptation d'un modèle appris sur un certain type de données, ici des retours utilisateurs pour classifier des données d'un autre type, ici, des tweets. Une fois les jeux de données utilisés présentés, nous comparerons les capacités des trois modèles décrits précédemment (SVM, SCL et FEDA) appris sur différents ensembles d'apprentissages.

Jeux de données & Modèles Pour initialiser nos modèles d'adaptation trans-média, nous proposons (cf. Table 4.1) l'utilisation de divers jeux de données de retours utilisateur ainsi que deux corpus de tweets étiquetés manuellement (en gras dans la Table 4.1). On remarque que ces deux derniers datasets présentent beaucoup moins de données que les précédents. Ceci est du au coût de l'étiquetage manuel. Les modèles sont ensuite testés sur un ensemble de tweets étiquetés : le dataset Golden Standard[Che+12]. Il est important de noter ce que jeu de données est très déséquilibré (82% de tweets positifs). Nous avons donc pris ce biais en compte au moment de l'apprentissage de tous nos modèles.

Jeu de données source	**Taille**
Amazon DVD [Bli+06]	10k
Amazon Books [Bli+06]	10k
Amazon Kitchen [Bli+06]	10k
Amazon Electronics [Bli+06]	10k
ACL IMDb [Maa+11]	50k
Trip Advisor [Wan+10]	50k
Amazon Huge DVD [JL08]	450k
Amazon Huge Books [JL08]	1.9M
Amazon Huge Kitchen [JL08]	70k
Amazon Huge Electronics [JL08]	140k
Twitter Sanders [San11]	1081
Jeu de données cible	**Taille**
Golden Standard [Che+12]	251

Table 4.1: Description des jeux de données. Le jeu Twitter Sanders est utilisé pour l'apprentissage de FEDA comme cible virtuelle.

Source(s)	Précision sur la cible
Amazon DVD	82.47%
Amazon Books	82.87%
Amazon Kitchen	82.07%
Amazon Electronics	82.07%
Amazon all	82.87%
ACL IMDb	84.46%
Trip Advisor	75.7%
ACL IMDb & Trip Advisor	84.46 %
Amazon Huge DVD	84.06%
Amazon Huge Books	83.27%
Amazon Huge Kitchen	83.67%
Amazon Huge Electronics	84.46%
Twitter Sanders	64.14%
All	**87.25%**

Table 4.2: Scores en précision sur le Golden Standard [Che+12] en fonction des sources considérées pour l'apprentissage d'un SVM classique.

Nous utilisons les trois modèles décrits précédemment pour calculer le score en sentiment de chaque tweet : un modèle sans transfert explicite, qui nous sert de modèle de référence, FEDA et SCL.

Modèles de Référence Nos modèles de référence consistent en un SVM appris, sans transfert explicite, en mono et en multi sources. Les résultats de ces méthodes sont disponibles dans la Table 4.2. Lors de l'étude de ces résultats, nous pouvons formuler diverses remarques, tout d'abord notre meilleur score en précision est de 87,5%, ce qui est significativement meilleur que la répartition des données dans le jeu de test (82% de positifs), les retours utilisateur sont donc une source viable pour l'apprentissage de nos modèles. De plus, ce score est atteint lors de l'utilisation de tous les jeux de données disponibles comme sources pour l'apprentissage, ce qui corrobore les résultats présentés dans [MS12]. Il est aussi intéressant de remarquer que le modèle appris sur Twitter Sanders fait montre de mauvaises performances, ceci peut s'expliquer par deux facteurs : le manque de données d'apprentissage d'une part, et l'écart entre les domaines traités par le jeu d'apprentissage (électronique) et celui de test (cinéma). L'importance du domaine est aussi illustré par les performances des modèles appris sur IMDb et Amazon Huge DVD qui présentent les meilleurs résultats pour les modèles mono source.

On remarque aussi qu'Amazon Huge présente de meilleurs résultats qu'Amazon et ce indépendamment du domaine considéré. Pour étudier plus précisément l'influence de la taille du jeu de données d'apprentissage sur les résultats, nous avons créé un nouvel ensemble d'apprentissage contenant tous les documents de toutes les sources et avons ensuite réalisé une nouvelle série d'expériences : pour différentes fractions de cet ensemble, nous apprenons un modèle et l'évaluons en terme de précision sur le Golden Standard. Les résultats, présentés dans la Figure 4.1 montrent clairement

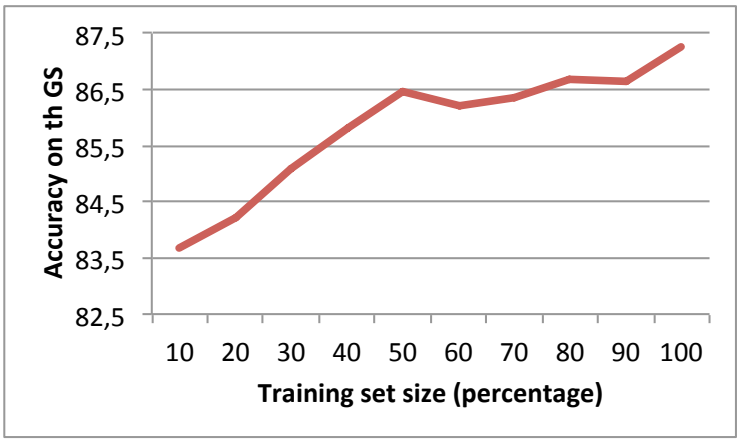


Figure 4.1: Évolution de la précision sur le Golden Standard en fonction du pourcentage de données utilisées en apprentissage (toutes sources fusionnées). Pour garantir une fiabilité des résultats, chaque expérience est réalisée cinq fois, sur des échantillon différents (sélection aléatoire).

une influence quasi linéaire de la taille de l'ensemble d'apprentissage sur les résultats. Toutefois, ce phénomène s'estompe passé une certaine taille.

4.2.4 Adaptation Explicite

Nous nous intéressons maintenant à l'adaptation trans-média explicite. Nous avons utilisé les modèles FEDA et SCL dans les mêmes conditions que pour les modèles de base. FEDA nécessite d'ajouter aux données d'apprentissage quelques données du domaine-cible étiquetées. Aux vues de la faible taille du Golden Standard, nous avons du utilisé, au cours de l'apprentissage le jeu Twitter Sanders comme échantillon de la cible, ce qui peut avoir biaisé les performances. Les résultats sont présentés dans la Table 4.3.

On remarque que les méthodes d'adaptation explicite présentent un léger gain de performances par rapport au modèle de référence et ce, malgré l'approximation sur l'échantillon cible. Cependant ce gain est minime comparé à celui octroyé par l'utilisation de toutes les données. Il semble donc plus intéressant d'utiliser de grandes bases d'apprentissage que d'utiliser des modèles d'adaptation explicite complexes.

Learning Dataset	FEDA	SCL
Amazon all	81.67%	82.47%
ACL IMDb	82.87%	**86.06%**
Trip Advisor	75.3%	73.71%
Amazon Huge All	**87.85%**	**87.06%**
All	**87.45%**	**87.06%**

Table 4.3: Scores en précision sur le Golden Standard [Che+12] pour les modèles à transfert explicite (FEDA et SCL).

4.3 Descripteurs d'Opinion et Prédiction de Haut Niveau

Dans un second temps nous nous sommes intéressés à une tâche d'apprentissage par supervision distante : nous avons utilisé les classifieurs appris dans la section précédente pour fournir des caractéristiques de sentiments à un modèle prédictif. Dans le cadre de ces travaux, nous avons considéré la tâche de prédiction de résultats au box office à partir de tweets. Ce procédé est effectué en deux temps : dans un premier temps, nous calculons d_r caractéristiques numériques pour chaque film et construisons ainsi, pour chaque film, un vecteur $\mathbf{x}_r$. Ensuite, chaque film est associé à son résultat au box office y_r (*i.e.* étiqueté) et nous apprenons un prédicteur linéaire $f(\mathbf{x}_r) = \langle \mathbf{x}_r, \mathbf{w}_r \rangle$ qui approxime y_r. L'apprentissage est effectué par minimisation du critère des moindres carrés régularisé L2 (*ridge regression*) :

$$\mathbf{w}_r^\star = \arg\min_{\mathbf{w}_r} \sum_{i=1}^{N_f} (f(\mathbf{x}_{r,i}) - y_{r,i})^2 + \lambda \|\mathbf{w}_r\|^2, N_f = 32 \tag{4.3}$$

Pour chaque film f, nous calculons les caractéristiques suivantes :

— *volume* (vol) : le nombre de tweets parlant de ce film ;

— *polarité moyenne* (aps) : la moyenne du score en polarité sur l'ensemble des tweets parlant de ce film ;

— *volume positif* (pv) : le volume de tweets positifs associés à ce film ;

— *volume négatif* (nv) : $vol - pv$ (nos classifieurs sont biaires).

Contrairement à `vol` qui ne comprend qu'une caractéristique, `aps`, `pv` et `nv` sont calculés pour chaque modèle de sentiment, soit 27 modèles. En effet nous avons un modèle sans transfert explicite et deux modèles avec transfert explicite (SCL et FEDA). Ces modèles sont appris sur 9 jeux de données différents décrits en détail dans la table Tab4.5. Nous avons ainsi créé un total de 82 caractéristiques. De part leur nature, ces caractéristiques sont très fortement corrélées, ce qui empêche un apprentissage efficace du prédicteur. Nous avons donc du opérer une sélection de variable. Elle est abordée plus en détail dans la section 4.3.2.

La solution analytique au problème de régularisation posé en Eq. 4.3 est calculée suivant $\mathbf{w}_r^\star = (X_r^T X_r + \lambda I)^{-1} X_r^T Y_r$, avec $X_r = \begin{bmatrix} \mathbf{x}_{r,1} \\ \dots \\ \mathbf{x}_{r,32} \end{bmatrix}$. Nous montrerons par la suite que ce problème est mal posé, aux vues de la forte corrélation entre les variables. Les modèles de sentiments étant très similaires, ils génèrent des caractéris-

tiques fortement corrélées. C'est pourquoi nous proposons une sélection de variable drastique ainsi qu'une stabilisation numérique dans le processus d'optimisation.

4.3.1 Jeux de Données et Génération de Marqueurs d'Opinion

Dans ces travaux, nous avons utilisé le jeu de données twitter [Che+12]. Une fois les tweets récupérés [5], le jeu de données comptait 168032 tweets traitant de 32 films. La distribution volumétrique est disponible dans la Table 4.4. Le résultat au box office, qui provient du site Box Office Mojo [6], est exprimé en dollars.

Titre	nombre de tweets	Box Office
Edge Of Darkness	3910	43313890
When In Rome	3271	32680633
Tooth Fairy	3111	60022256
Book Of Eli	5845	94835059
Legion	4863	40168080
Extraordinary Measures	799	12068313
Spy Next Door	1934	24307086
To Save A Life	922	3777210
Preacher's Kid	483	515065
Dear John	11229	80014842
From Paris With Love	3137	24077427
Valentine's Day	5335	110485654
Wolfman	3455	61979680
Shutter Island	20229	128012934
Cop Out	4628	44875481
Crazies	2602	39123589
Ghost Writer	2665	15541549
Alice In Wonderland	29112	334191110
Diary Of A Wimpy Kid	1211	64003625
Bounty Hunter	5968	67061228
She's Out Of My League	2474	2010860
Our Family Wedding	1073	20255281
How To Train Your Dragon	5728	217581231
Back Up Plan	955	37490007
Date Night	9041	98711404
Death At A Funeral	3232	42739347
Clash Of The Titans	10547	163214888
Last Song	5702	62950384
Iron Man 2	7075	312433331
My Name Is Khan	4941	4018771
Brooklyn's Finest	2	27163593
Shrek Forever After	2549	238736787

Table 4.4: Description des films inclus dans le jeu de données [Che+12] (nombre de tweets associés et résultat au box office, exprimé en dollars)

5. la politique de confidentialité de Twitter interdit le partage des tweets en clair. Il a donc fallu récupérer chaque tweet à partir de son identifiant. Malheureusement entre la publication du jeu de données et notre utilisation, certains tweets ont été effacés.

6. http ://www.boxofficemojo.com

Nous utilisons pour générer nos caractéristiques les trois modèles abordés dans la section précédente, sur l'ensemble des jeux de données décrits précédemment. Nous générons ainsi 82 caractéristiques à l'aide de 27 modèles différents. Le détail de ces 27 instanciation est disponible en Table 4.5. Notons que les caractéristiques 15, 16 et 17 se différencient des autres : la caractéristique 15 exclusivement apprise sur des tweets, et les caractéristiques 16 et 17 s'intéressent à la subjectivité (*i.e.* l'auteur donne-t-il un avis, qu'il positif ou négatif, ou garde-t-il un ton neutre ?) plutôt qu'à la polarité des textes observés. Pour la caractéristique 16 nous utilisons [PL04], qui contient 5000 phrases étiquetées manuellement comme objectives ou subjectives et pour la caractéristique 17 nous considérons les tweets étiquetés comme neutres dans le Golden Standard comme objectifs, et les autres comme subjectifs.

ID	Source	Nombre d'Exemples d'Apprentissage	Algorithme
1	IMDB [Maa+11]	50k	SVM
2	TripAdvisor [Wan+10]	50k	SVM
3	IMDB+TripAdvisor	100k	SVM
4	Amazon (books) [Bli+06]	10k	SVM
5	Amazon (dvd) [Bli+06]	10k	SVM
6	Amazon (electronics) [Bli+06]	10k	SVM
7	Amazon (kitchen) [Bli+06]	10k	SVM
8	Amazon (full) [Bli+06]	40k	SVM
9	Huge Amazon (books) [JL08]	1.9M	SVM
10	Huge Amazon (dvd) [JL08]	450k	SVM
11	Huge Amazon (electronics) [JL08]	140k	SVM
12	Huge Amazon (kitchen) [JL08]	70k	SVM
13	Huge Amazon (full) [JL08]	2.5M	SVM
14	All	2.8M	SVM
15	Twitter Sanders [San11]	1081	SVM
16	IMDB Subj/Obj [PL04]	5000	SVM
17	GS Subj/Obj [Che+12]	754	SVM
18	IMDB [Maa+11]	50k	SCL
19	TripAdvisor [Wan+10]	50k	SCL
20	Amazon [Bli+06]	40k	SCL
21	Huge Amazon [JL08]	2.5M	SCL
22	All	2.8M	SCL
23	IMDB [Maa+11]	50k	FEDA
24	TripAdvisor [Wan+10]	50k	FEDA
25	Amazon [Bli+06]	40k	FEDA
26	Huge Amazon [JL08]	2.5M	FEDA
27	All	2.8M	FEDA

Table 4.5: Caractéristiques et initialisations associées (ensemble d'apprentissage et algorithme utilisé).

4.3.2 Sélection de Variables et Régularisation

Le Golden Standard[Che+12] ne contenant que 32 films, il ne permettait pas une découpe en apprentissage, validation et test directe, c'est pourquoi nous avons choisi un protocole *leave-one-out* (LOO).

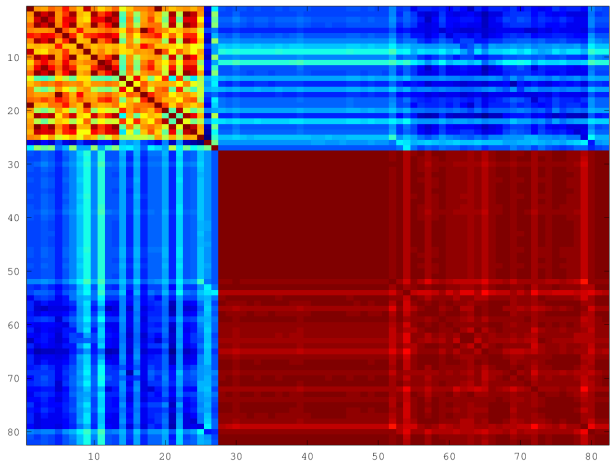


Figure 4.2: Corrélation entre les différentes caractéristiques pour le problème de prédiction de box office. On retrouve trois blocs de 27 caractéristiques (correspondant à `aps`, `pv` et `nv` pour les 27 modèles) et une caractéristique volumétrique.

Dans notre problème d'optimisation, le nombre de caractéristiques associées à chaque film ($d_r = 82$) excède le nombre de films ($N_f = 32$). De plus, les variables sont fortement corrélées (cf. Fig.4.2) : pour chaque film, les 27 modèles donnent des valeurs de `aps`, `pv` et `nv` très similaires.

Pour résoudre le problème, nous avons proposé une procédure de sélection de caractéristiques gloutonne pas à pas :

Données : Un ensemble de caractéristiques Car
Résultat : Un ensemble de caractéristiques retenues Res
$Res \leftarrow \emptyset$
tant que $Car \neq \emptyset$ **faire**
    On évalue tous les sous ensembles de caractéristiques $Res \cup c, c \in Car$
    On choisit la caractéristique c qui présente le plus grand gain de performances
    si c *propose un gain positif* **alors**
        $Res \leftarrow c$
        $Car \leftarrow Car \backslash c$
    fin
    sinon
        $Car \leftarrow \emptyset$
    fin
fin

Si cette procédure nous permet de sélectionner le sous-ensemble de variables le plus pertinent, il faut toutefois noter qu'un réel risque de sur-apprentissage subsiste. En effet, la taille réduite du jeu de données nous empêche d'optimiser notre sélection sur un ensemble vierge. Nous admettons ainsi que les performances présentées par la suites sont probablement légèrement surestimées. Le paramètre d'équilibrage λ est optimisé par recherche linéaire.

4.3.3 Évaluation et Résultats

Dans la littérature, la prédiction de résultats au box-office s'effectue généralement à l'aide de caractéristiques dites *expertes* (*i.e.* le budget alloué au film, la notoriété des acteurs, le nombre de salles de diffusion) [Del+07 ; Sht04 ; Tum+10] et les tweets ne sont utilisés que pour une estimation du *buzz* par analyse volumétrique. Ici, nous proposons de remplacer les information expertes par des caractéristiques de sentiment extraites des tweets.

Nous proposons d'évaluer nos performances suivant deux métriques. Dans un premier temps nous présenterons nos résultats suivant une méthode classique d'évaluation des problèmes de régression, le taux d'erreur moyen ρ, exprimé en pourcentage. Soit x_f le résultat d'un film f au box office, et $\hat{x}_f$ le résultat estimé par la fonction de prédiction, on définit $\rho(f) = 100 \times \frac{|\hat{x}-x|}{x}$. Les résultats au box office étant très différents d'un film à l'autre, nous avons préféré cette méthode à un critère des moindres carrés. Dans un second temps, nous évaluerons nos performances suivant un critère de classification en 10 classes, chaque classe correspondant à un décile des résultats au box-office (exprimés en dollars). Nous avons considéré cette mesure car les résultats d'un film à l'autre variaient de plusieurs ordres de grandeur (*cf.* Table 4.4).

Analyse des Résultats

Nous proposons d'utiliser comme modèle de base une approche purement volumétrique (cf Table 4.4), c'est à dire qu'elle n'utilise que le nombre de tweets publiés pour prédire le résultat. Ses résultats, présentés en Fig. 4.3, présentent un taux d'erreur moyen de 210% et un taux d'erreur en classification à 10 classes de 56%.

Après sélection de variables et optimisation du paramètre de régularisation λ, notre modèle de régression obtient les résultats suivants : 25% d'erreur moyenne (pour la prédiction du résultat numérique au box-office) et 10% d'erreur en classification. Il est donc clair que les caractéristiques en sentiment représentent de bons indicateurs. Même si notre processus d'optimisation génère des résultats légèrement surévalués [7], le fossé entre les résultats des deux approches reste clairement significatif et démontre l'utilité de l'analyse de sentiments pour cette tâche.

7. Nous n'avions pas d'ensemble de test réellement indépendant.

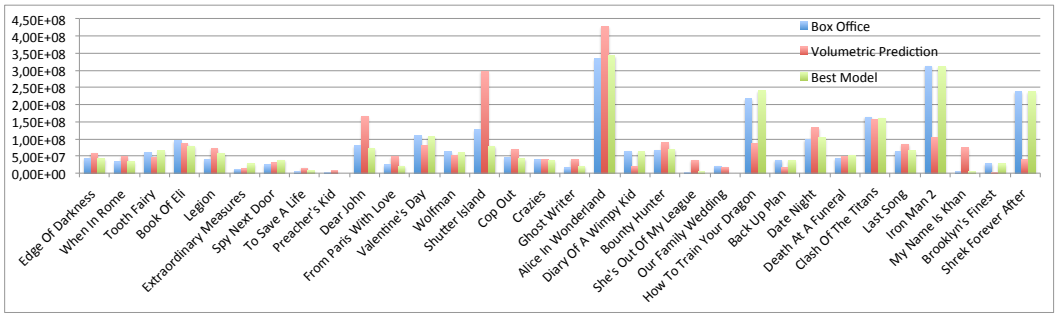


Figure 4.3: Comparaison entre les résultats réels au box office, l'approche volumétrique, et notre approche sentiment (exprimé en pourcentage d'erreur).

