Vers des systèmes de recommandation robustes pour la navigation Web : inspiration de la modélisation statistique du langage

Geoffray Bonnin

Nancy-Université
Université Nancy 2

UFR mathématiques et informatique **École doctorale IAEM Lorraine**

Département de formation doctorale en informatique

Vers des systèmes de recommandation robustes pour la navigation Web : inspiration de la modélisation statistique du langage

THÈSE

présentée et soutenue publiquement le 23 novembre 2010

pour l'obtention du

Doctorat de l'université Nancy 2

(spécialité informatique)

par

Geoffray Bonnin

Composition du jury

Rapporteurs :	Catherine Berrut Florent Masseglia
Examinateurs :	Dietmar Jannach Jean Lieber
Directrice :	Anne Boyer
Co-encadrante :	Armelle Brun

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Loria

REMERCIEMENTS

En premier lieu, je remercie mes deux directrices de thèse, Anne Boyer et Armelle Brun, d'avoir été les meilleures encadrantes du monde. Leur passion pour la recherche, leur motivation, leur ambition et leur optimisme communicatifs m'ont permis de toujours conserver un grand engouement et une grande sérénité pendant toute la durée de cette thèse. Je m'estime très chanceux de les avoir eues pour directrices.

Je remercie Catherine Berrut, Florent Masseglia et Dietmar Jannach d'avoir accepté de faire partie de mon jury de thèse et d'être venus de loin pour assister à ma soutenance. Je remercie aussi Jean Lieber d'avoir assuré avec sérieux, implication et compétence son rôle de référent de thèse, et d'avoir accepté d'être le président de mon jury.

Je remercie ma magnifique partenaire de PACS pour son soutien malgré son propre surmenage ; en particulier, pour ses crevettes aux fruits de la passion, ses ailes de poulet à la sauce de poisson, ses massages du crâne à l'huile chauffante, pour s'être occupée de préparer mon pot de thèse, qui était absolument magnifique ; et tout simplement pour être adorable en toute circonstance, même les plus inattendues, comme par exemple quand Frank se fait tuer par les mêchants dans *Cliffhanger*.

Je remercie ma famille pour son soutien, son écoute et ses encouragements. Je remercie Ilham Esslimani d'avoir constitué ma mémoire externe pendant presque quatre années avec une fiabilité à toute épreuve, de s'être toujours montrée patiente avec moi et une amie sur laquelle j'ai toujours pu compter. Je remercie Sylvain Castagnos pour ses coups de pouces efficaces et sa sympathie. Je remercie Nazim Fatès pour son aide précieuse sur la complexité algorithmique. Je remercie Didier Schwab pour nos échanges instructifs à propos du traitement automatique du langage naturel, qui m'ont permis de solidifier la partie de cette thèse qui traite de cet aspect.

Je remercie les membres de l'équipe KIWI que je n'ai pas encore remerciés pour toutes les discussions intéressantes et enrichissantes que nous avons eues, ainsi que pour leur sympathie au quotidien : Charif Alchiekh Haydar, Sonia Ben Ticha, Cédric Bernier, Antoinette Courrier, Nicolas Jones, Alain Lelu, Samuel Nowakowski, Azim Roussanaly, Sahbi Sidhom et Manel Sorba. Je remercie mes amis du LORIA que je n'ai pas encore remerciés de m'avoir procuré joie et sérénité pendant ces trois années et dix mois, en particulier Yoann Bertrand-Pierron, Christine Bourjot, Olivier Buffet, Pierre Caserta, Rodolphe Charrier, Chu Hoang Nam, Farid Feiz, Elham Ghassemi, Thomas Girod, Arnaud Glad, Ahmad Hamad, Mohamed-Ghaith Kaabi, Mathieu Lefort, Victor Odumuyiwa, Elias Ohayon, Jean-Charles Quinton, Maxime Rio, Cédric Rose, Jamal Saboune, Olivier Simonin, Vincent Thomas et Marie Tonnelier. J'en oublie probablement, mais ils me pardonneront probablement, connaissant ma mémoire de poisson rouge.