Sélection de Variables et Paramètres de Régularisation

Comme nous l'avons déjà évoqué précédemment en Section 4.3.2, les caractéristiques de sentiment sont fortement corrélées et il est impossible d'appliquer efficacement une régression aux données brutes, c'est pourquoi nous avons opté pour une méthode de sélection gloutonne séquentielle basée sur un critère d'évaluation LOO. La Figure 4.4 illustre l'évolution de l'erreur en fonction du nombre de variables utilisées.

On obtient des performances optimales avec 34 variables (25% d'erreur moyenne sur l'estimation du résultat au box office et 10% d'erreur en classification). L'allure en dents de scie de la courbe est due à la faible régularisation mise en œuvre. Trois autres points méritent d'être mis en avant :

— l'utilisation d'une seule variable mène, au mieux, à un taux d'erreur de 600% alors que l'approche volumétrique présente de meilleurs résultats avec un taux d'erreur de 210%. Cette disparité s'explique facilement. En effet, l'approche volumétrique consiste en une normalisation sur l'ensemble des données alors que les expériences présentées en Figure 4.4 sont basées sur une méthode LOO, moins biaisée et plus dure à optimiser.

— le premier plateau de performances est atteint après la sélection de dix variables. Dans ces 10 variables (16, 5, 15, 2, 17, 16, 7, 3, 4, 10), apparaissent trois indicateurs de subjectivité et cinq caractéristiques apprises sur des sources traitant de cinéma. Si la présence de variables liées au cinéma était à prévoir, il est intéressant de remarquer que ni la caractéristique volumétrique, ni celle apprise sur le golden standard ne sont retenues ;

— enfin, l'étude de la sélection de variables nous permet de remarquer que dans les 34 caractéristiques retenues, on retrouve à part égale indicateurs du score moyen `aps` et du volume négatif `nv`, mais aucune trace du volume positif. Ceci peut en partie s'expliquer par le déséquilibre du jeu de données (84% de tweets positifs). Dans cette disposition, le volume négatif semble plus discriminant.

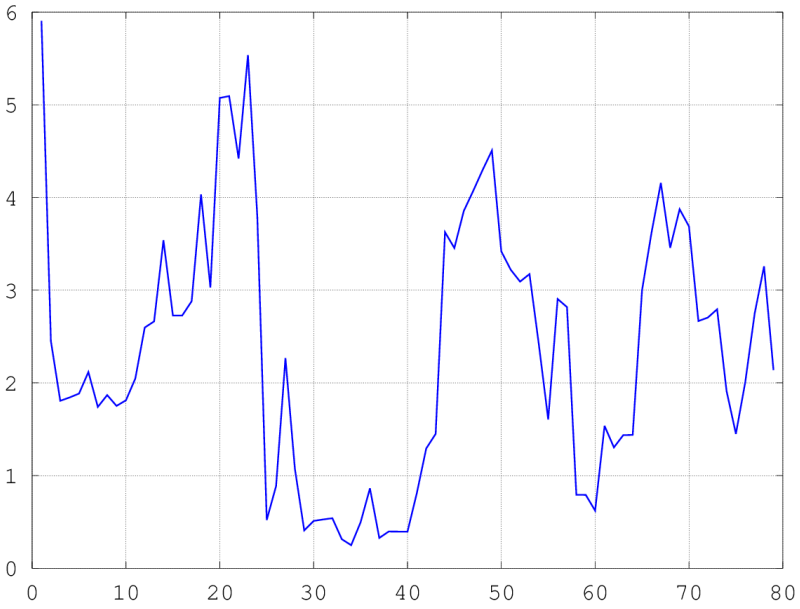


Figure 4.4: Erreur moyenne en prédiction du box office (exprimée en pourcentage) en fonction du nombre de caractéristiques sélectionnées.

Pour ce qui est de la régularisation, nous avons optimisé λ par recherche en grille sur une échelle linéaire. Les résultats sont présentés en Figure 4.5. Les meilleures performances sont obtenues pour $\lambda = 10^{-5}$.

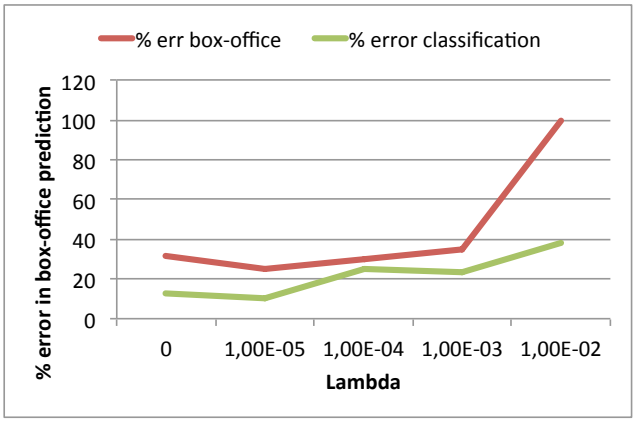


Figure 4.5: Évolution des deux critères d'erreur en fonction du paramètre de régularisation λ.

4.4 Conclusion

Dans ce chapitre, nous avons proposé une méthode inspirée de l'analyse de buzz pour l'extraction et l'utilisation de caractéristiques de sentiments. Faute de pouvoir mettre en place un paradigme d'évaluation en ligne, nous avons testé nos méthodes sur une tâche transverse, la prédiction de résultats au box office. Lors de nos expériences, nous avons pu valider l'intérêt de telles méthodes.

La différence d'ordre de grandeur entre les résultats de l'approche volumétrique et de celles utilisant les marqueurs d'opinion démontre l'intérêt d'utiliser ces marqueurs dans le cadre d'une tâche de prédiction. Si nous nous sommes concentrés ici sur le problème du box office, l'utilisation des descripteurs d'opinion peut aisément être transposée à la tâche de recommandation, notamment sous la forme d'un prior semblable au biais item décrit dans la section 3.2.2 dans le cadre du démarrage à froid.

5 Recommandation Contextualisée par l'Utilisation du Texte Brut

> *You taught me language, and my profit on 't*
> *Is I know how to curse.*
>
> — **William Shakespeare**

5.1 Introduction

Dans ce chapitre, nous nous intéressons à la recommandation par filtrage collaboratif et présentons une contribution double. D'une part nous proposons d'utiliser les données textuelles brutes pour affiner les profils latents utilisés dans la recommandation par factorisation matricielle, et d'autre part nous introduisons une méthode de génération de revue visant à mieux expliquer la recommandation à l'utilisateur.

5.1.1 Enrichissement des Profils Latents

Comme nous l'avons vu dans le chapitre 3, dans le cadre du filtrage collaboratif, les notes attribuées par un utilisateur dans le passé forment son profil et la comparaison des profils permet de générer des propositions (les films qui ont été appréciés par des utilisateurs ayant des profils proches). La classification de sentiment modélise les documents textuels pour caractériser leurs polarités en analysant des mots ou des groupes de mots qui seront considérés comme des marqueurs d'opinion. Dans ce chapitre, nous proposons une approche novatrice mêlant des techniques issues de la classification de sentiments et du filtrage collaboratif pour tirer parti des notations de consommateur ainsi que des revues textuelles associées.

Comme précédemment, nous notons respectivement u, i et $r_{u,i}$ les utilisateurs, les produits et les notes associées. De plus, nous désignerons les documents associés

aux couples (u, i) par $d_{u,i}$. Nous cherchons à construire un modèles $f(u, i) = \hat{r}_{u,i}$ qui soit une bonne approximation de $r_{u,i}$, ce modèle aura la forme suivante :

$$
\begin{aligned}
f_{Texte}(u,i) \quad = \quad & \lambda_0 g_0 + && \text{historique moyen} && (5.1)\\
& \lambda_1 g_1(u) + && \text{historique de l'utilisateur} && (5.2)\\
& \lambda_2 g_2(i) + && \text{historique du produit} && (5.3)\\
& \lambda_3 g_3(u,i) + && \text{historique joint} && (5.4)\\
& \lambda_4 g_4(d_{u,i})) && \text{documents associés à l'utilisateur et au produit} && (5.5)
\end{aligned}
$$

Les composantes (5.2) et (5.3) permettent de construire un système basique : il est établi que la note moyenne donnée à un produit dans le passé (indépendamment des utilisateurs) et la note moyenne d'un utilisateur (indépendamment des produits) sont des caractéristiques essentielles pour estimer $r_{u,i}$. La composante (5.4) a fait l'objet de nombreuses études [AT05] et repose souvent sur les techniques de factorisations matricielles permettant de prédire les valeurs manquantes de la matrice $R = \{r_{u,i}\}$ [WZ13]. La dernière composante est nouvelle : il s'agit de notre contribution. Elle est à mettre en regard avec la proposition de [ML13c] qui intègre directement les données textuelles dans le processus de recommandation dans l'idée de démontrer quantitativement l'intérêt de prendre en compte cette nouvelle ressource. Les auteurs ont fusionné les termes (5.4) et (5.5) dans un système à variable latente tandis que nous proposons d'utiliser directement le texte brut. Nous démontrerons l'intérêt de notre approche par rapport à un système d'analyse en variables latentes. Nous donnerons les détails de notre modèle en section 5.2.

5.1.2 Génération de Revue et Explication de la Recommandation

L'utilisation conjointe du modèle de recommandation et d'un modèle de fouille d'opinion permet d'envisager la tâche de prédiction du texte de la critique en plus de la note. Nous proposons d'utiliser cette prédiction de texte comme un moyen d'expliquer à l'utilisateur ce qui a poussé le système à lui proposer tel ou tel produit. Dans ce but, nous voulons que le texte souligne les facteurs décisifs tout en utilisant un vocabulaire proche de celui de l'utilisateur.

Comme évoqué dans la Section 3.6, le choix du protocole d'évaluation dépend fortement du paradigme que l'on veut tester. Dans le cadre de ces travaux, nous n'avons pas réussi à trouver une méthodologie d'évaluation qui soit adaptée et à un modèle prenant en compte le contexte textuel et à un modèle temporel (*Time Aware Recommender System* ou TARS). C'est pourquoi nous avons décidé de séparer

l'évaluation de ce modèle de celle des TARS présentés dans le chapitre 6. Une série d'expérience démontrant l'intérêt de cette approche est décrite dans la section 5.3.

5.2 Contributions

Lors de l'élaboration de ces travaux, nous nous demandions quelle était la meilleure façon de caractériser un utilisateur et ses intérêts. Est-ce par l'extraction de thèmes latents, comme proposé dans [ML13c] ou est-ce par la caractérisation de son style et vocabulaire ? Ici, nous nous concentrons sur cette seconde approche pour attaquer deux tâches complémentaires. Tout d'abord, nous utilisons le texte brut des revues pour enrichir les profils dans le cadre de la prédiction de notes. Dans un second temps, nous cherchons à utiliser ce texte pour expliquer la recommandation. A l'aide des revues écrites dans le passé, nous cherchons à caractériser le style et les goûts de l'utilisateur pour lui proposer une critique de l'objet proposé.

5.2.1 Texte Brut et Prédiction de Notes

Nous considérons des données sous la forme d'un quadruplet $(u, i, r_{u,i}, d_{u,i})$ où u et i sont respectivement l'index de l'utilisateur et de l'objet ou *item*, $r_{u,i}$ est la note laissée par l'utilisateur u à l'objet i et $d_{u,i}$ le texte critique accompagnant cette note. Par concision, nous utiliserons parfois le couple (u, i) à la place du quadruplet. L'objectif étant de prédire le mieux possible la note donnée par un utilisateur à un objet, le critère utilisé pour l'évaluation des modèles est l'erreur quadratique moyenne, notée MSE pour *Mean Squared Error* sur l'ensemble des m_{test} critiques de test, définie équation 5.6.

$$E_{MSE} = \frac{1}{m_{test}} \sum_{(u,i) \in test} (r_{u,i} - f(u,i))^2 \tag{5.6}$$

La contribution réside dans l'ajout aux termes g_0, g_1, g_2 et g_3 présentés en Sec.3.2.2 d'un terme utilisant le texte brut. La première étape est l'extraction d'un dictionnaire $\mathcal{D}$ sur l'ensemble des documents d'apprentissage. Il contient les mots qui apparaissent dans au moins 10 critiques différentes. Les mots vides ne sont pas enlevés. Pour pouvoir extraire les attentes de chaque utilisateur, ainsi que les qualités de chaque objet, nous proposons d'utiliser trois représentation par utilisateur et objet. Chaque représentation est un sac de mot codant la présence des mots du dictionnaire $\mathcal{D}$. La première , $b^{(a)}$ correspond à la concaténation de toutes les critiques dans l'ensemble d'apprentissage de l'utilisateur ou sur l'objet. Les deux autres correspondent à la

concaténation de toutes les critiques positives, $b^{(+)}$, d'une part et négatives, $b^{(-)}$ d'autre part.

Pour la prédiction, ce modèle calcule les neuf cosinus, comme présenté dans l'équation Eq.5.7, entre les trois représentations de l'utilisateur u et les trois représentations de l'objet i. Pour palier au problème de prédiction pour les utilisateurs ou objets n'étant pas présents dans la base de tests (*cold start*), une représentation générique est extraite avec l'ensemble des critiques et est utilisée en remplacement d'un utilisateur ou objet manquant lors de la validation ou du test.

$$\begin{aligned} g_4(u,i) \quad = \quad & (\cos(b_u^{(a)}, b_i^{(a)}), \cos(b_u^{(a)}, b_i^{(+)}), \cos(b_u^{(a)}, b_i^{(-)}) \\ & \cos(b_u^{(+)}, b_i^{(a)}), \cos(b_u^{(+)}, b_i^{(+)}), \cos(b_u^{(+)}, b_i^{(-)}) \\ & \cos(b_u^{(-)}, b_i^{(a)}), \cos(b_u^{(-)}, b_i^{(+)}), \cos(b_u^{(-)}, b_i^{(-)})) \end{aligned} \tag{5.7}$$

Le modèle réalise la combinaison linéaire, notée f_{Texte} des prédictions de tous les modèles : note moyenne, note moyenne par utilisateur, note moyenne par objet, prédiction de la factorisation matricielle et cosinus extraits du texte (voir équations 5.1 à 5.5 en introduction). Étant données les dimensions du problème, l'apprentissage des coefficients de la combinaison linéaire présentée dans l'équation Eq.5.1 se fait rapidement par régression linéaire sur l'ensemble de validation. Cette idée simple permet d'intégrer les prédictions des différents modèles.

5.2.2 Génération de Revue Personnalisée

Lorsque nous avons travaillé sur ces problématiques, la tâche de génération de revues personnalisée a été effectuée avec Mickael Poussevin [Pou14].

Nous avons choisi de construire les textes prédictifs comme l'agrégation de phrases extraites de critiques de l'item considéré écrites par d'autres utilisateurs. En d'autres termes, pour un couple (u,i), nous utilisons la prédiction $\hat{r}_{u,i}$ de notre modèle pour sélectionner les critiques sur l'objet i de la base d'entraînement faites par d'autres utilisateurs u' dont la note donnée $r_{u',i}$ est proche de $\hat{r}_{u,i}$. Les textes de ces critiques sont ensuite analysés pour en sélectionner les phrases dont le champ lexical est le est le plus proche de celui de l'utilisateur u (*i.e.* les phrases avec le plus de mots en commun avec le lexique de u). Il est donc nécessaire de définir deux mesures de

similarité, la première entre les notes, σ_r et la seconde entre les textes π_u et $\pi_{u'}$ de deux utilisateurs u et u', σ_t :

$$\begin{aligned} \sigma_r(r_{u'i}, r_{ui}) &= 1/(1 + |r_{u'i} - r_{ui}|) \\ \sigma_t(\pi_{u'}, \pi_u) &= \pi_{u'}\pi_u/(\|\pi_{u'}\|\|\pi_u\|) \end{aligned} \tag{5.8}$$

Nous proposons alors d'associer un score à un texte, $s_{u',i}$ écrit par un utilisateur u' au sujet d'un item i, de la façon suivante :

$$h(s_{u'i}, r_{u'i}, u', u, i) = \frac{\sigma_t(s_{u'i}, \pi_u) + \sigma_r(r_{u'i}, \hat{r}_{ui})}{2} \tag{5.9}$$

Notons que cette fonction permet d'évaluer n'importe quel texte, que ce soit une phrase, un paragraphe, ou une critique entière, ce qui permet de régler la granularité du modèle de prédiction. Grâce à ce critère, nous sommes en mesure de proposer trois approches pour la prédiction de revue. La première, baptisée 1S (pour one sentence) consiste à renvoyer la meilleure phrase $s_{u'i}$ de l'ensemble d'apprentissage. La seconde approche, CT (pour complete text) consiste à récupérer la meilleure critique $d_{u'i}$ parmi toutes celles disponible dans l'ensemble d'apprentissage. Enfin, la troisième méthode, XS (pour multiple sentences), consiste sélectionner de multiples phrases parmi l'ensemble d'apprentissage. La première phrase est sélectionnée par 1S. Ensuite, le choix des phrases suivantes est conditionné par trois contraintes : la pertinence, la diversité et la taille.

Pertinence : chaque phrase sélectionnée doit être pertinente au sens du critère de sélection h définit en Eq. 5.9.

Diversité : chaque nouvelle phrase sélectionnée doit apporter une information différente que celle contenue dans les phrases précédemment sélectionnées.

Taille : pour mieux être compris par l'utilisateur, il ne s'agit pas seulement d'utiliser le même lexique que lui, mais aussi son mode de pensée : quelqu'un de concis préférera un texte court alors qu'une autre personne pourra préférer une critique plus détaillée. Ainsi, nous pensons que la taille du texte doit s'adapter aux habitudes de l'utilisateur, c'est pourquoi nous proposons de personnaliser le nombre de phrases sélectionnées : il se verra proposer une critique de taille équivalente à sa critique moyenne en apprentissage.

L'algorithme décrit la procédure XS de génération d'un texte d_{ui} pour un utilisateur u sur l'item i.

5.3 Expériences

Données : u, i, $S = \{(s_{u'i}, r_{u'i}\ u'\}$
Résultat : $\hat{d}_{ui}$
$s^*_{u'i} \leftarrow \underset{s_{u'i} \in S}{\operatorname{argmax}}(h(s_{u'i}, r_{u'i}, u', u, i))$;
$\hat{d}_{ui} \leftarrow s^*_{u'i}$;
Retirer $s^*_{u'i}$ de S;
tant que taille $\hat{d}_{ui} <$ taille_moyenne(u) **faire**
    $s^*_{u'i} \leftarrow \underset{s_{u'i} \in S}{\operatorname{argmax}}(h(s_{u'i}, r_{u'i}, u', u, i) - \cos(s_{u'i}, \hat{d}_{ui}))$;
    $\hat{d}_{ui} \leftarrow s^*_{u'i}$;
    Retirer $s^*_{u'i}$ de S;
fin

Algorithme 6 : Algorithme d'extraction de plusieurs phrases (**XS**) : processus glouton. Sélection successive de phrases maximisant et la pertinence et la diversité. $\hat{d}_{ui}$ est le texte ainsi généré, phrase par phrase.

Nom	#Utilisateurs	#Objets	#Entraînement	#Validation	#Test
RBu52i200	52	200	7200	900	906
RBu520i2000	520	2000	388200	48525	48533
RBu5200i20000	5200	20000	1887608	235951	235960
RBu29265i110364	29265	110364	2339296	292412	292415
Au213i122	213	122	984	123	130
Au2135i1225	2135	1225	31528	3941	3946
Au21353i12253	21353	12253	334256	41782	41791
Au213536i122538	213536	122538	1580576	197572	197574
Au2135360i1225387	2135360	1225387	4642808	580351	580357

Table 5.1: Tailles des jeux de données utilisés. Le nom de chaque jeu de données se lit de la façon suivante : les deux premières lettres indiquent la source (Ratebeer ou Amazon), le chiffre après le u indique le nombre d'utilisateurs considérés et celui après le i, le nombre d'items considérés.

5.3.1 Données

Les données utilisées (présentées dans la table 5.1) sont issues des sites *ratebeer.com*, où des utilisateurs peuvent noter des bières en commentant leurs critiques (voir tableau 5.2), en anglais, et *amazon.com* (voir tableau 5.3). Des bases de tailles différentes sont utilisées. Elle sont extraites en indexant et comptant le nombre de critiques pour chaque utilisateur et pour chaque objet. La sélection s'effectue alors en récupérant toutes les critiques d'un certain nombre d'utilisateurs, sur un certain nombre d'objet, en conservant le rapport entre ces deux quantités sur le site. Les critiques de chaque base sont ensuite répartie aléatoirement en trois ensembles, un d'entraînement qui contient 80% des données, puis 10% en validation et 10% en test.

Champ	Valeur
`beer/beerId`	3213
`beer/brewerId`	232
`beer/name`	Third Coast Old Ale
`beer/ABV`	10.2
`beer/style`	Barley Wine
`review/profileName`	BeerandBlues2
`review/appearance`	3/5
`review/palate`	3/5
`review/taste`	5/10
`review/aroma`	7/10
`review/overall`	13/20
`review/time`	1145836800
`review/text`	Blind rating. Pours copper with virtually no head. Aroma is floral, perfumey hops, caramel malt (cookie) and maple notes. Heavily hopped, bitter and acidic drying flavor, maple and caremel malt , burning alcohol (isopropyl) finish. Medium bodied and smooth palate.

Table 5.2: Une critique du site *ratebeer.com*

Champ	Valeur
Utilisateur	A11K7IFISH954S
Objet	B0000025ZF
Date	January 31, 2002
Feedbacks positifs	2
Feedbacks	5
Note	2
Titre	Not feeling his latest style
Texte	I don't know what happened? Sure, he went into the spirit thing, but man what happened? Not only has he changed spiritually, but vocally and in style. Ma$e used to be a regular rap artist, just one of the best around. Who could forget his wonderful part in the song, "Mo Money Mo Problems"? And that was just a part of a song! Now, after 5 years, everything changed. His vocals sound extremly weak in every song. And what is with his style? On his "Welcome Back" vid, he wears what? What was that? A dollar store T-Shirt and cheap partially ripped jeans? Really, something happened to this guy, and I know it had absoultely nothing to do with him being spiritual. But the main question is if you should buy or not. If you own any Ma$e CD, just throw this in with your collection. If you don't, forget it. It's really not worth it.

Table 5.3: Une critique du site *amazon.com*

5.3.2 Modèle de Référence

Dans cette partie, nous présentons les trois modèles de référence que nous utilisons comme *baseline* dans la suite de ce chapitre. Le premier est un modèle à biais, le second une factorisation matricielle classique et le troisième est une factorisation matricielle à laquelle nous ajoutons un terme basé sur le texte, sous la forme d'une extraction de thématiques par allocation latente de Dirichlet (LDA).

Historique des Notes Nous proposons comme modèles de base les trois modèles d'historique présentés dans la section 3.2.2 : global, utilisateur et objet. Ces modèles simples estiment respectivement la note moyenne sur l'ensemble des critiques d'apprentissage, sur l'ensemble des critiques par utilisateur et par objet 5.10.

$$g_0 = \frac{1}{m_{tr}} \sum_{(u,i)\in tr} r_{u,i}, \qquad g_1(u) = \frac{1}{m_{tr}^{(u)}} \sum_{(u,i')\in tr} r_{u,i'} \qquad g_2(i) = \frac{1}{m_{tr}^{(i)}} \sum_{(u',i)\in tr} r_{u,i'} \tag{5.10}$$

Dans le cas où, en test ou validation, l'utilisateur ou l'objet de la critique à noter ne possède pas d'historique sur la base d'apprentissage, nous utilisons g_0 comme prédiction.

Factorisation Matricielle L'utilisateur u est alors représenté par le vecteur ϕ_u et l'objet i par ψ_i. La prédiction de ce modèle est issue de la correspondance entre les profils respectifs de l'utilisateur et de l'objet, qui s'exprime simplement au travers d'un produit scalaire, comme présenté en équation Eq.5.11

$$g_3(u,i) = \phi_u.\psi_i \tag{5.11}$$

L'apprentissage de ces représentations se fait, avec une fonction de coût erreur quadratique moyenne, sur l'ensemble des critiques de la base d'entraînement avec une régularisation L2 sur les paramètres. Pour alléger la notation, on note θ l'ensemble des paramètres du modèle.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{(u,i)} \left[(r_{u,i} - \phi_u.\psi_i)^2 \right] + \lambda_\phi \|\Phi_u\|^2 + \lambda_\psi \|\Psi_i\|^2] \tag{5.12}$$

Allocation Latente de Dirichlet Dans [ML13b], les auteurs montrent qu'il est avantageux d'intégrer le texte dans une représentation latente unique avec le modèle de factorisation matricielle, en utilisant l'allocation latente de Dirichlet (*LDA, Latent Dirichlet Allocation*) pour projeter les textes dans l'espace latent. Nous proposons

d'utiliser la représentation latente extraite par *LDA* comme une première version du terme $g_4(u,i)$ présenté dans l'équation Eq.5.5. Ce modèle est défini par un opérateur π qui associe à un document la probabilité d'appartenance à chaque thème extrait par le modèle. Pour chaque utilisateur (resp. item), on ainsi peut définir un profil en appliquant l'opérateur π à l'ensemble des documents de l'utilisateur (resp. item) que nous noterons $d_{u,*}$ (resp. $d_{*,i}$). Dès lors, le terme g_4 devient :

$$g_4(u,i) = \pi(d_{u,*}).\pi(d_{*,i}) \tag{5.13}$$

L'ensemble des paramètres $(\lambda_i)_{0\leq i\leq 4}$ est appris *a posteriori*, sur l'ensemble de validation, pour produire une prédiction unifiée $f_{LDA}(u,i)$ (voir équations 5.1 à 5.5 en introduction) qui utilise l'information des modèles d'historique, de la factorisation matricielle et de *LDA*.

5.3.3 Apprentissage des modèles

Les modèles de biais sont estimés sur la base d'apprentissage. Le modèle de factorisation matricielle est appris par descente de gradient stochastique avec une régularisation L2 sur les représentations latentes. La sélection des paramètres s'effectue sur l'ensemble de validation, au sens de l'erreur quadratique moyenne de reconstruction. Les coefficients de mélange des résultats des autres modèles sont estimés sur les résultats obtenus sur la base de validation afin de sélectionner le meilleur jeu de paramètres possibles pour le test.

5.4 Résultats

Dans cette partie, nous présentons les obtenus lors de nos expériences, tout d'abord sur la tâche de recommandation, puis sur celle de génération de revues.

5.4.1 Recommandation

Plusieurs points se dégagent des résultats présentés dans les tableaux 5.4 et 5.5. Premièrement, le modèle de biais sur les objets est toujours plus performant que celui sur les utilisateurs. Cela peut s'expliquer par la supériorité du nombre de critiques par objet sur le nombre de critiques par utilisateurs. De plus, une note dépend plus fortement des qualités intrinsèques de l'item considéré que de celles de l'utilisateur qui l'émet.

Base	g_0	$g_1(u)$	$g_2(i)$	$g_3(u,i)$	f_{LDA}	f_{Texte}
RBu52i200	0,675752	0,653250	0,209135	0,197762	**0,192080**	0,195085
RBu520i2000	0,568507	0,525631	0,250894	0,223771	0,221825	**0,220878**
RBu5200i20000	0,677442	0,587828	0,307919	0,284660	0,271934	**0,271559**
RBu29265i110364	0,702961	0,606446	0,348766	0,331576	0,310704	**0,308892**
Au213i122	1,534800	1,565839	1,491599	1,977556	1,370343	**1,340891**
Au2135i1225	1,531551	1,304329	1,278509	1,213574	**1,055429**	1,061476
Au21353i12253	1,471075	1,285846	1,236089	1,212675	1,049964	**1,045249**
Au213536i122538	1,507212	1,445383	1,322291	1,297096	1,155048	**1,147167**
Au2135360i1225387	1,60510	1,63127	1,49281	1,48153	1,33138	**1,32666**

Table 5.4: Résultats des modèles sur les bases Ratebeer (**RB**) et Amazon (**Au**) en erreur quadratique moyenne sur les critiques de test (meilleurs résultats en gras). On remarque que l'ajout de la dimension textuelle améliore les résultats sur tous les datasets. De plus, sur quasiment tous les datasets, l'utilisation du texte brut présente des résultats équivalents ou meilleurs que LDA.

Base	g_0	$g_1(u)$	$g_2(i)$	$g_3(u,i)$	f_{LDA}	f_{Texte}
RBu52i200	18,75	19,11	8,21	8,91	7,62	7,17
RBu520i2000	18,75	18,60	10,73	10,26	10,16	9,92
RBu5200i20000	25,03	24,65	14,43	14,32	12,54	12,42
RBu29265i110364	26,33	25,83	16,05	15,20	13,88	13,70
Au213i122	17,50	21,67	17,50	25,83	19,23	16,92
Au2135i1225	15,94	15,16	15,92	14,10	11,38	11,68
Au21353i12253	14,74	14,47	14,19	14,24	11,26	11,28
Au213536i122538	14,73	15,99	14,81	14,42	12,22	12,05
Au2135360i1225387	14,91	16,39	15,78	15,91	13,14	13,05

Table 5.5: Résultats des modèles sur les différentes bases en erreur de classification (positif/négatif) sur les critiques de test (meilleurs résultats en gras). Encoreune fois, l'ajout de la dimension textuelle améliore les résultats sur tous les datasets.

L'utilisation de l'information textuelle améliore les performances. En effet dans le tableau Tab.5.5, les meilleurs scores sont atteints par les modèles *Texte* et *LDA*. On peut par ailleurs noter que cet ajout est plus profitable dans le cas d'*amazon.com* que de *ratebeer.com*. La première raison a cela est que les textes sont plus fournis sur le premier jeu de données. La seconde raison est que sur les bases extraites d'*amazon.com*, un certain nombre d'objets et utilisateurs sont présents uniquement en validation et test, mais pas en entraînement, contrairement à celles issues de *ratebeer.com*. Pour ces objets ou utilisateurs, l'utilisation d'une représentation générique permet de palier au manque de connaissance. Les différences de *MSE* entre les bases issues d'*amazon.com* et de *ratebeer.com* sont en partie expliquées par le fait que les notes pour *amazon.com* sont exactement 1, 2, 3, 4 et 5 alors que pour *ratebeer.com* il s'agit de notes sur 20 divisées par 4 pour obtenir une note sur 5. Il existe donc dans la base des notes non entières comme 3,75, ce qui peut diminuer la distance entre la note et la prédiction.

5.4.2 Analyse des prédictions

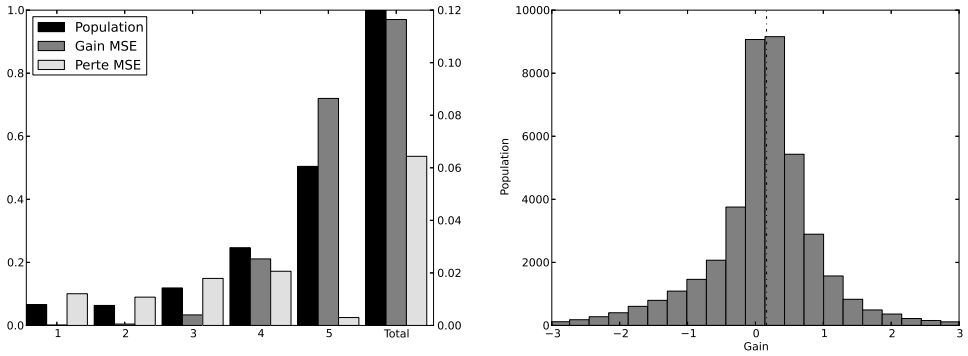
Afin de mieux comprendre les apports du modèle utilisant l'information textuelle dans la décision finale, nous en avons comparé les sorties, notée *Texte* avec le modèle de factorisation matricielle (*MF* pour *Matrix Factorization*).

Pour *amazon.com* sur Au21352i12253 Cette comparaison est effectuée sur les critiques de test de la base Au21352i12253. Les exemples que nous avons extraits, voir la table Tab.5.6, montrent que le modèle permet de corriger des erreurs de la part de la factorisation matricielle, que ce soit pour augmenter la note en cas de bonne correspondance entre ce qu'apprécie de l'utilisateur et les caractéristiques positives de l'objet ou pour la diminuer si, au contraire, les défauts de l'objet sont ceux que l'utilisateur n'apprécie pas.

Dans la suite , on note $f(u,i)$ la factorisation matricielle classique, l'indice $+$ (resp. $-$) représente la limitation au corpus de documents positifs (resp. négatif).

$$G_{\text{Texte}}(u,i) = (r_{u,i} - f_{\text{MF}}(u,i))^2 - (r_{u,i} - f_{\text{Texte}}(u,i))^2 \tag{5.14}$$

Dans la figure 5.1b, nous présentons l'histogramme des valeurs du gain $G_{\text{Texte}}(u,i)$, tel que défini équation (5.14), normalisé, pour chaque critique de test par l'utilisateur



(a) Population, gain en *MSE* et perte en *MSE* en fonction de la note. L'échelle de gauche représente la part de la population totale que représente chaque note. L'échelle de droite représente la MSE

(b) Histogramme des valeurs du gain $G_{\text{Texte}}(u,i)$ pour l'ensemble des critiques de test et leur médiane, évaluée à 0,164179

Figure 5.1: Histogrammes des Gains et Pertes en MSE (gauche) et $G_{Texte}(u,i)$ (droite)

u sur l'objet i. La figure affiche également la médiane de ces valeurs, qui vaut 0,164179. Nous confirmons donc qu'il y a un gain sur une majorité de critiques.

$$Gain_{\text{MSE}} = \frac{1}{\#\text{Test}} \sum_{(u,i)\in\text{Test}} |G_{\text{Texte}}(u,i)|_{+} \tag{5.15}$$

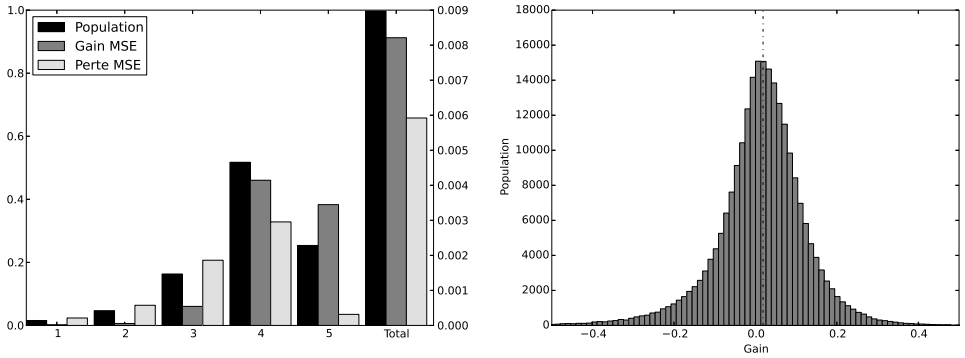
$$Perte_{\text{MSE}} = \frac{1}{\#\text{Test}} \sum_{(u,i)\in\text{Test}} -|G_{\text{Texte}}(u,i)|_{-} \tag{5.16}$$

La figure 5.1a montre à la fois la population par classe, le gain et la perte en erreur quadratique moyenne, présentés respectivement équations (5.15) et (5.16). On remarque une grande disparité entre les différentes notes : il y en a beaucoup plus de positives que de négatives. Ainsi la capacité du modèle à mieux estimer les notes positives permet de largement compenser la petite perte sur les mauvaises notes, ce qui explique le gain MSE moyen réalisé par le modèle de texte.

Pour *ratebeer.com* sur RBu5200i20000 Nous avons conduit la même étude sur RBu5200i20000 où nous avons agrégé les notes de *ratebeer.com*, qui sont initialement sur 20 et les avons divisé par 4 pour obtenir une note sur 5, sur 5 notes (1 à 5) par partie entière. Tout d'abord, l'histogramme et la médiane du gain, présentés figure 5.2b, confirme que le gain est moins important sur *ratebeer.com* que sur *amazon.com*, avec une répartition plus équilibrée entre gains et pertes et une médiane à 0,018727. Dans un premier temps, la figure 5.2a confirme l'écart entre gain total en *MSE* et perte totale est moins élevé que pour *amazon.com*. Elle indique également qu'il y a

Note	5,0
MF	4,479377
Text	4,764128
One of the best historical novels I've read This book is a wonderful tapestry of Norman/Angevin England and Wales. The characters are well-developed and complex. For example, historical treatments of King John invariably cast him as a villain, but here we see him as a character with many facets. The plot follows Joanna, or Joan, the illegitimate daughter of John, through her life from about age five to her late thirties. A reader of this book will learn much about culture clash, women, the Angevins, and England and Wales in the Middle Ages. The book is captivating – I was hardly able to put it down	
Note	1,0
MF	1,669823
Text	0,700161
Not taking it back. After comparing the print quality in best mode to my HP 970 CSE inkjet in best mode from the same source there is no comparison. The HP wins in print quality hands down. The CX5200 with its pigment ink is printing unsaturated colors and not sharp in best mode on my first day of use. The HP dye based ink colors are deep and the print is super sharp. I'm not taking this machine back to the dealer for a refund because the wife says the long life durabright ink is required for her scrapbooking. The software install is buggy on an XP home machine and the software is fairly worthless as well. Fortunately my MS Picture It that came with the Dell works with the scanner.	
Note	1,0
MF	1,296613
Text	0,900824
Man ! This one gave me a hemorrhoid This is just an awful attempt at making music. This guys music literally irritates my [ears] when I hear it. What is really messed up about the whole situation is this guy is polluting the minds of the children with the poor lyrics and ignorant subject matter.	
Note	4,0
MF	2,963587
Text	3,619592
Enjoy after repeated Play After spending hours actually forcing myself to listen to this CD, I have to begrudgingly admit that Alicia Keys MAY deserve some of the accolades she has received. The CD is set up so that each song compliments the one before. This is a nice album to mellow out and chill with.	

Table 5.6: Exemples de critiques où le texte apporte une meilleure classification sur Au21352i12253. Note décrit la note associée à cette revue, *MF* et *Text* décrivent les prédictions obtenues à l'aide de chacun de ces modèles.



(a) Population, gain en *MSE* et perte en *MSE* en fonction de la note. L'échelle de gauche représente la part de la population totale que représente chaque note. L'échelle de droite représente la MSE

(b) Histogramme des valeurs du gain $G_{\text{Texte}}(u, i)$ pour l'ensemble des critiques de test et leur médiane, évaluée à 0,018727

très peu de très bonnes notes sur *ratebeer.com*, qui sont les notes où notre modèle apporte beaucoup.

Nous avons également comparé les sorties des modèles et présentons des exemples où la prise en compte du texte améliore la prédiction du modèle table 5.7. La base RateBeer est complexe car elle contient en plus des utilisateurs traditionnels, une base d'experts qui notent les bières sur toute l'étendue de l'intervalle $[0, 20]$, 10 étant réservé à une bière moyenne. Les utilisateurs non experts ont tendance à utiliser des notes plus élevées.

5.4.3 Génération de Revues Personnalisées

A l'époque où nous avons réalisé ces travaux, la génération de revues personnalisées pour l'explication de la recommandation était une tâche nouvelle. Ainsi, il n'existait pas de protocole d'évaluation standard. Cette tâche se rapprochant d'une procédure de résumé, nous avons décidé d'utilisé la métrique ROUGE-n, comparant le texte généré à la revue écrite par l'utilisateur. La métrique ROUGE-n compare la proportion de n-grammes en commun entre le texte généré et le texte écrit, ainsi plus le score ROUGE-n est élevé, plus le texte généré peut être considéré comme un candidat valide. Nous avons utilisé des valeurs de n comprises entre 1 et 3. Un bon score ROUGE-1 signifie que le modèle a bien identifié le vocabulaire et un bon score ROUGE-2 ou ROUGE-3 signifie qu'il a réussi à identifier le style de l'utilisateur.

Nous proposons de comparer nos deux modèles *LDA* (f_A) et *Text* (f_T) à cinq modèles de référence que nous utiliserons comme baselines. Le premier, une fonction de tirage aléatoire (*Random*) des phrases représentera notre borne inférieure, nous

utiliserons trois oracles (*ROUGE-1*, *ROUGE-2*, *ROUGE-3*) pour définir notre borne supérieure. Enfin, pour notre dernier modèle de référence, nous utilisons une factorisation matricielle. Les performances de ces sept modèles sont disponibles en Figure 5.2. Ils sont composés de trois cellules correspondant aux mesures ROUGE-1,-2,-3. On remarque que la procédure 1S (extraction d'une phrase) présente systématiquement des résultats pire que CT (extraction d'une revue) et XS (extraction de X phrases). Ces résultats étaient prévisibles : une phrase contient moins de mots qu'une revue complète. Dans le cas de *RateBeer*, on remarque que la méthode XS propose des résultats bien meilleurs que CT, pour toutes les mesures. Ceci s'explique par la structure des revues très favorable à notre modèle. D'une part, les phrases sont courtes, précises et utilisent un vocabulaire spécifique. D'autre part, les textes décrivent autant les condition de la dégustation que la bière en elle même. Il est donc plus aisé de recréer une revue à partir de briques adaptées que de trouver une revue correspondant parfaitement.