Je remercie, enfin, les deux machines à café qui se sont succédées en bas de l'escalier de la tranche C du LORIA, sans qui tout cela n'aurait pas été possible ; ainsi que le personnel de la cantine du LORIA qui a bien voulu me resservir les deuxièmes rations de tartiflette, chili con carne, couscous, cassoulet et hachis parmentier chaque fois que je leur ai demandées, sans oublier les bonnes pâtisseries d'Isabelle.

Table des matières

Introduction

Sur quelle page Web me rendre pour trouver l'information que je recherche ? Quels sont les films que je n'ai pas encore vus et qui pourraient me plaire ? Quelle musique découvrir ? Quels articles lire pour enrichir ma connaissance de mon domaine de recherche ? Dans de nombreux domaines, le foisonnement de ressources rend difficile leur exploration et leur exploitation.

L'exemple le plus souvent mis en avant pour illustrer ce foisonnement est l'explosion du Web, impliquant des milliards de pages et de ressources. En 2005, Google indiquait encore sur la page d'accueil de son moteur de recherche le nombre de pages qu'il référençait, qui était d'environ huit milliards. Depuis 2008, il en référence plus d'un billion (mille milliards) (Alpert et Hajaj, 2008). Rien qu'au sein d'un site Web, la quantité de pages et de ressources mises à disposition des utilisateurs peut être considérable. Le site Wikipédia possède aujourd'hui plus de trois millions d'articles en anglais, et plus d'un million en français [1]. Enfin, le nombre de sites par type de service est également important : il existe des centaines de sites de météo, des centaines de sites de recettes de cuisine, des centaines de webmails, etc.

Si le Web est le domaine où cette surabondance s'observe le plus nettement, d'autres domaines renferment des quantités importantes de données. En particulier, les bases de données cinématographiques et musicales sont elles aussi gigantesques. Deux bases de données cinématographiques importantes sont proposées respectivement par le groupe de recherche GroupeLens (Resnick *et al.*, 1994) et la compagnie de location de films Netflix (Bennett et Lanning, 2007). La première contient environ 15 000 titres [2] ; la seconde contient environ 130 000 titres [3]. Une des bases de données musicales les plus conséquentes est la base de données AudioScrobbler. En seulement une journée, un échantillon de 30 000 utilisateurs du site last.fm [4], le site qui utilise cette base de donnée, écoute environ un million de chansons différentes [5].

Un autre domaine prolifique enfin, est la recherche scientifique. Le nombre total d'articles publiés dans des revues scientifiques est estimé à plus de 50 millions (Jinha, 2010), nombre qui n'inclut pas les articles parus dans des actes de conférences. La plupart de ces articles consiste en de petites innovations incrémentales, et très peu en des travaux de synthèse, dont l'écriture est plus laborieuse. En outre, plus les articles sont nombreux, plus l'écriture de leur synthèse

1. Information disponible sur la page d'accueil du site (`http://www.wikipedia.org/`).
2. Information obtenue en lançant une recherche sans mot clé sur leur interface (`http://www.movielens.org/`).
3. Information disponible sur le blog de FeedFlix (`http://feedflix.wordpress.com/`), service dédié à l'analyse statistique de l'utilisation du service Netflix.
4. `http://www.lastfm.com/`
5. Information obtenue en envoyant des requêtes à la base de données AudioScrobbler.

est difficile. Dans un tel contexte, il est difficile d'avoir une vision véritablement cohérente des avancées actuelles de la recherche.