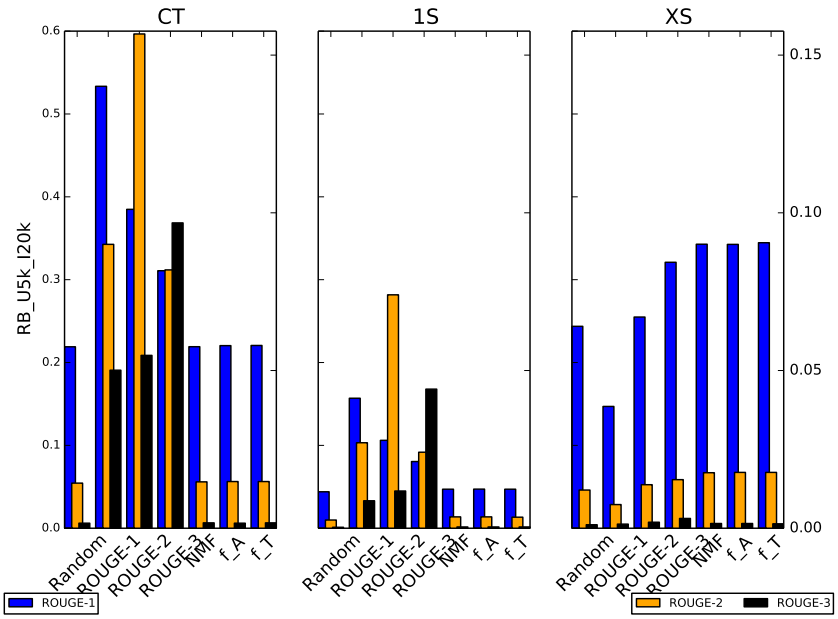
Dans le cas d'*Amazon*, les résultats sont plus mitigés : si XS et CT tiennent toujours le haut du tableau, elles sont difficiles à départager. Ceci est principalement dû à la grande variété de communautés d'utilisateurs présentes sur le site : on retrouve une plus grande disparité dans la maîtrise de la langue ainsi qu'une quantité non négligeable de textes *troll*.

Note	0.250000
MF	0.537500
Texte	0.293097
Tastes like a bad bottle of apple juice. I wouldn't give it to my dog (she prefers Bell's anyways).	
Note	0.750000
MF	1.235812
Texte	0.931757
Pours dark bronze. Smell of corn and metal. Metallic, blood (!) and some caramel extract aromas. Thin feel. Not good.	
Note	3.750000
MF	2.781928
Texte	3.025306
4th November 2008. Hazy brown beer. Small tan head. Malty start with a sweet malt mid. Then a few hops and spices in the dry finish. Well joined up.	
Note	2.250000
MF	1.812040
Texte	2.047176
22oz bottle-pours a foamy to ring white head and yellow/gold color. Aroma is sweet light/medium malt/grassy, some herbal. Taste is sweet light/medium malt/grassy, some herbal. Semi-dry/champagny. Thanks jwc215 for sharing.	

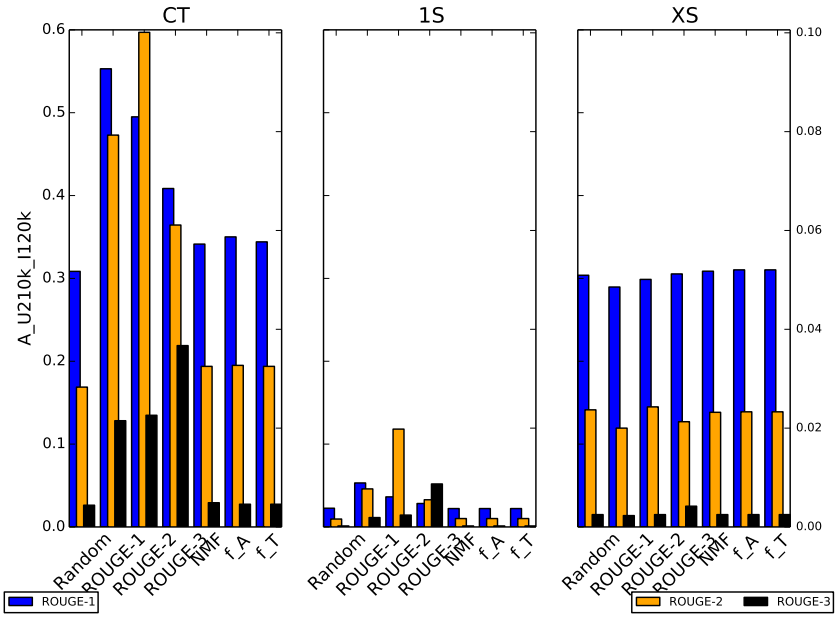
Table 5.7: Exemples de critiques où le texte apporte une meilleure classification sur RBu5200i20000

Texte	Prédictions
Great story and characters; often mannered writing I am going to weigh in very briefly on this book. It has a great story, but it is marred by Arundhati Roy's too frequent displays of mannerism. Many of the characters are very original and the story is full of credible twists and turns, but every thirty pages or so, Ms. Roy lapses into several pages of writing in apretentious stream-of-conscience/surreal style that soon had me skimming toward the next section of narrative substance. Ms. Roy must have felt that she needed to show off how well she could write, but she ended up underscoring the fact that this was her first novel.	The individual stories of at least twelve characters are told and each story would be rather simple but the stories are all shuffled together with no regard for tense and this makes the book seem much more complex than it actually is. Plus she moves forwards and backwards and sideways in time towards a central event which has been hinted at in countless ways but by the time you get to that event you are mad because all of the confusion could so easily have been avoided if she'd simply told the story, or each of the twelve stories, chronologically. At times the repetitions and sentence fragments and other affectations become more of a hindrance than a benefit, but it seems that some Indian writers feel compelled to write in this sort of native style, and if it is inevitable, then better Roy's fairly controlled method than Rushdie's incomprehensible over-the-top method.
This is a good read, it is a beautiful epic whose true force lies in the poignant details of its characters, richly detailed, woven into a wonderful tapestry. You all so get three books in one, a good bargin, and a good story for teen readers. OVERALL SCORE : (B+) READABILITY : (B), PLOT : (B-), CHARATERS : (A-), DIALOGUE : (B-), SETTING : (B+), ACTION/COMBAT : (B-), MONSTERS/ANTAGONISTS : (B-), ROMANCE : (B+), SEX : (n/a), AGE LEVEL : (PG)	Terry Brooks is probably the most talented of the derivative-of-Tolkien authors, a pleasant read for those who enjoy basic fantasy tales with a few original bits among the "Tough Guide to Fantasyland" plotlines. Best of the subgenre Terry Brooks' Shannara series was one of the first doorstopper series, varying in originality and in quality but good brain-candy fun. Now the first three books of his trilogy are rereleased in a hardvcover three-in-one edition, not recommended for people with weak wrists, but for those who like good covers and big heavy tomes.
Great story and characters; often mannered writing I am going to weigh in very briefly on this book. It has a great story, but it is marred by Arundhati Roy's too frequent displays of mannerism. Many of the characters are very original and the story is full of credible twists and turns, but every thirty pages or so, Ms. Roy lapses into several pages of writing in apretentious stream-of-conscience/surreal style that soon had me skimming toward the next section of narrative substance. Ms. Roy must have felt that she needed to show off how well she could write, but she ended up underscoring the fact that this was her first novel.	The tracks on this CD are made-up of acoustical-versions (without the drums or fancy synthesizers) of Dave's songs consisting from his earlier albums such as : "Under The Table and Dreaming," "Crash," some B-Sides material, and some other cool-stuff that didn't get played very much on the radio. Not only does it contain the Dave Matthew's 'favorites' such as "Crash into me", "Satellite", "Ants Marching", etc. but it contains songs never released on previous DMB albums, but some of the songs thrown in for a treat at Dave's famous concerts. If you don't buy this album for the fact that all the songs are great and played perfectly by both the musicians then buy it simpy for the fact that Tim Reynolds is an amazing guitarist the likes of which is rarely seen these days.

Table 5.8: Prédictions de texte de critiques issues de notre modèle.



(a) Histogramme des résultats sur RateBeer



(b) Histogramme des résultats sur Amazon

Figure 5.2: Histogrammes des performances en génération de revue sur les deux plus gros jeux de données, Ratebeer et Amazon. Le scores ROUGE-1, est représenté en bleu et associé à l'axe de gauche*. Les scores ROUGE-2,-3 sont respectivement représentés jaune et noir et associé à l'axe de droite. Sont représentés sept modèles : le modèle aléatoire (RNG), les trois oracles (ROUGE-1,-2,-3) la factorisation matricielle (NMF), un modèle textuel latent utilisant LDA (f_A) et le modèle textuel brut (f_T). Les résultats sont donnés pour les trois paradigmes étudiés : l'extraction de revue (CT), l'extraction de phrase unique (1S) et l'extraction de phrases multiples (XS)

5.5 Conclusion

Dans ce chapitre, nous avons fait l'hypothèse que la prise en compte du texte brut au travers d'un modèle de fouille de sentiments simple, une représentation d'un

utilisateur ou d'un objet comme trois documents textuels issus de la concaténation de toutes ses revues, de ses revues positives et de ses revues négatives, permet d'affiner la prédiction de notes dans un contexte de recommandation de produit.

Ce gain est issu de l'ajout d'information aux profils utilisateurs et objets et de l'utilisation de la correspondance entre attentes, positives ou négatives, d'un utilisateur et les qualités ou défauts d'un objet qu'il est difficile d'apprendre avec un modèle classique de recommandation. Les expériences que nous avons réalisées sur des bases de données de différentes tailles issues de *ratebeer.com* et *amazon.com* confirment notre hypothèse. L'amélioration des performances est liée à une meilleure estimation des bonnes notes (4 et 5). Comme elles sont les sont les plus nombreuses, elles permettent, par leur masse, une bonne estimation des attentes des utilisateurs et qualités des objets mais aussi des gains significatifs si elles sont mieux prédites.

Dans un second temps, nous avons cherché à apporter de l'explicativité aux résultats en proposant à l'utilisateur une critique personnalisée du produit recommandé. Cette approche, très novatrice reste cependant difficile à évaluer. En effet, les résultats présentés montrent bien les limitations des métriques ROUGE-n dans ce cadre-ci.

6 Utilisation du Contexte Temporel

> *Tempus fugit, augebitur scientia*
>
> — **Francis Bacon**

6.1 Introduction

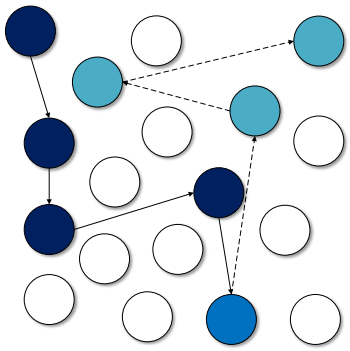


Figure 6.1: Représentation schématique du trajet d'un utilisateur au sein de l'espace de représentation des items. Le passé est représenté en bleu marine et le futur en cyan. Notre but est de modéliser un modèle de transition personnalisé capable de prédire efficacement et de manière robuste le prochain item de la trace.

Durant ces travaux de thèse, nous nous sommes intéressés à des paradigmes de recommandation où l'utilisateur prend une part active dans le processus d'accès à l'information, comme le e-commerce (*e.g* l'utilisation de la barre de recherche) ou encore l'accompagnement de visite dans les musées (*e.g* sélection d'un parcours thématique ou accès au descriptif d'une œuvre). Dans ce paradigme, l'ordre des actions des utilisateurs doit être pris en compte, c'est l'objet du présent chapitre.

Concrètement, cela signifie que pour améliorer la pertinence des réponses fournies par les systèmes de recommandation, nous proposons de mélanger deux approches. D'une part, nous définissons une topologie des items basée sur la modélisation des enchaînements décrits par la dynamique des utilisateurs au niveau local. D'autre part, nous utilisons des méthodes de filtrage collaboratif pour évaluer le niveau d'appétence d'un utilisateur pour les produits proposés (sous la forme d'une prédiction de note).

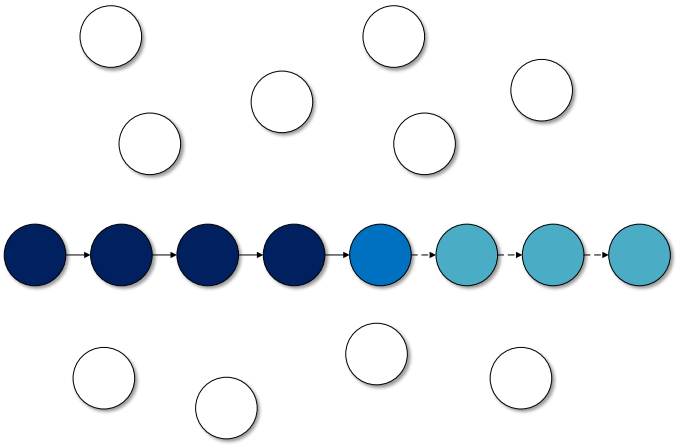


Figure 6.2: Représentation schématique d'un espace de représentation des items idéal où le trajet de l'utilisateur est aisément modélisable. Le passé est représenté en bleu marine et le futur en cyan.

Nous considérons d'abord le premier aspect du problème : la prédiction d'items. Nous pensons que la dynamique de l'utilisateur est l'élément clé pour aborder ce problème difficile : prédire la future position de l'utilisateur doit nous permettre d'émettre une proposition crédible et pertinente. C'est pourquoi nous proposons d'évaluer diverses stratégies de modélisation de cette dynamique. Nos premières expériences avec des données temporelles continues s'étant montrées peu concluantes, nous avons opté pour une modélisation plus simple, mais aussi plus robuste : le concept de temps est remplacé par celui de séquence et l'utilisateur est modélisé comme un mouvement au sein de l'espace de représentation des items. Si une régression globale sur l'ensemble des items vus précédemment semble peu performante, une modélisation de la transition entre deux items semble prometteuse. Afin de traiter la forte variabilité des utilisateurs isolés, nous proposons aussi une extension à ce modèle permettant la prise en compte d'information communautaire. Pour ce faire, nous segmentons la population d'utilisateurs en plusieurs communautés. Ainsi, nous pouvons apprendre un endomorphisme de l'espace de représentation des items, nous permettant de mieux tirer parti des préférences de chaque communauté.

Notre modèle est donc basé sur trois axes majeurs, la définition d'un espace de représentation des items, l'extraction de la dynamique des utilisateurs dans cet espace, et enfin, l'extraction d'information communautaire. Pour que l'espace de représentation des items se rapproche au maximum du cas idéal présenté dans la figure 6.2, il est impératif de construire un espace de représentation des items prenant en compte la dynamique des utilisateurs (*e.g.* deux items souvent vus l'un après l'autre devront avoir des représentations proches dans l'espace latent). Pour répondre à cet impératif, nous avons proposé deux modèles, un modèle à entrée/sortie, et un modèle issu de l'analyse de texte (Word2Vec). Une fois l'espace de représentation des

items créé, nous pouvons nous atteler à la tâche de personnalisation. La première piste étudiée fut la représentation de l'utilisateur comme une trajectoire dans l'espace des items [Guà+15]. Nous avons ensuite voulu affiner cette méthode en associant à chaque utilisateur une métrique sur l'espace des items, permettant d'adapter les représentations des items aux préférences de l'utilisateur. Malheureusement nous nous sommes heurtés à un problème de données. Les traces utilisateurs sont en moyenne trop courtes et trop bruitées pour apprendre efficacement les paramètres du modèle, devenus trop nombreux. Nous avons alors opté pour un regroupement des utilisateurs au sein de communautés.

La suite de ce chapitre s'articule comme suit : la section 6.2 décrit les processus de création d'espaces de représentation des items. Dans la section 6.3 nous présentons nos deux méthodes de personnalisation (l'une au niveau de l'utilisateur, et l'autre au niveau de la communauté) de la recommandation pour la prédiction d'items. La section 6.5 s'articule autour de l'utilisation des représentations apprises précédemment pour l'amélioration des performances de la factorisation matricielle dans le cadre de la prédiction de notes. Nous présentons l'évaluation des modèles dans 6.6 et enfin les conclusions dans 6.7.

6.2 Espaces de Représentation Temporalisés

6.2.1 Modèle à Entrée/Sortie

Dans [GS+14], nous nous sommes intéressés à la recommandation de parcours dans les musées. Le problème de la recommandation de parcours touristique présente plusieurs particularités : il s'agit de recommander non seulement des œuvres ou des points d'intérêt mais aussi des itinéraires thématiques et personnalisés. Du point de vue scientifique, les enjeux sont nombreux : transposer le problème de la recommandation à des séquences, prendre en compte les aspects dynamiques pour affiner les profils au fil des trajets effectifs des utilisateurs, intégrer des contraintes de temps de visite ou d'engorgement de certaines salles.

Les musées proposent à leurs visiteurs des parcours thématiques et des audioguides, et aujourd'hui diverses applications de recommandation embarquée sont en cours de développement, comme par exemple les applications smartphone du Louvre-Lens et du Guggenheim ou encore le projet AMMICO [1]. Les systèmes développés récemment dans les musées [Boh10 ; Kar+12] reposent fortement sur la recommandation de contenu : l'idée est d'élargir le champ autour d'une œuvre en proposant à l'utilisateur les ressources à la thématique proche, en lui montrant aussi des œuvres actuelle-

1. http ://ammico.fr/

ment non exposées et en lui fournissant des informations connexes sur l'auteur de l'œuvre.

Ici, nous nous sommes intéressés à la recommandation dynamique de points d'intérêt (POI) à laquelle nous avons intégré la notion de parcours. Nous avons proposé un modèle de prédiction temporelle non personnalisée basé sur une représentation des items par deux points de l'espace latent, un point d'entrée (représentant l'item avant que l'utilisateur ne l'ait vu) et un point de sortie (représentant l'item une fois qu'il a été vu).

A l'instar de [Bou+14], nous nous sommes basés sur un modèle de diffusion. Nous cherchons à construire un espace de représentation tel qu'un utilisateur interagissant avec un POI i sera plus intéressé par les POI dans le voisinage de i que par ceux qui en sont plus éloignés. En d'autres termes, deux items ayant une représentation latente proche seront susceptibles d'intéresser les même visiteurs.

Nous avons décidé d'ajouter une amélioration proposée par [Che+12]. Celle-ci consiste à dédoubler chaque représentation d'un POI en deux points pour tenir compte de l'ordre dans lequel les POI s'enchaînent : le premier correspondant à l'entrée, le second à la sortie. Cette amélioration permet bien d'orienter la visite : si un trajet entre trois POI a, b et c est souvent fait dans cet ordre, alors l'entrée de b sera proche de a et sa sortie de c, défavorisant ainsi le passage inverse. Dans le cas contraire, si le trajet $a-b-c$ est aussi emprunté que $c-b-a$, l'entrée et la sortie de b seront proches, ne favorisant ainsi aucun sens.

Soient I l'ensemble des POI, $\mathcal{T}$ l'ensemble des traces, ψ_i^{in} la représentation de l'entrée d'un POI $i \in I$ et ψ_i^{out} sa sortie, l'optimisation de ce modèle est effectuée par minimisation par descente de gradient de la fonction de coût suivante :

$$\mathcal{L} = \sum_{\tau \in \mathcal{T}} \sum_{(i,j) \in \tau} \sum_{f \in I \setminus j} (||\psi_i^{out} - \psi_j^{in}|| - ||\psi_i^{out} - \psi_f^{in}||) \quad \text{avec } j \text{ suivant de } i \text{ dans } \tau \tag{6.1}$$

Cette fonction permet d'obtenir des représentations proches pour les POI successifs tandis que les autres POI sont éloignés du point de référence. En inférence, nous utilisons simplement une distance euclidienne dans l'espace latent :

$$next_{IOLM}(i) = \underset{j \in I \setminus i}{\operatorname{argmin}} \|\psi_i^{out} - \psi_j^{in}\| \tag{6.2}$$

En s'affranchissant de la structure de graphe et en passant à une méthode d'optimisation stochastique (*Stochastic Gradient Descent*, ou SGD), ce modèle permet de s'attaquer à de très gros jeux de données, ce que les modèles précédents étaient

incapables de faire. Ce modèle présente d'autres avantages : il est résistant au bruit et surtout particulièrement évolutif.

6.2.2 Modèle Word2Vec

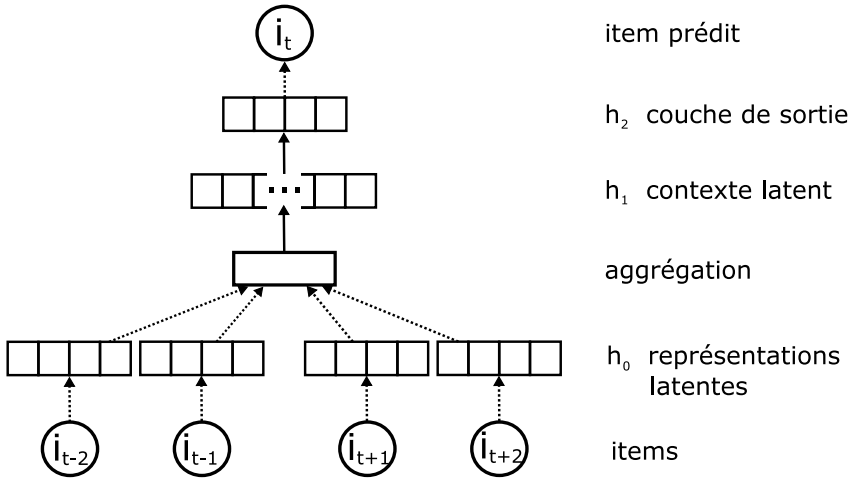


Figure 6.3: L'architecture Word2Vec [Mik+13] permet de prédire l'item courant à partir de son contexte.

En 2013, *Mikolov et al.* proposent une approche très efficace pour la modélisation de la dynamique locale dans les espaces latents. Dans [Mik+13], ils proposent de faire émerger de la sémantique dans un espace de représentation des mots : les mots sémantiquement proches (*e.g.* roi et reine) doivent avoir des représentations latentes proches.

Dans notre cas, nous cherchons à rapprocher les items qui plaisent aux mêmes personnes, on peut donc définir une sémantique des items, dans laquelle le sens serait défini par les différentes caractéristiques d'appétence d'un item. Ainsi, en remplaçant les mots par des items et les phrases par des traces utilisateurs, nous proposons, en adaptant Word2Vec à nos données, une méthode de construction d'espace de représentation des items respectant les dynamiques des utilisateurs. Il n'y a alors plus qu'une seule représentation par item.

Word2Vec, se divise en deux modèles, continuous bag of words (CBOW), et skip-gram. Le premier vise à retrouver un mot à partir des son contexte (les autres mots de la phrase) alors que skip-gram propose l'inverse : retrouver la phrase autour d'un mot donné. Dans notre cas, nous disposions de séquences complètes, l'utilisation d'une méthode ou l'autre était équivalente et nous avons choisi CBOW.

L'architecture générale de CBOW est illustrée dans la figure 6.3. Dans cet exemple, le but est de prédire l'item i_t à l'aide de sous-séquences τ_t de la trace utilisateur τ de taille fixe n (ici, $i_{t-2}, i_{t-1}, i_{t+1}, i_{t+2}$). Après avoir récupérer les représentations

latentes $\phi i_{t-2}, \phi i_{t-1}, \phi i_{t+1}, \phi i_{t+2}$ des items de la sous séquence τ_t , on les concatène (ou somme, suivant l'implémentation utilisée) en un vecteur unique v_1 qui sert ensuite à prédire la représentation latente de v_2 proche de ϕi_t.

L'apprentissage de modèle se fait en itérant un processus en deux étapes sur toutes les sous-séquences de taille n des traces utilisateur. C'est ce découpage des traces en sous-séquence qui permet à *Word2Vec* de capturer l'information séquentielle. Dans un premier temps, on applique le processus décrit au dessus, puis dans un deuxième temps, on inverse le protocole et le système doit prédire la sous-séquence extraite à partir de i_t. Comme expliqué dans [Mik+13], l'alternance de ces deux méthodes permet l'apprentissage de représentations robustes.

L'inférence est encore une fois effectuée par recherche des plus proches voisins :

$$next_{W2V}(i) = \underset{j \in I \setminus i}{\operatorname{argmin}} \|\psi_i - \psi_j\| \tag{6.3}$$

Si ce modèle permet l'apprentissage de représentations plus riches que le précédent, il est aussi, contrairement à IOLM, très compliqué à apprendre.

6.3 Personnalisation

L'espace de représentation appris à l'aide des méthodes précédentes est global à l'ensemble des utilisateurs. Nous voulons ici adapter cet espace à chaque utilisateur. Pour cela , nous allons introduire une transformation de l'espace appris qui dépend de l'utilisateur. Pour limiter le nombre de paramètres du nouveau modèle (une transformation par utilisateur), nous avons choisi une transformation simple sous la forme d'une translation dans l'espace de représentation. Le vecteur de translation associé à u sera noté b_u :

$$\Phi_u(\psi_{i_t}) = \psi_{i_t} + b_u, \qquad b_u \in \mathbb{R}^d \tag{6.4}$$

L'étape d'inférence s'effectue donc encore une fois par recherche des plus proches voisins de l'item courant.

$$next(i, u) = \underset{j \in I}{\operatorname{argmin}} \|\Phi_u(\psi_i) - \psi_j\| \tag{6.5}$$

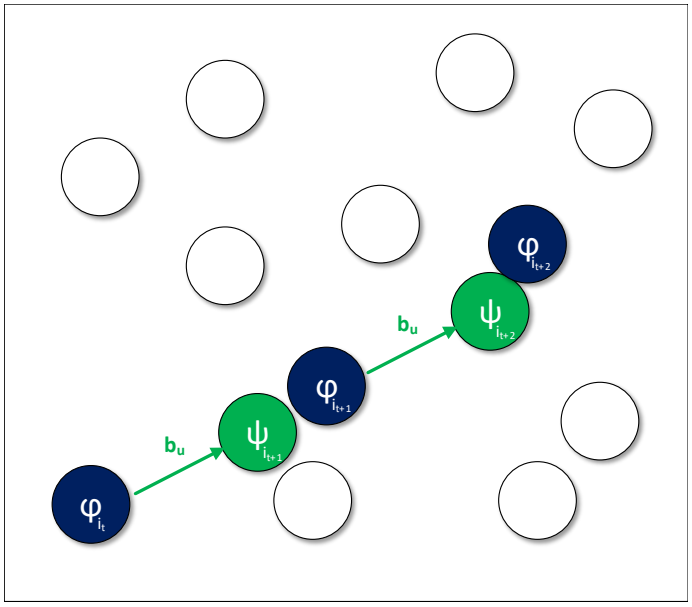


Figure 6.4: Personnalisation par modélisation de l'utilisateur comme une translation dans l'espace de représentation des items : considérons que l'utilisateur u interagis avec l'item i_t à un instant donné t. En appliquant la transformation Φ_u à la représentation ψ_{i_t} de l'item i_t, on obtient une estimation de la position de u dans l'espace de représentation des items à l'instant $t+1$. Cette estimation doit alors être proche de $\psi_{i_{t+1}}$, la représentation de i_{t+1}, l'item avec lequel u interagit à l'instant $t+1$.

Lors de l'apprentissage de ce modèle, nous utilisons les représentations des items (fixes) comme des points d'ancrage, et nous concentrons l'apprentissage sur les vecteurs b_u. Nous proposons ici deux méthodes pour l'apprentissage de $b_u, u \in U$.

Méthode statique Dans un premier temps, nous avons envisagé une méthode sans itérations pour l'apprentissage des vecteurs $b_u, u \in U$:

$$\begin{gathered} \forall u, i_t = \theta_{u,t}, \qquad (b_u)_{t+1} = \alpha(b_u)_t + (1-\alpha)(\psi_{i_{t+1}} - \psi_{i_t}), \\ (b_u)_0 = 0^d, \qquad b_u \in \mathbb{R}^d, \qquad \alpha \in [0,1] \end{gathered} \tag{6.6}$$

Apprentissage par Descente de Gradient Stochastique Dans un second temps, nous avons aussi proposé l'apprentissage des vecteurs $b_u, u \in U$ par descente de gradient stochastique sur la fonction de coût suivante :

$$\mathcal{L} = \sum_{\substack{\tau \in \mathcal{T} \\ i \in \tau \\ j \in I \setminus i}} \|\Phi_u(\psi_{i_{t-1}}) - \psi_j\|^2 + \lambda \Omega(\Phi) \tag{6.7}$$

Comparaison des Deux Méthodes Lors de nos expérimentations, nous avons pu remarquer que les deux méthodes proposaient des scores équivalents. Cependant, elles ont chacune leurs avantages et leurs inconvénients.

La méthode statique est simple à implémenter, très rapide (il suffit, par définition, d'une seule passe sur les données d'apprentissage) et permet de gérer artificiellement l'amortissement temporel de la mémoire du modèle (à l'aide du paramètre α). Mais contrairement à la descente de gradient stochastique, elle ne permet pas la mise en place et l'apprentissage de modèles plus élaborés comme l'application d'une métrique à l'espace de représentation des items (c.f. section 6.5).

6.4 Approche Communautaire

Comme dit précédemment, lorsque nous avons essayé d'ajouter une transformation de l'espace des représentations modélisant les goûts de l'utilisateur, nous nous sommes heurtés à un problème de données. Les traces utilisateur étaient en moyenne courtes (allant de 17 à 66 éléments en moyenne, suivant le jeu de données). Nous avons donc adopté une famille de transformations simples, les translations. Pour aller plus loin et définir des transformations plus complexes, nous nous sommes tournés, vers une stratégie d'agrégation des utilisateurs.

Nous avons donc ajouté un contexte communautaire. Pour ce faire, nous avons segmenté la base des utilisateurs en fonction de leurs traces. Pour chaque communauté ainsi définie, nous avons appris une métrique sur l'espace des items permettant d'adapter leurs représentations aux goûts des communautés. La métrique associée à une communauté c est la matrice A_c, de dimension $d \times d$.

6.4.1 Extraction des Communautés

Chaque utilisateur est représenté par un vecteur cl_u défini comme il suit :

$$\forall u \in U \begin{cases} cl_u[i] = 1 & ssi \;\; i \in \tau_u \\ cl_u[i] = 0 & ssi \;\; i \notin \tau_u \end{cases} \tag{6.8}$$

Pour mettre en évidence l'importance de l'information disponible dans les communautés, nous avons décidé d'utiliser une méthode de segmentation simple, l'algorithme des k-moyennes. On appelle c_u la communauté associée à l'utilisateur u.

6.4.2 Utilisation des Communautés

Soit c_u la communauté associée à l'utilisateur u, on applique sa métrique à l'ensemble de l'espace de représentation des items. On définit alors une version des vecteurs item dépendante de la communauté, la transformation associée à l'utilisateur restant, elle, inchangée :

$$\begin{aligned} \psi_{i_{c_u}} &= A_{c_u}\psi_i, & A_{c_u} &\in M_n(\mathbb{R}) \\ \Phi_u(\psi_{i_t}) &= A_{c_u}\psi_{i_t} + b_u, & b_u &\in \mathbb{R}^d \end{aligned} \tag{6.9}$$

Ici, la matrice A_c représente l'endomorphisme associé aux communautés. Le modèle est alors appris par descente de gradient sur une version mise à jour de la fonction Eq.6.7 :

$$\mathcal{L} = \sum_{\substack{\tau\in\mathcal{T} \\ i\in\tau j\in I\setminus i}} \left[1 - \|\Phi_u(\psi_{i_{c_u,t-1}}) - \psi_{j_{c_u}}\|^2 + \|\Phi_u(\psi_{i_{c_u,t-1}}) - \psi_{i_{c_u,t}}\|^2 \right]^+ + \lambda\Omega(\Phi) \tag{6.10}$$

Le terme de régularisation est définit comme suit :

$$\Omega(\Phi) = \sum_{u\in U} \|b_u\|^2 + \|A_c\|_F^2 \tag{6.11}$$

6.5 Approche Collaborative

Durant ces travaux de thèse, nous ne cherchions pas seulement à être capable de prédire le prochain item que l'utilisateur irait voir, mais aussi son appétence pour ce dernier. Par conséquent, il était donc nécessaire à notre système d'être capable de quantifier cette appétence. Pour ce faire, nous proposons d'aussi évaluer notre système sur la tâche de prédiction de notes.

Nous avons précédemment défini des représentations, aussi bien des utilisateurs (b_u) que des items (ϕ_i). Ces représentations sont supposément proches pour les utilisateurs (respectivement les items) semblables, c'est pourquoi une approche par filtrage collaboratif nous paraissait s'imposer d'elle même.

6.5.1 Enrichissement de la Factorisation Matricielle

Nous voulons ici améliorer les résultats de la factorisation matricielle en tirant un maximum d'information des représentations utilisées pour la prédiction d'items. Si cette tâche n'est pas évidente car il n'y a pas de corrélation directe entre la trace utilisateur et les notes attribuées aux items, nous pensons toutefois que lesdites traces contiennent une quantité d'information non négligeable quand à la similarité entre les utilisateurs d'un côté mais aussi entre les items. Les nouveaux profils sont définis de la façon suivante pour le modèle à personnalisation simple :

$$\gamma_u = \begin{bmatrix} \bar{\gamma_u} \\ b_u \end{bmatrix} \in \mathbb{R}^{2d}, \qquad \gamma_i = \begin{bmatrix} \psi_i \\ \bar{\gamma_i} \end{bmatrix} \in \mathbb{R}^{2d} \tag{6.12}$$

et de la manière suivante pour le modèle communautaire :

$$\gamma_u = \begin{bmatrix} \bar{\gamma_u} \\ b_u \end{bmatrix} \in \mathbb{R}^{2d}, \qquad \gamma_i = \begin{bmatrix} A_{c_u}\psi_i \\ \bar{\gamma_i} \end{bmatrix} \in \mathbb{R}^{2d} \tag{6.13}$$

Grâce à cette modélisation, nous assurons une certaine proximité entre les utilisateurs (*resp.* items) avec un b_u (*resp.* ψ_i) proche tout en calculant un score par factorisation matricielle (Eq. 6.14) :

$$g(u,i) = \phi_u.\psi_i \tag{6.14}$$

Les représentations $\{\bar{\gamma_u}, u \in U\}$ et $\{\bar{\gamma_i}, i \in I\}$ sont initialisées aléatoirement et apprises par descente de gradient sur la fonction de coût :

$$\mathcal{L} = \frac{1}{m} \sum_{(u,i)\in App} (\gamma_u \cdot \gamma_i) + g_0 + g_1(u) + g_2(i) + \lambda_U \left\|\hat{\gamma_u}\right\|_F^2 + \lambda_I \left\|\hat{\gamma_i}\right\|_F^2 \tag{6.15}$$

b_u, A_c et ψ_i restent constantes durant tout le processus d'apprentissage pour assurer la proximité des profils similaires.

6.6 Évaluation

Dans cette section, nous proposons d'évaluer les performances de nos modèles pour deux tâches différentes : la prédiction d'items, et la prédiction de notes. Comme

expliqué dans la section 6.6, nous utilisons une mesure de rappel@k pour la prédiction d'items, et le critère d'erreur quadratique moyenne (MSE) dans le cadre de la prédiction de notes.

Jeux de données Nous avons effectué nos expériences sur cinq jeux de données : *BeerAdvocate* et *RateBeer* proviennent de [ML13a] et contiennent des revues de bières. MovieLens (10M), (Amazon) Movies et Flixster traitent de films. Les propriétés de chaque jeu de données sont répertoriées dans la Table 6.1.

Table 6.1: Propriétés des jeux de données.

Jeu de Données	# items	# utilisateurs	# notes
BeerAdvocate	66 051	33 387	1 586 259
RateBeer	110 419	40 213	2 924 127
MovieLens	10 000	72 000	10 000 000
Flixster	49 000	1 000 000	8 200 000
Movies	253 059	889 176	7 911 684

Nous avons aussi utilisé deux jeux de données supplémentaires pour l'analyse qualitative du modèle à entrée-sortie IOLM présenté en section Sec.6.2.1 : un jeu de données Flickr-Pisa contenant les traces de touristes lors de leur visite de la ville de Pise [Bar+13], ainsi que le jeu de données AMMICO [GS+14], contenant les traces des visiteurs de l'exposition Great Black Music lors de son installation à la Réunion.

Table 6.2: Descriptif des jeux de données AMMICO et Flickr-Pisa explicitant leur nombre d'items (POI) et de parcours-utilisateur (traces)

Jeu de données	#POI	#traces
AMMICO	69	3713
Pisa	110	992

Cadre Général Nous avons normalisé les notes sur l'ensemble $[0, 5] \subset \mathbb{N}$ et utilisé le cadre proposé dans [Zha+14a] : nous n'avons gardé que les traces utilisateur contenant au moins dix revues et chaque jeu de données a été divisé en dix fenêtres temporelles de même durée. Pour apprendre les hyperparamètres du modèle, nous avons utilisé les huit premières fenêtres comme base d'apprentissage et la neuvième comme base de validation. Une fois les hyperparamètres appris, nous avons ajouté la fenêtre de validation à l'ensemble d'apprentissage et utilisé la dernière fenêtre comme ensemble de test. Pour minimiser l'influence des processus stochastiques (l'initialisation aléatoire et la descente de gradient stochastique), nous avons lancé chaque expérience 5 fois et rapporté la moyenne des résultats.

Nous présentons le rappel@k pour k variant entre 5 et 50. Nous pensons que des valeurs plus élevées de k n'ont que peu d'intérêt dans le cadre de la recommandation.

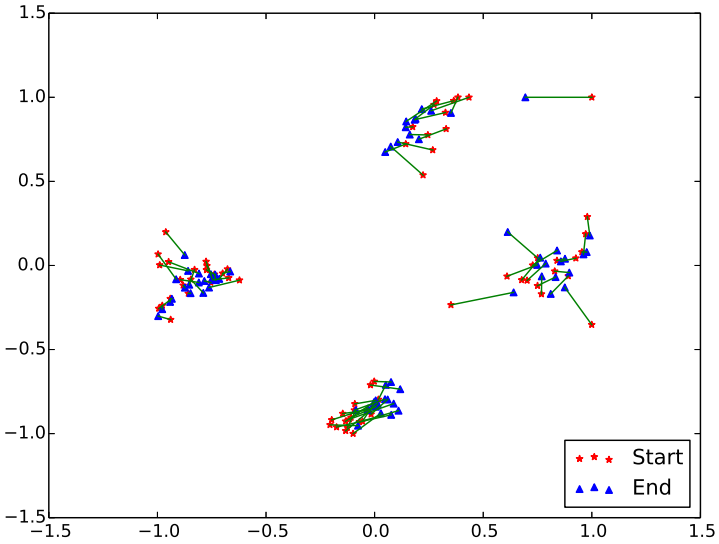


Figure 6.5: . Représentation des POI du jeu de données AMMICO dans un espace latent 2D

6.6.1 Prédiction d'Items

Modèles de Référence Dans le cadre de la prédiction d'items, nous utilisons trois modèles de référence, IOLM et Word2Vec, présentés en 6.2 et RankALS [TT12] présenté en 3.2.2.

Par la suite, on appelle TRANS le modèle de personnalisation à translation et COMM celui prenant en compte les communautés.

Lorsque l'on utilise un espace latent, il est nécessaire de déterminer sa dimension optimale. Nos expérimentations sur l'ensemble de validation montrent que l'impact de la dimension diminue pour $d > 20$. Nous avons donc fixé la dimension à 20 pour toutes nos expériences.

Hyperparamètres et Analyse Qualitative

Modèle à Entrée/Sortie (IOLM) L'étude de la répartition des POI dans un espace latent 2D proposée en figure 6.5 et figure 6.6 montre la capacité du modèle à prendre en compte les dynamiques utilisateur.

L'exposition Great Black Music (figure 6.5) était divisée en quatre salles. Si chaque salle pouvait se visiter dans n'importe quel ordre, l'ordre des salles était quand à lui fixe, ce que l'on retrouve clairement dans la modélisation. En effet, le cluster nord, qui représente la première salle, pointe vers la seconde salle (le cluster ouest), c'est

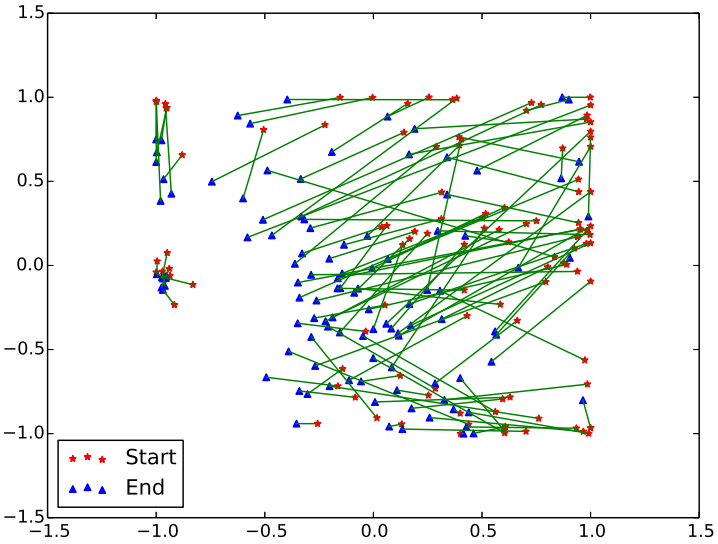


Figure 6.6: Représentation des POI du jeu de données Pisa dans un espace latent 2D

à dire que les vecteurs entrée-sortie des items contenus dans le cluster nord sont dirigés vers le cluster ouest. Ce cluster pointe lui-même vers la troisième salle (le cluster sud). Enfin, le cluster sud pointe vers la dernière salle (cluster est) qui pointe vers elle-même.