Un des domaines de recherche principaux relatifs à la problématique de la surabondance de données est le domaine de la recherche d'information (Baeza-Yates et Ribeiro-Neto, 1999 ; Lamprier, 2008). Le principe est d'élaborer des algorithmes afin de rechercher des ressources (telles que des page Web, des films, des œuvres musicales, des articles scientifiques, etc.) en fonction de requêtes formulées par des utilisateurs via une interface prévue à cet effet. Pour cela, une indexation des données est d'abord effectuée. Cette indexation peut se faire de façon automatisée, en particulier pour des données textuelles où l'on peut par exemple indexer efficacement les documents en fonction de la fréquence des mots qu'ils contiennent. Cependant, une automatisation devient beaucoup plus difficile sur des données telles que des vidéos, de la musique, etc. Il est évidemment toujours possible d'effectuer une indexation manuelle, mais cela ne représente pas une solution appropriée face au foisonnement de données, même si des indexations collaboratives semblent devenir possibles avec l'émergence actuelle des réseaux sociaux. Quand bien même une indexation satisfaisante serait obtenue, les requêtes formulées par l'utilisateur posent également problème. En effet, il n'est pas toujours évident pour un utilisateur de savoir comment exprimer sa demande, et il peut y avoir un décalage entre sa formulation et la représentation de cette formulation dans le système (Kleinberg, 1999 ; Kwok *et al.*, 2001). Enfin, en supposant que l'utilisateur formule sa demande de façon optimale, et que l'indexation obtenue est satisfaisante, le problème de la présentation des résultats intervient. En effet, une quantité importante de ressources correspond généralement à la requête d'un utilisateur, et il est difficile de savoir quels résultats lui présenter en premier, d'autant que d'un utilisateur à l'autre, l'ordre de priorité peut changer. Au-delà des problèmes énoncés dans ce paragraphe, la recherche d'information est confrontée à nombreuses autres limitations et problématiques (Baeza-Yates et Ribeiro-Neto, 1999).

Un autre domaine de recherche relatif à la problématique de la surabondance de données est le domaine des systèmes de recommandation (Adomavicius et Tuzhilin, 2005 ; Burke, 2007 ; Lousame et Sánchez, 2009). Les systèmes de recommandation regroupent tous les systèmes capables de fournir des recommandations adaptées aux goûts, aux besoins ou aux moyens des utilisateurs, afin de les aider à accéder à des ressources utiles ou intéressantes au sein d'un espace de données important. Dans ce domaine, l'utilisateur n'a pas besoin de formuler de requête, la seule requête est implicite et peut se traduire par : « quelles sont les ressources qui correspondent à mes goûts, mes besoins ou mes moyens ? »

Les systèmes de recommandation peuvent être classés en deux types d'approches : les recommandations basées sur le contenu, et les recommandations basées sur les usages (Adomavicius et Tuzhilin, 2005). Les recommandations basées sur le contenu sont effectuées en identifiant les ressources similaires à celles appréciées par un utilisateur en fonction de leur contenu. Ces approches sont très proches de celles de la recherche d'information, les requêtes étant implicitement composées du contenu des documents déjà consultés ou consommés par l'utilisateur. Par conséquent, les mêmes limitations sont présentes. En particulier, la limitation liée au besoin d'extraction de caractéristiques associées aux ressources est présente ici aussi : l'efficacité de cette catégorie de systèmes est hautement dépendante du type de données. De plus, seules les ressources similaires aux ressources pour lesquelles une appréciation positive

a pu être obtenue peuvent être recommandées, ce qui limite fortement la diversité des recommandations (Adomavicius et Tuzhilin, 2005). Or, le compromis entre précision et diversité est déterminant du sentiment de satisfaction de l'utilisateur (Castagnos *et al.*, 2010). Une autre limitation de ces approches est qu'il faut qu'un certain nombre d'appréciations ou d'informations aient été fournis par les utilisateurs pour que les recommandations obtenues soient pertinentes (Nguyen *et al.*, 2006).

Les recommandations basées sur les usages, quant à elles, permettent de fournir des recommandations à un utilisateur sans considérer le contenu des ressources, mais en se basant sur l'analyse du comportement et/ou des appréciations des utilisateurs. Dans ce cadre, deux stratégies principales sont possibles. La première consiste à exploiter les appréciations de l'utilisateur afin de recommander les ressources qui ont été appréciées par d'autres utilisateurs ayant des goûts similaires. La seconde consiste à détecter des régularités dans le comportement des utilisateurs afin de prédire quelles ressources sont susceptibles d'intéresser un utilisateur particulier. Dans les deux cas, le contenu des ressources n'est pas considéré, et cette catégorie de systèmes de recommandation est applicable à n'importe quel type de données. Toutefois, comme pour la recommandation basée sur le contenu, le système doit avoir été manipulé un certain temps par l'utilisateur avant que les recommandations qui lui sont faites soient satisfaisantes. De plus, cette limitation s'applique aux ressources elles-mêmes : il faut un certain temps avant qu'une nouvelle ressource introduite dans le système puisse être recommandée de façon pertinente (Adomavicius et Tuzhilin, 2005). Lorsque des appréciations sont utilisées, une autre difficulté vient du faible pourcentage d'appréciations fournies explicitement par les utilisateurs relativement à l'importance des espaces de données considérés (Grčar *et al.*, 2006). De plus, pour la plupart de ces systèmes, les appréciations sont considérées comme étant statiques : il ne peut y avoir qu'une seule appréciation par utilisateur et par ressource. Or, une appréciation peut être fortement dépendante du contexte. Un tel contexte peut être géographique, météorologique, social, culturel, etc. (Adomavicius *et al.*, 2005 ; Palmisano *et al.*, 2008). Un utilisateur pourrait ainsi aimer la glace au yaourt par temps ensoleillé, mais pas par temps de neige. Un autre pourrait aimer écouter les trois sonates de Chopin, mais pas au milieu d'une soirée agitée. Le contexte est donc souvent déterminant de l'appréciation de l'utilisateur.