Il en va de même pour le jeu de données des traces de touristes dans la ville de Pise (6.6) : on remarque que tous les POIs pointent vers la même région. Cette région de l'espace latent est presque intégralement composée de POIs appartenant au voisinage de la tour, c'est à dire la zone touristique la plus dense et la plus visitée de la ville. On retrouve donc bien le comportement classique des touristes visitant Pise.

Modèle à Trajectoire(TRANS) Pour ce modèle, nous avons aussi du utiliser l'ensemble de validation pour optimiser les valeurs de α. L'évolution du rappel@30 en fonction des valeurs de α sur le jeu de données BeerAdvocate est donné en Fig.6.7.

On remarque que les meilleurs résultats sont obtenus pour $\alpha \approx 0.3$ avec une très forte détérioration des performances pour $\alpha \geq 0.5$. Ceci signifie que s'il est important de garder une mémoire des items rencontrés précédemment, il est tout aussi primordial d'atténuer le poids d'interactions trop anciennes, ce qui s'apparente à une forme d'oubli. Ce comportement conforte le parti pris dans [ML13a]. En effet, ici, l'idée d'oubli est cohérente avec le principe d'expérience : au fur à mesure que les goûts de

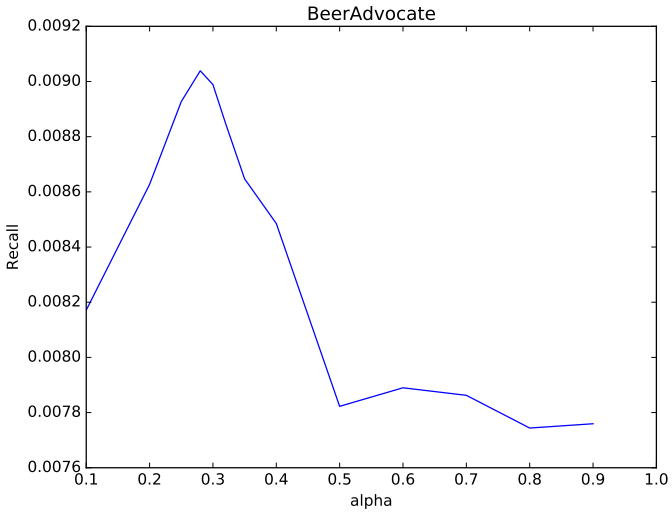


Figure 6.7: Evolution du rappel@30 pour $\alpha \in [0, 1]$ sur le jeu de données BeerAdvocate.

l'utilisateur s'affinent, il s'éloigne de ses premiers amours pour s'orienter vers des produits plus adapté à son palais plus aiguisé.

Détermination du Nombre de Communautés Nous avons effectué de nombreuses expériences sur les ensembles de validation pour déterminer le nombre optimal de communautés pour chaque jeu de données. Les résultats sont présentés en Fig. 6.8. On remarque que le nombre de communauté optimal est un paramètre très dépendant du jeu de données utilisé et ne semble pas être directement lié au domaine ou à la taille dudit jeu de données.

6.6.2 Prédiction de Notes

Pour cette tâche nous avons utilisé trois modèles de référence : la factorisation matricielle classique avec termes de biais telle que proposée dans [Kor+09] (MF), une extension temporelle de ce modèle [Kor09b] (TSVD) et un modèle à niveaux d'expérience [ML13a] (EXP).

Pour ce dernier modèle, nos expériences sur l'ensemble de validation nous ont permis de fixer le nombre de niveaux d'expérience à 5 (comme dans l'article d'origine). En conséquence, ce modèle contient cinq fois plus de paramètres que MF. CBOW est calculé sur un espace latent de dimension 20, TRANS et COMM sont donc en dimension 40. Dans un souci d'équité, nous avons évalué tous nos modèles dans un espace de dimension $d = 40$.

6.6.3 Analyse Quantitative

6.6.4 Prédiction d'Items (rappel@k)

Pendant nos travaux préliminaires, nous avons considéré plusieurs mesures d'évaluation comme la précision@k ou le rang moyen, cependant ces méthodes semblaient mal adaptées. Le rang moyen est peu robuste. D'un autre côté, la précision@k est trop stricte. En effet, un utilisateur peut avoir raté un item simplement parce qu'il ne le connaissait pas et non par manque d'intérêt. Il fallait donc une méthode plus souple pour une évaluation hors ligne. C'est pourquoi nous avons utilisé le rappel@k, comme préconisé dans [Zha+14a].

La Figure 6.9 contient les courbes de performance en rappel@k pour les différents modèles. On, remarque que, dans l'ensemble, les modèles temporels (COMM, TRANS, CBOW et IOLM) donnent de meilleurs résultats que RankALS, ce qui montre l'importance du contexte temporel. De plus, CBOW propose de meilleures performances ainsi que moins de paramètres que IOLM (sauf sur le jeu de données Movies).

Les modèles personnalisés (TRANS et COMM) tiennent le haut du tableau, spécialement COMM, qui écrase tous les autres modèles. Cette démonstration de l'impact des informations sur les communautés est encourageante pour la suite de nos travaux.

6.6.5 Prédiction de Notes (MSE)

Nous avons utilisé la méthode classique d'évaluation pour la prédiction de notes, le critère d'erreur quadratique moyenne dont l'adéquation n'est plus à démontrer.

Les résultat sont disponibles dans la Table 6.3. Les meilleurs résultats sont surlignés en gras.

Table 6.3: Résultats en prédiction de notes, exprimés en MSE pour les modèles MF, TSVD, EXP, TRANS et COMM.

Dataset	MF	TSVD	EXP	TRANS	COMM
BeerAdvocate	0.4	0.381	0.367	**0.361**	**0.360**
RateBeer	0.331	0.301	0.297	**0.279**	**0.279**
MovieLens	0.691	0.681	0.684	0.663	**0.660**
Flixster	0.912	0.867	0.827	0.816	**0.811**
Movies	1.377	1.211	1.05	**0.913**	**0.913**

Comme mis en avant dans [Kor09b], la marge d'amélioration en terme de MSE est très faible et même un faible gain peut avoir un énorme impact sur la recommandation. Tout d'abord, on remarque que les modèles temporels surpassent clairement

MF ce qui encore une fois montre l'intérêt de la prise en compte du contexte temporel. De plus nos modèles surpassent clairement TSVD et EXP, démontrant l'ajout d'information que représente l'utilisation des représentations de trajectoire.

De plus l'ajout des représentations de COMM permet d'améliorer encore les résultats de TRANS sur la plupart des jeux de données. Même si MF fonctionne par segmentation implicite des utilisateurs en communautés, on voit ici que l'ajout d'information explicite améliore encore les performances.

Il est intéressant de se pencher sur la question de la complexité spatiale : TRANS et COMM comptent moitié moins de paramètres que MF (la moitié de leurs paramètres sont constants et appris en amont) alors que EXP en compte n fois plus (avec n le nombre total de niveaux d'expérience). Ceci montre que même la plus simple des approches a assez de degrés de libertés. Le défi n'est pas la complexification, mais plutôt l'adoption de la stratégie la plus intelligente.

Le comportement des modèles d'un jeu de données à l'autre est somme toute stable. Certains présentent un léger gain, mais aucun domaine ne semble particulièrement privilégié par notre approche.

6.7 Conclusion

Dans ce chapitre, nous avons montré qu'il était possible de tirer beaucoup d'information de la caractérisation des transitions entre items pour chaque utilisateur, permettant ainsi d'améliorer les performances des modèles de recommandation, aussi bien pour la prédiction d'items que pour la prédiction de notes.

De plus, nous avons prouvé l'intérêt de l'ajout d'un contexte communautaire explicite. En apprenant une métrique propre à chaque communauté, nous avons pu adapter les représentations latentes des items et ainsi faciliter la modélisation des dynamiques de l'utilisateur.

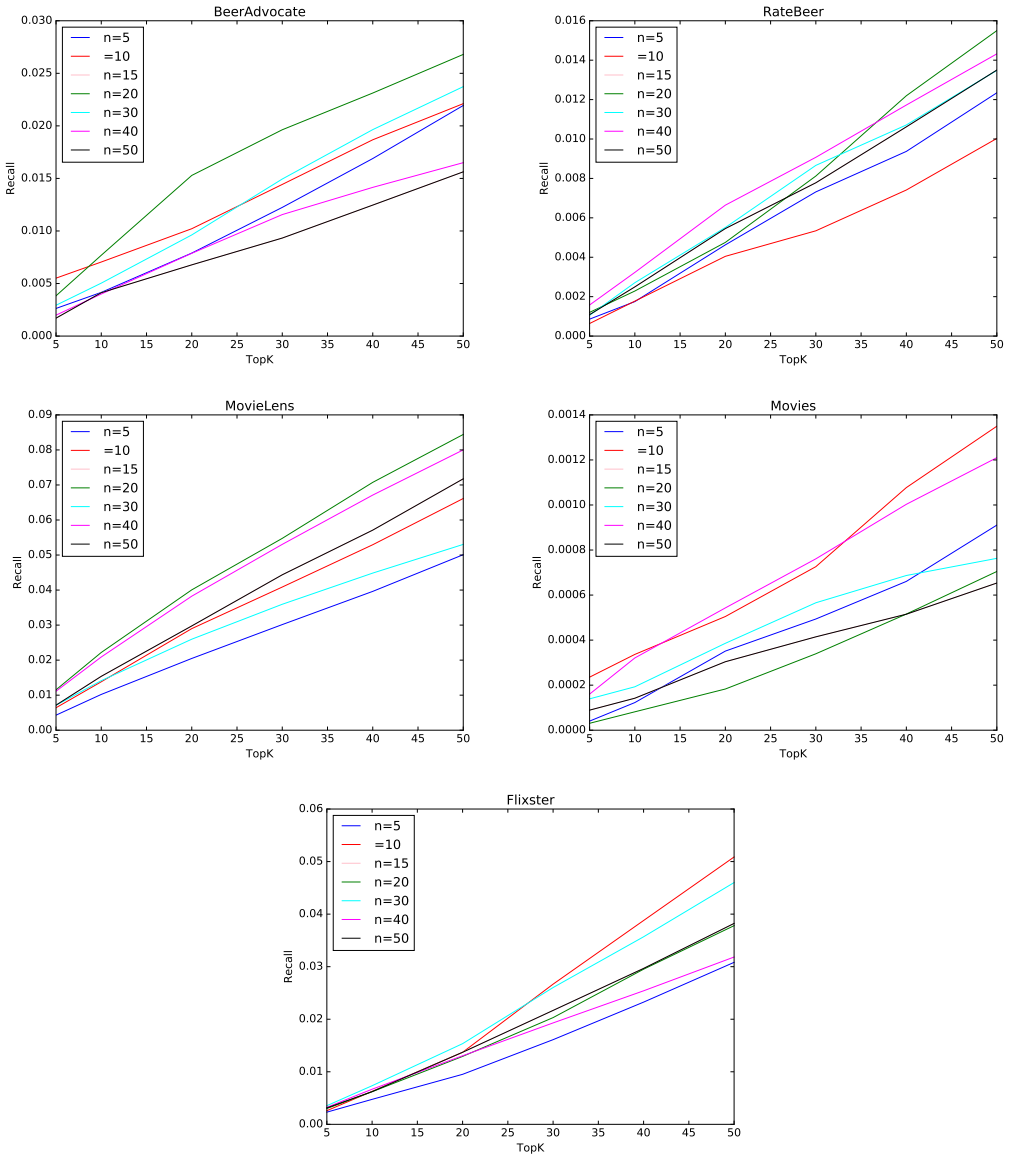


Figure 6.8: Evolution du rappel@k sur les ensembles de validation pour différents nombres de communautés $n \in [5, 50]$.

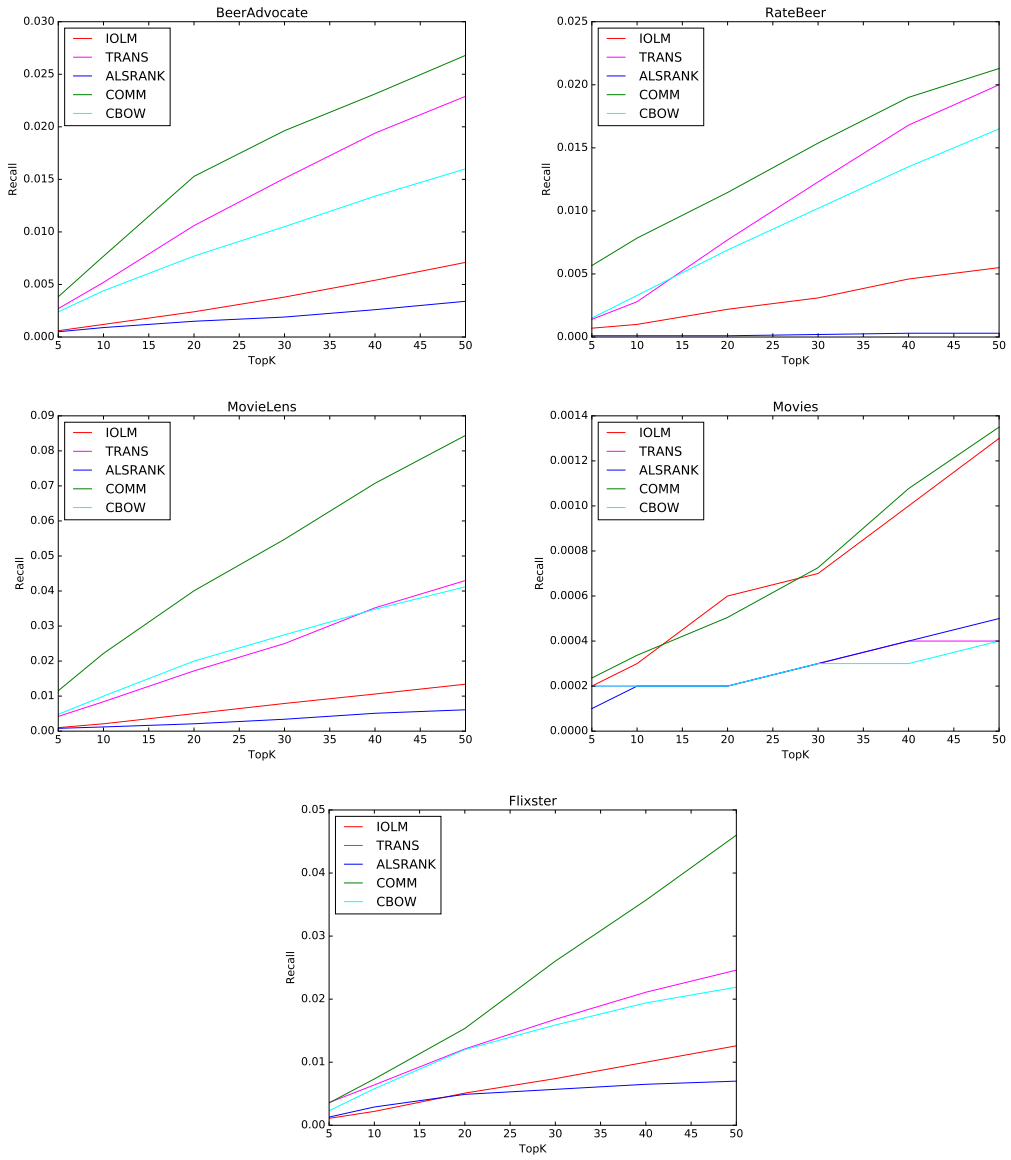


Figure 6.9: Evolution du rappel@k sur les ensembles de test pour chaque modèle.

7 Conclusions et Perspectives

> *Sometimes science is more art that science, most people don't get that*
>
> — **Rick**

7.1 Conclusions

Au cours de ce travail de thèse, nous avons cherché à remettre l'utilisateur au centre de l'accès à l'information. Nous avons travaillé à enrichir son profil, en explorant principalement deux pistes : l'utilisation de données textuelles, et d'informations temporelles. Mais nous avons aussi contribué à faire évoluer les tâches de recommandation en mettant l'accent sur la prédiction du comportement de l'utilisateur d'une part et sur l'explication associée à la recommandation d'autre part.

7.1.1 Texte et Marqueurs d'Opinion

Dans les deux premiers chapitres, nous avons démontré l'utilité des marqueurs d'opinion dans le cadre de la recommandation. Le gain apporté par ces marqueurs se divise en trois axes : la gestion du démarrage à froid, à travers l'ajout de nouveaux produits, le gain en performances, et enfin l'accompagnement de l'utilisateur.

Dans une première publication [GS+13], nous avons utilisé les données textuelles pour en extraire des marqueurs d'opinion et ainsi prédire, à l'aide de *tweets*, les résultats de films au box office. La différence d'ordre de grandeur entre les résultats de l'approche volumétrique et de celle basée sur les marqueurs d'opinion démontre l'intérêt de l'utilisation de ces marqueurs dans le cadre d'une tâche de prédiction. Ainsi, nous préconisons l'utilisation des descripteurs d'opinion sous la forme d'un a priori semblable au biais item décrit dans la section 3.2.2 dans le cadre d'ajouts de nouveaux produits au catalogue. De plus, contrairement aux apparences, cette tâche de prédiction est très proche de celle de la recommandation en effet, la prédiction de résultats au box office, comme la recommandation vise à prédire le comportement des gens. La seule différence se situe dans la granularité. Quand la recommandation

s'intéresse à l'individu, la prédiction de résultats au box-office s'intéresse à une population.

Dans une seconde publication [Pou+14], nous avons tout d'abord proposé d'enrichir le filtrage collaboratif par factorisation matricielle à l'aide de texte brut. Cet enrichissement, réalisé à l'aide d'un modèle d'analyse d'opinion simple (représentation d'un utilisateur/produit par trois documents textuels générés par concaténation de toutes ses revues, ses revues positives et ses revues négatives) permet d'affiner la prédiction de notes. Les expériences que nous avons réalisées sur des bases de données de différentes tailles issues de *ratebeer.com* et *amazon.com* montrent que l'amélioration des performances est liée à une meilleure estimation des bonnes notes (4 et 5). Comme elles sont les plus nombreuses, elles permettent, par leur masse, une bonne estimation des attentes des utilisateurs et qualités des objets mais aussi des gains significatifs si elles sont mieux prédites.

Toujours dans [Pou+14], nous avons cherché à apporter de l'explicativité aux résultats en proposant à l'utilisateur une critique personnalisée du produit recommandé. Cette approche, très novatrice reste cependant difficile à évaluer. En effet, les résultats présentés montrent bien les limitations des métriques ROUGE-n dans ce cadre-ci.

7.1.2 Données Temporelles et Dynamique

Durant ces travaux de thèse, nous nous sommes intéressés à des paradigmes de recommandation où l'utilisateur prend une part active dans le processus d'accès à l'information tout d'abord dans lors de visites touristiques, que ce soit dans des villes ou dans les musées (l'utilisateur choisit un parcours qui sera affiné au fur et à mesure par des recommandations) ; puis dans [Guà+15], lors de visites de sites spécialisés ou de boutiques en ligne (l'utilisateur peut utiliser la barre de recherche ou l'arborescence des catégories).

Dans ces travaux nous avons voulu utiliser les données temporelles comme source d'information sur la dynamique des utilisateurs. Les premières approches que nous avons envisagées utilisaient une modélisation continue ou discrète du temps et se sont révélées peu efficaces. Nous avons donc opté pour une modélisation plus simple, mais aussi plus robuste : nous avons restreint le problème à la considération de l'ordre dans les traces des utilisateurs (*i.e.* la séquence des items avec lesquels il a interagi).

Dans [GS+14] nous nous sommes concentrés sur l'assistance aux visites touristiques aussi bien dans les villes que dans les musées. Face à la taille des données disponibles,

nous n'avons pas pu mettre en place de personnalisation et nous sommes concentrés sur la création d'un espace de représentation des items permettant toutefois de prendre en compte la dynamique des utilisateurs. Dans les deux cas d'application présentés en Fig. 6.6 (recommandation de points d'intérêts dans la ville de Pise) et en Fig. 6.5 (recommandation d'œuvres dans l'exposition Great Black Music), on remarque que l'ajout de la dynamique utilisateur à la création de l'espace latent permet de retrouver la topologie des lieux : dans le cas de Pise, on retrouve tous les PoI du centre historique dans une même zone de l'espace alors que les autres PoI sont disposés de manière plus espacée dans la périphérie. De plus on remarque que les transitions tendent à aller de l'extérieur vers le centre, et à rester dans le centre, ce qui correspond au parcours classique d'un touriste à Pise. De la même façon, pour l'exposition Great Black Music, on retrouve bien le plan de l'exposition divisée en quatre salles ainsi que les transitions de salle en salle.

Dans [Guà+15], nous nous sommes intéressés au problème de recommandation sur les sites spécialisés et les boutiques en ligne. Le volume de données disponibles étant significativement plus grand que dans le cas précédent, nous avons pu mettre en place des méthodes de personnalisation. Contrairement aux approches classiques, nous n'avons pas cherché à représenter l'utilisateur comme un point de l'espace de représentation mais comme une fonction de transition entre les items. Lors de l'implémentation nous avons cherché à modéliser l'utilisateur comme une transformation affine de type $A_u x + b_u$ où A_u est une matrice carrée et b_u un vecteur. Cependant, les traces utilisateur étant en moyenne trop courtes, nous avons du diviser la personnalisation en deux méthodes de granularité différentes : si le vecteur de translation b_u reste au niveau de l'utilisateur, la métrique A, elle passe au niveau des communautés d'utilisateurs. Outre le fait de proposer une légère amélioration des performances, l'utilisation de communautés a permis d'ajouter une dimension collaborative au calcul des recommandations. De plus, nous avons enrichi le problème proposé dans [GS+14] : nous ne cherchions plus seulement à prédire vers quel produit un utilisateur se dirigerait, mais aussi son appétence pour ce produit. Dans la pratique ceci s'est traduit par l'ajout d'une tâche de prédiction de notes à notre système de recommandation. Pour répondre à cette tâche de prédiction de notes, nous avons enrichi une factorisation matricielle classique à l'aide des données calculées pour la prédiction d'items. Nous pensons que deux produits proches dans l'espace (resp. deux utilisateurs avec des fonctions de transition similaires) auront de fortes chances d'être notés (resp. de noter) de la même façon. Les résultats présentés vérifient cette hypothèse, mais seulement jusqu'à un certain point, en effet, si l'ajout de l'information communautaire avait un effet bénéfique sur la prédiction de produits, elle n'a qu'un effet somme toute marginal sur la prédiction de notes. Nous pensons ceci dû au fait que la factorisation matricielle extrait déjà implicitement une partie de cette information, limitant la marge de gain sur cet aspect.

7.2 Discussion et Perspectives

7.2.1 Enrichissement des Profils et Problèmes Éthiques

Aujourd'hui, l'avancée de l'accès à l'information passe majoritairement par deux axes, la définition du profil utilisateur et celle du contexte attenant à la recommandation. Chacune de ces deux approches commence par la même étape, la récupération de données utilisateurs toujours plus riches (*e.g* sexe, age, CSP, géolocalisation, nationalité). Le stockage de ces informations par de grandes entreprises telles que Google, Facebook ou Foursquare, ou même des États soulève à lui seul de sérieuses questions éthique, légales et morales dont l'importance ne cessera d'augmenter au cours des années à venir. On pourra déjà en relever deux :

— Si le cadre légal demande à toute entreprise d'informer ses utilisateurs du stockage et de l'utilisation de leurs information personnelles, la position de force des acteurs majeurs du web oblige l'utilisateur à les accepter. Dans un monde hyperconnecté comme le notre, les refuser signifierait accepter de se passer du confort de son smartphone (liés à un compte Google, Apple ou Windows) ou de l'achat en ligne, mais aussi de besoins aujourd'hui jugés vitaux comme l'accès au réseau électrique (*cf.* l'adoption forcée du compteur communiquant *Linky*). Il est donc important de réfléchir à ce que serait le pouvoir de modélisation utilisateur acceptable, car pour chaque avancée que présente la modélisation toujours plus fine de l'utilisateur, il existe une dérive. Par exemple, en appliquant les méthodes de détection de pannes aux profils utilisateur, il serait possible de prévoir les maladies, les traiter plus tôt, voire permettre une médecine plus préventive. Seulement, cela signifie aussi que les assurances de santé pourraient prévoir qu'un client risque de leur coûter trop cher et donc augmenter ses mensualités en conséquence, voire le radier purement et simplement. Et si en France et en Europe (contrairement aux États Unis par exemple), le stockage et le partage de ces données utilisateur est sévèrement contrôlé par des organismes tels que la CNIL (en France), les actions pénales disponibles à ces organismes ne sont pas adaptés aux grands groupes du web comme Google, Facebook ou Amazon. Par exemple, en 2016, Google a été condamné à payer une amende de 100 000 euros et Meetic à une amende de 20 000 euros. Aux vues du chiffre d'affaire annuel de ces deux entreprises, il est clair que le caractère dissuasif (ou punitif) de ces actions est au mieux marginal, voir nul.

— Un système pro-actif influence (par définition) son écosystème. Si on prend l'exemple de l'accès à l'information politique, la modélisation de l'utilisateur pour améliorer les résultats qu'on lui remonte implique de construire un *a priori* sur ses opinions. Il serait donc possible d'utiliser ses marqueurs pour

prédire ses intentions de vote et son affiliation politique. Dans le cadre d'un État totalitaire, c'est la porte ouverte à une nouvelle forme de propagande, moins évidente et plus intrusive. Si nous sommes encore loin des dystopies d'Orwell et d'Huxley, les dernières élections présidentielles, Américaines comme Françaises nous ont permis de voir poindre ce genre de problématiques. En effet un des problèmes souvent remonté est le prisme qu'a pu représenter Facebook pour des gens dont les réseaux sociaux sont la source majeure d'information : dans un réseau social, les utilisateurs tendent à être plus fortement connectés à d'autres utilisateurs partageant les mêmes idées et appartenances politiques. Ainsi, un utilisateur n'aura accès sur son mur qu'à des informations validant son point de vue, ne laissant que peu de place au débat. Cette année, ce phénomène a été fortement accentué par la prolifération d'articles de désinformation (via les *fake news*) dont l'impact sur l'élection américaine a été jugé important à défaut d'avoir été décisif [AG17].

7.2.2 Perspectives

Le Profilage Universel Aujourd'hui, les réseaux de neurones récurrents (*e.g.* Long Short-Term Memory, Neural Machine Translation) bouleversent la façon que l'on a d'appréhender les données textuelles et, plus précisément, les tâches de traduction automatique. Lorsqu'il y a encore quelques années, elles étaient vu comme un problème d'alignement, aujourd'hui, nous sommes passé à une logique d'encodeur-décodeur, où l'espace latent représenterait une sorte de méta-langage universel. Est-il possible d'appliquer cette logique à la recommandation ? Existe-t-il une représentation universelle de chaque utilisateur ? Au lieu de donner toujours plus d'informations à toujours plus d'entreprises, ne pourrions-nous pas avoir une représentation publique, dans laquelle ne figureraient que les informations que nous sommes prêts à divulguer que nous transporterions avec nous au cours de nos navigations sur Internet ? Ce genre de proposition semble prometteuse, notamment sur le plan éthique. Cependant, si on laisse l'utilisateur remplir son profil, à la manière d'une revue d'opinion, celui-ci le fera-t-il, sera-t-il honnête sur ses goûts ou un thésard de trente ans sera-t-il tenté de désavouer son amour pour Starmania ?

Une Recommandation Moins Intrusive... Dans [Teo+16], *Teo et al.* proposent un nouveau paradigme de recommandation : au lieu de pousser des produits à un utilisateur, le système lui propose des vitrines au travers desquelles il pourra flâner, rendant l'expérience moins intrusive. Comme nous l'avons expliqué au cours de ce manuscrit, nous pensons qu'aujourd'hui la problématique de recommandation se joue plus sur l'accompagnement de l'utilisateur dans son accès à l'information que dans des modèles de *push* classique. C'est pourquoi ce genre de travaux nous parle particulièrement. Une perspective d'ouverture serait d'appliquer notre modèle

à trajectoires à ce genre de problématique. En effet, la trajectoire de l'utilisateur ne dépendant que de sa trace, il est aisé de la recalculer en temps réel, mettant ainsi à jour ses préférences tout au long de son exploration.

... Voire sur Demande Les dernières années ont vu l'avènement des assistant électroniques tels que SIRI (Apple), CORTANA (Windows) ou ALEXA (Amazon). Grâce aux *memory networks*, ces systèmes sont capables de mieux en mieux gérer les problématiques du langage naturel, semblant esquisser un futur où l'interface homme-machine serait un chatbot. Dans ce genre de configuration, la recommandation sera-t-elle vraiment différente de la recherche d'information ? Ou sera-t-elle un système pour répondre aux questions trop ouvertes. *Hey Cortana, trouve-moi un bon bar pour un pot de thèse.*

Bibliographie

[Abb+15] Assad ABBAS, Limin ZHANG et Samee U. KHAN. « A Survey on Context-aware Recommender Systems Based on Computational Intelligence Techniques ». In : *Computing* 97.7 (juil. 2015), p. 667–690 (cf. p. 3).

[Ado+11] Gediminas ADOMAVICIUS, Bamshad MOBASHER, Francesco RICCI et Alexander TUZHILIN. « Context-Aware Recommender Systems ». In : *AI Magazine* 32.3 (2011), p. 67–80 (cf. p. 25, 28).

[AG17] Hunt ALLCOTT et Matthew GENTZKOW. *Social Media and Fake News in the 2016 Election*. Working Paper 23089. National Bureau of Economic Research, 2017 (cf. p. 93).

[AH10] S. ASUR et B. A. HUBERMAN. « Predicting the Future With Social Media ». In : *ACM IC Web Intelligence*. 2010 (cf. p. 37).

[Ama+96] S. AMARI, A. CICHOCKI et H. H. YANG. « A New Learning Algorithm for Blind Signal Separation ». In : *Advances in Neural Information Processing Systems*. MIT Press, 1996, p. 757–763 (cf. p. 8).

[AT01] Gediminas ADOMAVICIUS et Alexander TUZHILIN. « Multidimensional Recommender Systems : A Data Warehousing Approach. » In : *WELCOM*. Sous la dir. de Ludger FIEGE, Gero MÜHL et Uwe G. WILHELM. T. 2232. Lecture Notes in Computer Science. Springer, 2001, p. 180–192 (cf. p. 27).

[AT05] Gediminas ADOMAVICIUS et Alexander TUZHILIN. « Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions ». In : *IEEE trans. on Knowledge and data engineering* 17.6 (2005), p. 734–749 (cf. p. 18, 21, 27, 33, 34, 54).

[AT08] Gediminas ADOMAVICIUS et Alexander TUZHILIN. « Context-aware Recommender Systems ». In : *Proceedings of the 2008 ACM Conference on Recommender Systems*. RecSys '08. Lausanne, Switzerland : ACM, 2008, p. 335–336 (cf. p. 27).

[Bar+13] Ranieri BARAGLIA, Cristina Ioana MUNTEAN, Franco Maria NARDINI et Fabrizio SILVESTRI. « LearNext : learning to predict tourists movements ». In : *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM. 2013, p. 751–756 (cf. p. 81).

[BC92] Nicholas J. BELKIN et W. Bruce CROFT. « Information filtering and information retrieval : two sides of the same coin ? » In : *Commun. ACM* (1992) (cf. p. 20).

[Bel+11] Alejandro BELLOGÍN, Pablo CASTELLS et Iván CANTADOR. « Precision-oriented evaluation of recommender systems : an algorithmic comparison ». In : *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*. 2011, p. 333–336 (cf. p. 19, 34).

[Ben+06] Emmanouil BENETOS, Margarita KOTTI et Constantine KOTROPOULOS. « Musical Instrument Classification using Non-Negative Matrix Factorization Algorithms and Subset Feature Selection ». In : *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006, Toulouse, France, May 14-19, 2006*. 2006, p. 221–224 (cf. p. 13).

[Ben+07] James BENNETT, Stan LANNING et Netflix NETFLIX. « The Netflix Prize ». In : *In KDD Cup and Workshop in conjunction with KDD*. 2007 (cf. p. 18, 34).

[Ber+07] O. BERNÉ, C. JOBLIN, Y. DEVILLE et al. « Analysis of the emission of very small dust particles from Spitzer spectro-imagery data using blind signal separation methods ». In : *Astronomy and Astrophysics - A&A* 469 (juil. 2007). 14 pages, 11 figures, to appear in A&A, p. 575–586 (cf. p. 8).

[Bes+11] Dmitriy BESPALOV, Bing BAI, Yanjun QI et Ali SHOKOUFANDEH. « Sentiment classification based on supervised latent n-gram analysis ». In : *ACM CIKM*. 2011, p. 375–382 (cf. p. 39).

[Ble+03] David M. BLEI, Andrew Y. NG, Michael I. JORDAN et John LAFFERTY. « Latent dirichlet allocation ». In : *Journal of Machine Learning Research* 3 (2003), p. 2003 (cf. p. 7).

[Bli+06] John BLITZER, Ryan MCDONALD et Fernando PEREIRA. « Domain adaptation with structural correspondence learning ». In : *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. EMNLP '06. Sydney, Australia : Association for Computational Linguistics, 2006, p. 120–128 (cf. p. 38–40, 42, 47).

[Boh10] Fabian BOHNERT. « Personalising the Museum Experience ». In : *Proceedings of the 2010 Workshop on Pervasive User Modeling and Personalization (PUMP-10), held in conjunction with the 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP-10)*. 2010, p. 33–36 (cf. p. 73).

[Bou+14] Simon BOURIGAULT, Cedric LAGNIER, Sylvain LAMPRIER, Ludovic DENOYER et Patrick GALLINARI. « Learning Social Network Embeddings for Predicting Information Diffusion ». In : *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. WSDM '14. ACM, 2014, p. 393–402 (cf. p. 74).

[Bre+98] John S. BREESE, David HECKERMAN et Carl KADIE. « Empirical Analysis of Predictive Algorithms for Collaborative Filtering ». In : *Conference on Uncertainty in Artificial Intelligence*. 1998, p. 43–52 (cf. p. 17).

[BS97] Marko BALABANOVIĆ et Yoav SHOHAM. « Fab : content-based, collaborative recommendation ». In : *Communications of the ACM* 40 (1997), p. 66–72 (cf. p. 20, 21).

[Bur07] Robin BURKE. « Hybrid web recommender systems ». In : *The adaptive web*. Springer, 2007, p. 377–408 (cf. p. 18).

[BYRN99] Ricardo A. BAEZA-YATES et Berthier RIBEIRO-NETO. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999 (cf. p. 20, 35).

[Che+12] Shuo CHEN, Josh L. MOORE, Douglas TURNBULL et Thorsten JOACHIMS. « Playlist prediction via metric embedding ». In : *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '12. ACM, 2012, p. 714–722 (cf. p. 38, 42–44, 46, 47, 74).

[Cos+03] Dan COSLEY, Shyong K LAM, Istvan ALBERT, Joseph A KONSTAN et John RIEDL. « Is seeing believing ? : how recommender system interfaces affect users' opinions ». In : *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2003, p. 585–592 (cf. p. 32).

[Cre+10] Paolo CREMONESI, Yehuda KOREN et Roberto TURRIN. « Performance of Recommender Algorithms on Top-n Recommendation Tasks ». In : *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. Barcelona, Spain : ACM, 2010, p. 39–46 (cf. p. 34).

[Cre+11] Paolo CREMONESI, Franca GARZOTTO, Sara NEGRO, Alessandro Vittorio PAPADOPOULOS et Roberto TURRIN. « Looking for "good" recommendations : A comparative evaluation of recommender systems ». In : *IFIP Conference on Human-Computer Interaction*. Springer Berlin Heidelberg. 2011, p. 152–168 (cf. p. 32).

[Del+07] Chrysanthos DELLAROCAS, Xiaoquan (Michael) ZHANG et Neveen F. AWAD. « Exploring the value of online product reviews in forecasting sales : The case of motion pictures ». In : *Journal of Interactive Marketing* 21.4 (2007), p. 23–45 (cf. p. 37, 49).

[Dey01] Anind K. DEY. « Understanding and Using Context ». In : *Personal Ubiquitous Comput.* 5.1 (jan. 2001), p. 4–7 (cf. p. 25).

[DI09] Hal DAUMÉ III. « Frustratingly Easy Domain Adaptation ». In : *CoRR* abs/0907.1815 (2009) (cf. p. 38, 40, 41).

[Dia+16] Charles-Emmanuel DIAS, Vincent GUIGUE et Patrick GALLINARI. « Recommandation et analyse de sentiments dans un espace latent textuel ». In : *CORIA 2016 - Conférence en Recherche d'Informations et Applications- 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, Toulouse, France, March 9-11, 2016*. 2016, p. 73–88 (cf. p. 31).

[Din+06] Yi DING, Xue LI et Maria E. ORLOWSKA. « Recency-based Collaborative Filtering ». In : *Proceedings of the 17th Australasian Database Conference - Volume 49*. ADC '06. Hobart, Australia : Australian Computer Society, Inc., 2006, p. 99–107 (cf. p. 34).

[Dou04] Paul DOURISH. « What We Talk About when We Talk About Context ». In : *Personal Ubiquitous Comput.* 8.1 (fév. 2004), p. 19–30 (cf. p. 25).

[Dud+00] Richard O. DUDA, Peter E. HART et David G. STORK. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000 (cf. p. 33).

[EY36] C. ECKART et G. YOUNG. « The approximation of one matrix by another of lower rank ». In : *Psychometrika* 1.3 (1936), p. 211–218 (cf. p. 12).

[Gan+09] Gayatree GANU, Noemie ELHADAD et Amélie MARIAN. « Beyond the Stars : Improving Rating Predictions using Review Text Content. » In : *WebDB*. 2009 (cf. p. 31).

[Gor+11] Michele GORGOGLIONE, Umberto PANNIELLO et Alexander TUZHILIN. « The Effect of Context-aware Recommendations on Customer Purchasing Behavior and Trust ». In : *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. Chicago, Illinois, USA : ACM, 2011, p. 85–92 (cf. p. 25).

[GS+13] Elie GUARDIA SEBAOUN, Abdelhalim RAFRAFI, Vincent GUIGUE et Patrick GALLINARI. « Cross-Media sentiment Classification and Appication to Box-Office Forecasting ». In : *Proceedings of the 10th International Conference in the RIAO Series*. RIAO '13. Lisbon, Portugal, 2013 (cf. p. 4, 89).

[GS+14] Elie GUÀRDIA SEBAOUN, Vincent GUIGUE et Patrick GALLINARI. « Recommandation Dynamique dans les Graphes Géographiques ». In : *MARAMI 2014 : 5ième conférence sur les modèles et l'analyse des réseaux : Approches mathématiques et informatiques*. 2014 (cf. p. 4, 73, 81, 90, 91).

[GS+16] Elie GUÀRDIA SEBAOUN, Vincent GUIGUE et Patrick GALLINARI. « Apprentissage de trajectoires temporelles pour la recommandation dans les communautés d'utilisateurs, » in : *CAp : Conférence Francophone sur l'Apprentissage Automatique*. 2016 (cf. p. 4).

[GS09] Asela GUNAWARDANA et Guy SHANI. « A Survey of Accuracy Evaluation Metrics of Recommendation Tasks ». In : *J. Mach. Learn. Res.* 10 (déc. 2009), p. 2935–2962 (cf. p. 19, 32, 33, 35).

[Guà+15] Élie GUÀRDIA-SEBAOUN, Vincent GUIGUE et Patrick GALLINARI. « Latent Trajectory Modeling : A Light and Efficient Way to Introduce Time in Recommender Systems ». In : *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*. 2015, p. 281–284 (cf. p. 4, 73, 90, 91).

[GVL12] Gene H. GOLUB et Charles F. VAN LOAN. *Matrix Computations (4th Ed.)* Baltimore, MD, USA : Johns Hopkins University Press, 2012 (cf. p. 9).

[Her+04] Jonathan L. HERLOCKER, Joseph A. KONSTAN, Loren G. TERVEEN et John T. RIEDL. « Evaluating Collaborative Filtering Recommender Systems ». In : *ACM Trans. Inf. Syst.* 22.1 (jan. 2004), p. 5–53 (cf. p. 19, 32, 35).

[Her+99] Jonathan L. HERLOCKER, Joseph A. KONSTAN, Al BORCHERS et John RIEDL. « 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. Berkeley, California, USA : ACM, 1999, p. 230–237 (cf. p. 21).

[HL04] M. HU et B. LIU. « Mining and summarizing customer reviews ». In : *ACM SIGKDD*. 2004, p. 168–177 (cf. p. 41).

[HO00] A. HYVÄRINEN et E. OJA. « Independent Component Analysis : Algorithms and Applications ». In : *Neural Netw.* 13.4-5 (mai 2000), p. 411–430 (cf. p. 8).

[Hor+12] Inbal HOREV, Ori BRYT et Ron RUBINSTEIN. *ADAPTIVE IMAGE COMPRESSION USING SPARSE DICTIONARIES*. 2012 (cf. p. 7).