1 PROBLÉMATIQUE

Le contexte, dont nous venons d'évoquer l'importance, peut prendre différente formes. Une forme de contexte particulièrement déterminante est le contexte temporel. C'est une notion très vaste qui peut désigner par exemple l'emploi du temps de l'utilisateur, ou encore la périodicité dans son comportement. Au réveil, un utilisateur peut ne pas apprécier un morceau imposant de rock et lui préférer une œuvre plus propice aux réveils en douceur, comme le *Boléro* de Ravel. Le temps inclut également l'ordre chronologique dans lequel les ressources sont consultées. Par exemple, la lecture de *L'ABC de la psychologie freudienne* (Hall, 1957) peut grandement faciliter celle des œuvres de Freud. De même, il n'est pas rare qu'ayant été séduit par une œuvre d'un artiste, on s'intéresse à ses autres œuvres, et qu'on apprécie davantage ces dernières que si elles

nous avaient été soumises directement. L'ordre chronologique est donc souvent essentiel dans l'appréciation des ressources.

C'est sur cet aspect particulier que nous nous focaliserons dans cette thèse : améliorer la qualité des recommandations en prenant en compte les séquences de consultation et de consommation des ressources. Or, l'importance de l'ordre chronologique est très fortement dépendante du domaine considéré. En particulier, il est difficilement exploitable pour le cinéma. Il semble en effet délicat de détecter des suites typiques de visionnage de films de façon à ne recommander certains films qu'après le visionnage d'autres films. De plus, la datation de ces visionnages est difficilement récupérable. Par exemple, sur le site web de MovieLens, le système de recommandation du groupe de recherche GroupeLens, les appréciations attribuées aux films par les utilisateurs le sont rarement dans l'ordre dans lequel les films ont été visionnés.

En revanche, dans d'autres domaines, l'ordre chronologique est déterminant. C'est en particulier le cas de la navigation Web. En effet, les utilisateurs naviguent sur le Web de lien en lien, et dessinent donc des suites typiques de ressources identifiables et exploitables. Il est fréquent de rechercher une ressource sur un site et de devoir parcourir un certain nombre de pages avant de trouver celle que l'on recherche. Un exemple de parcours typique, pourrait être la recherche d'une information précise figurant sur le site Web d'un artiste. La connaissance de ce parcours permettrait de recommander directement la page recherchée après quelques clics, qui correspondent au début du parcours. Aussi, l'ordre chronologique dans lequel différents sites ont été consultés peut également être déterminant pour les intentions de l'utilisateur. Par exemple, un utilisateur peut visionner la webcam d'une station de ski afin de constater l'état des pistes, puis en fonction de cela consulter ou non le site de la SNCF pour acheter des billets. Il sera alors probablement intéressé par un site sur lequel le prix d'hôtels ou encore de fournitures de ski est comparé.

La prise en compte de la notion d'ordre chronologique pour la recommandation Web a beaucoup été étudiée dans la littérature. En général, lorsque l'ordre chronologique est pris en compte, soit les appréciations sont mises de côté (Nakagawa et Mobasher, 2003b ; Pitkow et Pirolli, 1999 ; Deshpande et Karypis, 2004 ; Eirinaki et Vazirgiannis, 2007), soit les auteurs font l'hypothèse que si une ressource a été consultée ou consommée, c'est qu'elle a été appréciée (Zimdars *et al.*, 2001 ; Shani *et al.*, 2005). Dans quelques rares travaux toutefois, l'ordre chronologique est combiné avec les appréciations (Trousse, 2000 ; Gündüz et Özsu, 2003).