[Hot33] H. HOTELLING. « Analysis of a complex of statistical variables into principal components ». In : *J. Educ. Psych.* 24 (1933) (cf. p. 7).

[Hoy02] Patrik O. HOYER. « Non-Negative Sparse Coding ». In : *IN NEURAL NETWORKS FOR SIGNAL PROCESSING XII (PROC. IEEE WORKSHOP ON NEURAL NETWORKS FOR SIGNAL PROCESSING*. 2002, p. 557–565 (cf. p. 9, 16).

[Hoy04] Patrik O. HOYER. « Non-negative Matrix Factorization with Sparseness Constraints ». In : *J. Mach. Learn. Res.* 5 (déc. 2004), p. 1457–1469 (cf. p. 9, 16).

[Hus+12] Tim HUSSEIN, Timm LINDER, Werner GAULKE et Jürgen ZIEGLER. « Hybreed : A Software Framework for Developing Context-Aware Hybrid Recommender Systems ». In : *User modeling and user adapted interaction*. 2012 (cf. p. 25).

[Jah+10] Michael JAHRER, Andreas TÖSCHER et Robert LEGENSTEIN. « Combining Predictions for Accurate Recommender Systems ». In : *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '10. Washington, DC, USA : ACM, 2010, p. 693–702 (cf. p. 30).

[Jel+11] Herbert JELINEK, Anderson ROCHA, Tiago CARVALHO, Siome GOLDENSTEIN et Jacques WAINER. « Machine Learning and Pattern Classification in Identification of Indigenous Retinal Pathology ». In : *IEEE Engineering in Medicine and Biology Society*. 2011 (cf. p. 3).

[JL07] Nitin JINDAL et Bing LIU. « Review Spam Detection ». In : *WWW*. 2007 (cf. p. 38).

[JL08] Nitin JINDAL et Bing LIU. « Opinion Spam and Analysis ». In : *ACM WSDM*. 2008 (cf. p. 42, 47).

[Joa02] Thorsten JOACHIMS. *Learning to Classify Text using Support Vector Machines*. Springer - Kluwer Academic Publishers, 2002 (cf. p. 39).

[JT12] Michael JAHRER et Andreas TÖSCHER. « Collaborative Filtering Ensemble for Ranking. » In : *KDD Cup*. Sous la dir. de Gideon DROR, Yehuda KOREN et Markus WEIMER. T. 18. JMLR Proceedings. JMLR.org, 2012, p. 153–167 (cf. p. 24).

[Kap+15] Komal KAPOOR, Vikas KUMAR, Loren TERVEEN, Joseph A. KONSTAN et Paul SCHRATER. « "I Like to Explore Sometimes" : Adapting to Dynamic User Novelty Preferences ». In : *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. Vienna, Austria : ACM, 2015, p. 19–26 (cf. p. 3, 36).

[Kar+12] Rasoul KARIMI, Alexandros NANOPOULOS et Lars SCHMIDT-THIEME. « RFID-Enhanced Museum for Interactive Experience ». In : *Multimedia for Cultural Heritage*. T. 247. Springer Berlin Heidelberg, 2012, p. 192–205 (cf. p. 73).

[Kar11] Alexandros KARATZOGLOU. « Collaborative Temporal Order Modeling ». In : *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. Chicago, Illinois, USA : ACM, 2011, p. 313–316 (cf. p. 29, 34).

[Kni+12] Bart P KNIJNENBURG, Svetlin BOSTANDJIEV, John O'DONOVAN et Alfred KOBSA. « Inspectability and control in social recommenders ». In : *Proceedings of the sixth ACM conference on Recommender systems*. ACM. 2012, p. 43–50 (cf. p. 32).

[Koe+11] Noam KOENIGSTEIN, Gideon DROR et Yehuda KOREN. « Yahoo ! Music Recommendations : Modeling Music Ratings with Temporal Dynamics and Item Taxonomy ». In : *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys '11. Chicago, Illinois, USA : ACM, 2011, p. 165–172 (cf. p. 28, 34).

[Koh+09] Ron KOHAVI, Roger LONGBOTHAM, Dan SOMMERFIELD et Randal M. HENNE. « Controlled Experiments on the Web : Survey and Practical Guide ». In : *Data Min. Knowl. Discov.* 18.1 (fév. 2009), p. 140–181 (cf. p. 18, 32, 33).

[Kor+09] Yehuda KOREN, Robert BELL et Chris VOLINSKY. « Matrix Factorization Techniques for Recommender Systems ». In : *Computer* 42.8 (août 2009), p. 30–37 (cf. p. 12, 18, 22, 84).

[Kor08] Yehuda KOREN. « Factorization Meets the Neighborhood : A Multifaceted Collaborative Filtering Model ». In : *ACM SIGKDD*. 2008, p. 426–434 (cf. p. 8, 18).

[Kor09a] Yehuda KOREN. « Collaborative Filtering with Temporal Dynamics ». In : *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France : ACM, 2009, p. 447–456 (cf. p. 27, 28).

[Kor09b] Yehuda KOREN. « Collaborative Filtering with Temporal Dynamics ». In : *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France : ACM, 2009, p. 447–456 (cf. p. 84, 85).

[KR12] Joseph A. KONSTAN et John RIEDL. « Recommender systems : from algorithms to user experience ». In : *User Modeling and User-Adapted Interaction* 22.1 (2012), p. 101–123 (cf. p. 32, 35).

[KR90] Leonard KAUFMAN et Peter J. ROUSSEEUW. *Finding groups in data : an introduction to cluster analysis*. Wiley series in probability and mathematical statistics. A Wiley-Interscience publication. New York : Wiley, 1990 (cf. p. 7).

[Lat+09] Neal LATHIA, Stephen HAILES et Licia CAPRA. « Temporal Collaborative Filtering with Adaptive Neighbourhoods ». In : *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '09. Boston, MA, USA : ACM, 2009, p. 796–797 (cf. p. 34).

[Lee+07] Honglak LEE, Alexis BATTLE, Rajat RAINA et Andrew Y. NG. « Efficient sparse coding algorithms ». In : *In NIPS*. NIPS, 2007, p. 801–808 (cf. p. 15).

[Lin+03] Charles X. LING, Jin HUANG et Harry ZHANG. « AUC : A Better Measure than Accuracy in Comparing Learning Algorithms ». In : *Advances in Artificial Intelligence : 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003, Halifax, Canada, June 11–13, 2003, Proceedings*. Sous la dir. d'Yang XIANG et Brahim CHAIB-DRAA. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003, p. 329–341 (cf. p. 35).

[Lin07] Chih-Jen LIN. « Projected Gradient Methods for Nonnegative Matrix Factorization ». In : *Neural Comput.* 19.10 (oct. 2007), p. 2756–2779 (cf. p. 14).

[Liu10] Bing LIU. « Sentiment Analysis and Subjectivity ». In : *Handbook of Natural Language Processing, Second Edition*. Sous la dir. de Nitin INDURKHYA et Fred J. DAMERAU. ISBN 978-1420085921. Boca Raton, FL : CRC Press, Taylor et Francis Group, 2010 (cf. p. 37).

[LS01] Daniel D. LEE et H. Sebastian SEUNG. « Algorithms for Non-negative Matrix Factorization ». In : *Advances in Neural Information Processing Systems 13*. Sous la dir. de T. K. LEEN, T. G. DIETTERICH et V. TRESP. MIT Press, 2001, p. 556–562 (cf. p. 13, 14).

[LS99] Daniel D. LEE et H. Sebastian SEUNG. « Learning the parts of objects by nonnegative matrix factorization ». In : *Nature* 401 (1999), p. 788–791 (cf. p. 14).

[Maa+11] Andrew L. MAAS, Raymond E. DALY, Peter T. PHAM et al. « Learning Word Vectors for Sentiment Analysis ». In : *Association for Computational Linguistics (ACL)*. 2011 (cf. p. 38, 42, 47).

[Mcc+04] Kevin MCCARTHY, James REILLY, Lorraine MCGINTY et Barry SMYTH. *Thinking positively - explanatory feedback for conversational recommender systems*. Rapp. tech. In Proceedings of the ECCBR 2004 Workshops, 2004 (cf. p. 3).

[MH04] Matthew R. MCLAUGHLIN et Jonathan L. HERLOCKER. « A Collaborative Filtering Algorithm and Evaluation Metric That Accurately Model the User Experience ». In : *ACM SIGIR*. 2004, p. 329–336 (cf. p. 17).

[Mik+13] Tomas MIKOLOV, Kai CHEN, Greg CORRADO et Jeffrey DEAN. « Efficient Estimation of Word Representations in Vector Space ». In : *CoRR* abs/1301.3781 (2013) (cf. p. 75, 76).

[ML13a] J. J. MCAULEY et J. LESKOVEC. « From amateurs to connoisseurs : modeling the evolution of user expertise through online reviews ». In : *World Wide Web*. 2013 (cf. p. 30, 81, 83, 84).

[ML13b] Julian MCAULEY et Jure LESKOVEC. « Hidden Factors and Hidden Topics : Understanding Rating Dimensions with Review Text ». In : *Proceedings of the 7th ACM Conference on Recommender Systems*. RecSys '13. Hong Kong, China : ACM, 2013, p. 165–172 (cf. p. 31, 60).

[ML13c] Julian MCAULEY et Jure LESKOVEC. « Hidden Factors and Hidden Topics : Understanding Rating Dimensions with Review Text ». In : *ACM Conference on Recommender Systems*. 2013, p. 165–172 (cf. p. 54, 55).

[MS10] Prem MELVILLE et Vikas SINDHWANI. « Recommender Systems. » In : *Encyclopedia of Machine Learning*. Sous la dir. de Claude SAMMUT et Geoffrey I. WEBB. Springer, 2010, p. 829–838 (cf. p. 13).

[MS12] Yelena MEJOVA et Padmini SRINIVASAN. « Crossing Media Streams with Sentiment : Domain Adaptation in Blogs, Reviews and Twitter ». In : *ICWSM'12*. 2012, p. –1–1 (cf. p. 38, 39, 43).

[Oku+06] Kenta OKU, Shinsuke NAKAJIMA, Jun MIYAZAKI et Shunsuke UEMURA. « Context-Aware SVM for Context-Dependent Information Recommendation ». In : *Proceedings of the 7th International Conference on Mobile Data Management*. MDM '06. Washington, DC, USA : IEEE Computer Society, 2006, p. 109– (cf. p. 29, 33).

[PA11] Denis PARRA et Xavier AMATRIAIN. « Walk the Talk : Analyzing the Relation Between Implicit and Explicit Feedback for Preference Elicitation ». In : *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*. UMAP'11. Girona, Spain : Springer-Verlag, 2011, p. 255–268 (cf. p. 35).

[Pan+02] Bo PANG, Lillian LEE et Shivakumar VAITHYANATHAN. « Thumbs Up ? : Sentiment Classification Using Machine Learning Techniques ». In : *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. EMNLP '02. Stroudsburg, PA, USA : Association for Computational Linguistics, 2002, p. 79–86 (cf. p. 37, 39).

[Pan+09] Umberto PANNIELLO, Michele GORGOGLIONE et Cosimo PALMISANO. « Comparing Pre-filtering and Post-filtering Approach in a Collaborative Contextual Recommender System : An Application to E-Commerce ». In : *E-Commerce and Web Technologies, 10th International Conference, EC-Web 2009, Linz, Austria, September 1-4, 2009. Proceedings*. 2009, p. 348–359 (cf. p. 34).

[Par+07] Moon-Hee PARK, Jin-Hyuk HONG et Sung-Bae CHO. « Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices ». In : *Ubiquitous Intelligence and Computing : 4th International Conference, UIC 2007, Hong Kong, China, July 11-13, 2007. Proceedings*. Sous la dir. de Jadwiga INDULSKA, Jianhua MA, Laurence T. YANG, Theo UNGERER et Jiannong CAO. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007, p. 1130–1139 (cf. p. 33).

[Par08] Sun PARK. « Personalized Summarization Agent Using Non-negative Matrix Factorization ». In : *PRICAI 2008 : Trends in Artificial Intelligence : 10th Pacific Rim International Conference on Artificial Intelligence, Hanoi, Vietnam, December 15-19, 2008. Proceedings*. Sous la dir. de Tu-Bao HO et Zhi-Hua ZHOU. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008, p. 1034–1038 (cf. p. 13).

[PB97] Michael PAZZANI et Daniel BILLSUS. « Learning and Revising User Profiles : The Identification ofInteresting Web Sites ». In : *Machine Learning* (1997) (cf. p. 20).

[Pea01] K. PEARSON. « On lines and planes of closest fit to systems of points in space ». In : *Philosophical Magazine* 2.6 (1901), p. 559–572 (cf. p. 7).

[PL04] Bo PANG et Lillian LEE. « A Sentimental Education : Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts ». In : *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*. ACL '04. Barcelona, Spain : Association for Computational Linguistics, 2004 (cf. p. 47).

[Pou+14] Mickaël POUSSEVIN, Vincent GUIGUE et Patrick GALLINARI. « Extended Recommendation Framework : Generating the Text of a User Review as a Personalized Summary ». In : *CoRR* abs/1412.5448 (2014) (cf. p. 4, 36, 90).

[Pou14] Mickael POUSSEVIN. « Representation learning of user-generated data ». Thèse de doct. Université Pierre et Marie Curie - Paris VI, 2014 (cf. p. 56).

[PP10] Alexander PAK et Patrick PAROUBEK. « Twitter as a Corpus for Sentiment Analysis and Opinion Mining ». In : *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. 2010 (cf. p. 38).

[PZ11] Jiyuan PAN et Jiang-She ZHANG. « Large margin based nonnegative matrix factorization and partial least squares regression for face recognition ». In : *Pattern Recognition Letters* 32.14 (2011), p. 1822–1835 (cf. p. 8).

[Ren12] Steffen RENDLE. « Factorization Machines with libFM ». In : *ACM Trans. Intell. Syst. Technol.* 3.3 (mai 2012), 57 :1–57 :22 (cf. p. 27).

[Res+94] Paul RESNICK, Neophytos IACOVOU, Mitesh SUCHAK, Peter BERGSTROM et John RIEDL. « GroupLens : An Open Architecture for Collaborative Filtering of Netnews ». In : *ACM Conference on Computer Supported Cooperative Work*. 1994 (cf. p. 18).

[Roc71] J. J. ROCCHIO. « Relevance feedback in information retrieval ». In : *The Smart retrieval system - experiments in automatic document processing*. Sous la dir. de G. SALTON. Englewood Cliffs, NJ : Prentice-Hall, 1971 (cf. p. 20).

[Sal89] Gerard SALTON. *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989 (cf. p. 20).

[San11] Niek J. SANDERS. *Twitter Sentiment Corpus*. 2011 (cf. p. 41, 42, 47).

[Sar+01] Badrul SARWAR, George KARYPIS, Joseph KONSTAN et John RIEDL. « Item-based collaborative filtering recommendation algorithms ». In : *Proceedings of the 10th international conference on World Wide Web*. ACM. 2001, p. 285–295 (cf. p. 17).

[SG11] Guy SHANI et Asela GUNAWARDANA. « Evaluating Recommendation Systems ». In : *Recommender Systems Handbook*. Springer US, 2011, p. 257–297 (cf. p. 17, 18, 32, 33).

[Sht04] Itai SHTRIMBERG. « Good News or Bad News ? Let the Market Decide ». In : *In AAAI Spring Symposium on Exploring Attitude and Affect in Text. Palo Alto : AAAI*. Press, 2004, p. 86–88 (cf. p. 49).

[Sin+10] Sabato Marco SINISCALCHI, Jeremy REED, Torbjørn SVENDSEN et Chin-Hui LEE. « Exploiting context-dependency and acoustic resolution of universal speech attribute models in spoken language recognition ». In : *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*. 2010, p. 2718–2721 (cf. p. 34).

[Sto07] Henrik STORMER. « Improving E-Commerce Recommender Systems by the Identification of Seasonal Products ». In : *Proceedings of the 22nd Conference on Artifical Intelligence (AAAI), Workshop on Recommender Systems*. 2007 (cf. p. 34).

[Teo+16] Choon Hui TEO, Houssam NASSIF, Daniel HILL et al. « Adaptive, Personalized Diversity for Visual Discovery ». In : *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys '16. Boston, Massachusetts, USA : ACM, 2016, p. 35–38 (cf. p. 93).

[TT12] Gábor TAKÁCS et Domonkos TIKK. « Alternating Least Squares for Personalized Ranking ». In : *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys '12. Dublin, Ireland : ACM, 2012, p. 83–90 (cf. p. 82).

[Tum+10] A. TUMASJAN, T.O. SPRENGER, P.G. SANDNER et I.M. WELPE. « Predicting elections with twitter : What 140 characters reveal about political sentiment ». In : *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. 2010, p. 178–185 (cf. p. 49).

[VC11] Saúl VARGAS et Pablo CASTELLS. « Rank and relevance in novelty and diversity metrics for recommender systems ». In : *Proceedings of the 5th ACM conference on Recommender systems*. ACM. 2011, p. 109–116 (cf. p. 36).

[Wan+07] J. WANG, X. SHEN et W. PAN. « On transductive support vector machines ». In : *Joint Summer Research Conference, Machine and Statistical Learning*. T. 443. 2007, p. 7 (cf. p. 38).

[Wan+10] Hongning WANG, Yue LU et Chengxiang ZHAI. « Latent Aspect Rating Analysis on Review Text Data : A Rating Regression Approach ». In : *ACM SIGKDD*. 2010, p. 783–792 (cf. p. 42, 47).

[WB11] Chong WANG et David M. BLEI. « Collaborative Topic Modeling for Recommending Scientific Articles ». In : *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. San Diego, California, USA : ACM, 2011, p. 448–456 (cf. p. 31).

[Wen+09] Sung-Shun WENG, Binshan LIN et Wen-Tien CHEN. « Using contextual information and multidimensional approach for recommendation ». In : *Expert Systems with Applications* 36.2, Part 1 (2009), p. 1268 –1279 (cf. p. 33).

[WM15] Elizabeth WAYMAN et Sriganesh MADHVANATH. « Nudging Grocery Shoppers to Make Healthier Choices ». In : *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. Vienna, Austria : ACM, 2015, p. 289–292 (cf. p. 3).

[WZ13] Yu-Xiong WANG et Yu-Jin ZHANG. « Nonnegative Matrix Factorization : A Comprehensive Review ». In : *IEEE Transactions on Knowledge and Data Engineering* 25.6 (2013), p. 1336–1353 (cf. p. 54).

[Xio+10] Liang XIONG, Xi CHEN, Tzu-Kuo HUANG, Jeff G. SCHNEIDER et Jaime G. CARBONELL. « Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. » In : *SDM*. SIAM, 2010, p. 211–222 (cf. p. 28).

[Xu+03] Wei XU, Xin LIU et Yihong GONG. « Document Clustering Based on Non-negative Matrix Factorization ». In : *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. SIGIR '03. Toronto, Canada : ACM, 2003, p. 267–273 (cf. p. 13).

[Zaf+06] Stefanos ZAFEIRIOU, Anastasios TEFAS, Ioan BUCIU et Ioannis PITAS. « Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification ». In : *IEEE Transactions on Neural Networks* 17.3 (2006), p. 683–695 (cf. p. 13).

[Zha+14a] Chenyi ZHANG, Ke WANG, Hongkun YU, Jianling SUN et Ee-Peng LIM. « Latent Factor Transition for Dynamic Collaborative Filtering ». In : *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*. 2014, p. 452–460 (cf. p. 28, 81, 85).

[Zha+14b] Yongfeng ZHANG, Guokun LAI, Min ZHANG et al. « Explicit Factor Models for Explainable Recommendation Based on Phrase-level Sentiment Analysis ». In : *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR '14. Gold Coast, Queensland, Australia : ACM, 2014, p. 83–92 (cf. p. 31).

[Zho+10] T. ZHOU, Z. KUSCSIK, J.G. LIU et al. « Solving the apparent diversity-accuracy dilemma of recommender systems ». In : *Proceedings of the National Academy of Sciences* 107.10 (2010), p. 4511–4515 (cf. p. 36).

[Zie+05] Cai-Nicolas ZIEGLER, Sean M. MCNEE, Joseph A. KONSTAN et Georg LAUSEN. « Improving Recommendation Lists Through Topic Diversification ». In : *Proceedings of the 14th International Conference on World Wide Web*. WWW '05. Chiba, Japan : ACM, 2005, p. 22–32 (cf. p. 36).

[Zim+01] Andrew ZIMDARS, David Maxwell CHICKERING et Christopher MEEK. « Using Temporal Data for Making Recommendations ». In : *UAI '01 : Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2001, p. 580–588 (cf. p. 27).

Table des figures

Liste des tableaux

Colophon

This thesis was typeset with LaTeX 2_ε. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.