Dans ce cadre, un certain nombre de problèmes spécifiques est alors soulevé. En particulier, des problèmes de complexité en temps et en mémoire et d'adaptabilité au comportement des utilisateurs (couverture) limitent la précision générale des systèmes utilisés. Un compromis entre ces trois critères est donc recherché (Pitkow et Pirolli, 1999 ; Deshpande et Karypis, 2004). Deux autres problématiques particulières à la navigation Web sont la résistance au bruit et la prise en compte des navigations parallèles. Le bruit correspond aux erreurs de navigation, aux retours en arrière en utilisant le bouton « précédent » du navigateur, à l'apparition intempestive de pop-ups, etc. Or, la plupart des modèles étudiés dans ce cadre exploitent soit des suites contiguës de ressources sans chercher à détecter les ressources correspondant à du bruit et ne sont donc pas robustes face à ce phénomène ; soit des suites discontiguës de ressources et impliquent un problème de complexité en temps et en mémoire, le nombre de suites possibles étant alors

très important. Les navigations parallèles apparaissent lorsque plusieurs fenêtres ou plusieurs onglets sont utilisés pour naviguer sur le Web. Or, les suites de ressources qui en résultent sont alors imbriquées en une même séquence, ce qui implique un traitement particulier, qui n'a, à notre connaissance, jamais été étudié du point de vue de la recommandation.

La problématique de cette thèse est donc d'améliorer la modélisation de la navigation Web en prenant en compte l'ordre chronologique dans les actions des utilisateurs, afin de fournir une meilleure qualité dans les recommandations, tout en utilisant des algorithmes de faible complexité, et maximisant la couverture. Les verrous auxquels nous nous intéressons pour répondre à cette problématique sont l'amélioration de la résistance au bruit contenu dans les navigations et la prise en compte des navigations parallèles.

2 PRINCIPALES CONTRIBUTIONS

Pour répondre à cette problématique, nous nous inspirons des Modèles Statistiques de Langage (MSL). En effet, les caractéristiques générales de la navigation Web sont très semblables aux caractéristiques du langage naturel (Boyer et Brun, 2007). Les premières tentatives de modélisation statistique du langage remontent aux débuts de l'informatique, alors que la recommandation n'est étudiée que depuis une vingtaine d'années. De plus, la plupart des modèles de langage proposés prennent en compte l'ordre chronologique des mots, de la même manière que l'on peut prendre en compte l'ordre chronologique de consultation des ressources. Partant de ce constat, nous avons effectué une étude des modèles de langage, et nous nous sommes inspirés des approches que nous avons jugées les plus intéressantes afin d'élaborer une modélisation de la navigation Web qui réponde au mieux à la problématique soulevée.

Nouveau modèle utilisant des n-grammes non contigus pour augmenter la qualité des recommandations

Notre première contribution est le Skipping-Based Recommender (SBR), un modèle séquentiel basé sur le concept de *skipping* [6]. Le skipping (Chen et Goodman, 1999) consiste à considérer des n-grammes non contigus dans une fenêtre de taille fixe. Par exemple, pour la suite de ressources (a, b, x, y, z, c, d) et $n = 3$, plutôt que de ne considérer que les triplets contigus comme (a, b, x) ou (y, z, c), le skipping permet de considérer aussi des triplets comme (a, x, d), (a, b, c) ou (b, c, d). Il permet donc de prendre en compte de l'information distante, tout en restreignant la taille des séquences manipulées à n ressources afin de maximiser leur représentativité tout en limitant la complexité.

Le skipping peut être effectué selon différentes *variantes*. Par exemple, pour $n = 3$ et en reprenant la suite de ressources précédente : (a, b, x, y, z, c, d), une variante utilisée dans (Shani *et al.*, 2005) permet de considérer des triplets comme (a, b, y) ou (a, b, c), mais pas (a, x, c). Nous

6. Nous conserverons dans cette thèse l'expression en anglais car il est assez difficile de trouver une traduction en bon français du terme « skipping ». En effet, des traductions telles que « sautage » ou « passage », ne nous semblent pas suffisamment précises.

proposons une nouvelle variante, appelée *skipping multinavigationnel*, fournissant des résultats comparables à la variante de skipping la plus performante de l'état de l'art (Nakagawa et Mobasher, 2003b) mais de plus faible complexité en temps et en mémoire. Ce travail a été publié dans (Bonnin *et al.*, 2008a).

Nous avons montré expérimentalement que le SBR utilisé avec la nouvelle variante de skipping fournit des recommandations dont la qualité est soit meilleure que celle des modèles de l'état de l'art, soit comparable mais avec une complexité en temps et en mémoire très inférieure, et possède soit une meilleure résistance au bruit que les modèles de l'état de l'art, soit une résistance comparable mais avec une complexité en temps et en mémoire très inférieure. Ce travail a été publié dans (Bonnin *et al.*, 2010a).

Apprentissage automatique de coefficients pour mettre en valeur les actions informatives de l'utilisateur

Utiliser une fenêtre de taille fixe permet de rendre plus léger le modèle de n-grammes avec skipping ; cependant les ressources sont soudainement ignorées dès qu'elles se retrouvent en dehors de cette fenêtre, ce que nous jugeons trop brutal. De plus, les ressources les plus éloignées dans le temps sont moins susceptibles d'être informatives des intentions de l'utilisateur que les dernières ressources qu'il a consultées. Il semble donc opportun de pondérer les occurrences des n-grammes en fonction de la distance entre les éléments qui les composent.

Nous avons dans un premier travail proposé d'utiliser un nouveau schéma de pondération. Ce schéma de pondération est une décroissance exponentielle qui prend en compte la distance entre tous les éléments qui composent les n-grammes, et fournit de meilleures recommandations que les schémas de pondérations de l'état de l'art. Ce travail a été publié dans (Bonnin *et al.*, 2008b).

Cependant, ce schéma restant relativement arbitraire, nous avons proposé dans un second travail différentes techniques d'apprentissage automatique de pondérations optimales, à partir desquelles nous avons effectué une étude empirique sur deux corpus réels. Les résultats montrent que le schéma de pondération que nous avons proposé dans (Bonnin *et al.*, 2008b) fournit des résultats comparables aux résultats obtenus en appliquant les pondérations optimales obtenues par apprentissage automatique.

Technique de calcul incrémental pour fournir des recommandations dans un temps limité

Pour calculer les recommandations, le modèle SBR considère plusieurs séquences discontiguës de taille $n-1$ dans la session active, ce qui peut engendrer un temps de calcul important. Nous avons proposé de résoudre ce problème en incorporant ces séquences une à une dans le calcul des recommandations, en allant de plus en plus loin dans la session active.

Ainsi, des recommandations peuvent être fournies en un temps limité. L'hypothèse est qu'en procédant ainsi, plus le temps imparti est grand, plus la qualité de la recommandation est grande. Nous avons réalisé une étude empirique sur deux corpus de navigation réels. Les

résultats confirment l'hypothèse, et montrent qu'un compromis satisfaisant entre temps de calcul et qualité des recommandations pouvait être fourni. Ce travail a été publié dans (Bonnin *et al.*, 2009a).

Adaptation dynamique aux actions de l'utilisateur pour fournir des recommandations quelles que soient ces actions

Le modèle SBR n'est pas en mesure de fournir des recommandations quelles que soient les actions de l'utilisateur, même si sa couverture est proche du 100%. Ce problème est un problème classique dans l'utilisation du modèle de n-grammes. Un moyen de remédier à cela est l'utilisation des all-k^{th}-order Markov models [7] (Pitkow et Pirolli, 1999) (un modèle de n-grammes est équivalent à un modèle de Markov d'ordre $n-1$). Cela consiste à fournir la recommandation du modèle de k-grammes avec une valeur k la plus élevée telle qu'une recommandation puisse être fournie. Cependant, de par la contiguïté des ressources considérées, les all-k^{th}-order Markov models, bien que fournissant une couverture totale, ne sont pas robustes au bruit.

Un nouveau modèle exploitant les caractéristiques de chacun de ces modèles a donc été proposé. Ce modèle est basé sur le principe des all-k^{th}-order Markov models, et est appelé All-k^{th}-Order Skipping-based Recommender ou AKO SBR. Le modèle obtenu est donc résistant au bruit tout en fournissant une couverture totale. Ce travail a été publié dans (Brun *et al.*, 2009).

Nouveau modèle utilisant les alignements de séquences pour prendre en compte les navigations parallèles

Il est très commun de nos jours pour les utilisateurs du Web d'utiliser les onglets du navigateur pour effectuer des navigations parallèles. Il est donc de première importance de prendre en compte les navigations parallèles dans le cadre de la recommandation Web.

Nous avons proposé d'utiliser des algorithmes d'alignement de séquence pour extraire les sous-navigations imbriquées à l'intérieur d'une même session issue de navigations parallèles. Nous avons ensuite proposé un nouveau modèle de recommandation, appelé TABAKO pour Tabbing-Based All-k^{th}-Order Markov model. Ce modèle est basé sur le all-k^{th}-order Markov model, un modèle de l'état de l'art, et utilise l'algorithme d'extraction de sous-navigations. Ce travail a été publié dans (Bonnin *et al.*, 2010b).

Entre-temps nous avons effectué des expérimentations sur un corpus de navigation qui ont montré une amélioration significative des résultats du modèle TABAKO en comparaison au all-k^{th}-order Markov model ; cependant, le modèle TABAKO étant une première proposition largement améliorable, ses résultats restent inférieurs à ceux fournis par le modèle SBR. Un article présentant ce travail a été accepté pour publication dans (Bonnin *et al.*, 2010c).

7. Nous conserverons dans cette thèse l'expression en anglais, car sa traduction, « modèles de Markov de tous les ordres k », est trop lourde à manipuler.

3 PRÉSENTATION DU PLAN

Dans un premier chapitre, les principales approches de l'état de l'art sont présentées, selon une structure en trois sections. La première section regroupe les approches basées sur le contenu, la seconde les approches basées sur les usages et la troisième les approches hybrides. Ayant cadré le contexte de cette thèse, le deuxième chapitre s'intéresse à l'exploitation de la notion de séquentialité dans la recommandation. En particulier, une discussion sur la part de séquentialité des différents types de données liés à la recommandation est présentée. Cette discussion met en avant la navigation Web, et aboutit à une étude de l'efficacité des différentes approches de la littérature qui confirme l'importance de la prise en compte de l'ordre sur ce type de données et présente ses problématiques spécifiques. Ensuite une discussion sur les similarités entre la navigation Web et le langage naturel nous a menés à envisager l'exploitation de la modélisation statistique du langage pour répondre à ces problématiques. Le troisième chapitre présente notre modèle SBR (Skipping-Based Recommender), qui est inspiré des modèles de langage et adapté aux contraintes spécifiques de la navigation Web. Dans le quatrième chapitre, trois approches sont présentées afin de rechercher respectivement une plus grande précision, une plus grande rapidité et une plus grande adaptabilité du modèle SBR que nous proposons. Le cinquième chapitre présente notre modèle TABAKO (Tabbing-Based All-k^{th}-Order Markov model) qui permet de détecter les imbrications de navigations lorsque plusieurs fenêtres ou plusieurs onglets sont utilisés. Enfin, un chapitre de conclusion revient sur l'ensemble des apports de cette thèse et présente quelques perspectives à plus ou moins long terme.

Chapitre 1

État de l'art des systèmes de recommandation

Les systèmes de recommandation peuvent être définis de plusieurs façons, qui peuvent se rapporter à différents types de données ou approches spécifiques. La définition que nous utiliserons dans cette thèse est une définition générale de Robin Burke (Burke, 2002), que nous avons traduite :

> **Système de recommandation** *Système capable de fournir des recommandations personnalisées ou permettant de guider l'utilisateur vers des ressources intéressantes ou utiles au sein d'un espace de données important.*

En pratique, la plupart des systèmes de recommandation consistent en des applications Web qui proposent des listes de ressources à des utilisateurs. De telles ressources peuvent correspondre à différents types de données tels que des films (Miller *et al.*, 2003), de la musique (Su *et al.*, 2010), des livres (Mooney et Roy, 2000), des restaurants (Burke, 2007), des news (Das *et al.*, 2007), des blagues, (Miyahara et Pazzani, 2000), des articles scientifiques (Pavlov *et al.*, 2004), des pages Web (Pitkow et Pirolli, 1999), etc.

Il est possible de classer les systèmes de recommandation de différentes manières. La classification la plus fréquente est une classification selon deux approches : les recommandations basées sur le contenu et le filtrage collaboratif (Shahabi *et al.*, 2001 ; Adomavicius et Tuzhilin, 2005 ; Huang et Huang, 2009 ; Lousame et Sánchez, 2009). En plus de ces deux approches, (Burke, 2007) propose de considérer trois autres approches : la recommandation basée sur les données démographiques, la recommandation basée sur l'utilité et la recommandation basée sur la connaissance. Dans cette thèse nous considérons que ces trois approches peuvent être considérées comme des formes particulières de recommandation basée sur le contenu, et les présentons en tant que telles. Une autre approche de la recommandation est la recommandation basée sur la détection de motifs (Nakagawa et Mobasher, 2003b), qui n'est incluse ni dans la recommandation basée sur le contenu, ni dans le filtrage collaboratif. Le filtrage collaboratif et la recommandation basée sur la détection de motifs calculent les recommandations à partir

de l'analyse de *usages* des utilisateurs. Pour définir ce que sont les usages, nous utiliserons une définition issue du dictionnaire du TLFI [1] :

> **Usage** *Fait de se servir de quelque chose, d'appliquer un procédé, une technique, de faire agir un objet, une matière selon leur nature, leur fonction propre afin d'obtenir un effet qui permette de satisfaire un besoin.*

Plus précisément, les usages considérés dans cette thèse sont d'une part l'attribution explicite d'appréciations aux ressources par les utilisateurs, et d'autre part la consultation et la consommation de ces ressources, et englobent donc les critères exploités par le filtrage collaboratif et la détection de motifs. Ainsi, la classification que nous proposons est une classification en deux types d'approches générales : la recommandation basée sur le contenu et la recommandation basée sur les usages.

Dans ce chapitre, nous introduisons dans un premier temps quelques définitions et terminologies relatives à ces deux types d'approche. Nous présentons ensuite les approches principales de la recommandation basée sur le contenu, puis les approches principales de la recommandation basée sur les usages. Enfin, nous nous intéressons aux approches hybrides, en mettant l'accent sur les approches combinant la prise en compte de séquence avec d'autres approches.

1 DÉFINITIONS ET TERMINOLOGIE

Dans cette section, nous définissons quelques concepts relatifs aux systèmes de recommandation qui seront utilisés dans cette thèse.

- **Notions liées aux usages :**

> **Note** *Une note est une valeur numérique représentant l'appréciation d'un utilisateur pour une ressource.*

Le terme « vote » est également utilisé dans la littérature, mais n'est pas approprié, un vote exprimant une préférence plus qu'une appréciation. Une note peut être attribuée directement par un utilisateur grâce à une interface prévue à cet effet, auquel cas on parle de *notation explicite*, ou déduite du comportement de l'utilisateur grâce à diverses techniques et algorithmes, auquel cas on parle de *notation implicite* (Chan, 1999 ; Claypool *et al.*, 2001 ; Fox *et al.*, 2005 ; Castagnos, 2008 ; Esslimani *et al.*, 2009).

> **Motif** *Un motif est un sous-ensemble, une sous-séquence ou une sous-structure qui se répète au sein d'un espace de données* (Han et Kamber, 2006).

> **Séquence** *Le mot* séquence *est un anglicisme désignant le terme* suite.

1. Trésor de la Langue Française Informatisé (`http://atilf.atilf.fr/`)

Suite *Famille d'éléments indexée par l'ensemble des entiers naturels.*

Dans cette thèse, nous utiliserons aussi bien l'anglicisme « séquence » que l'expression « suite de ressources ».

Ordre *Une* relation d'ordre *est une relation réflexive, transitive et antisymétrique qui permet de comparer deux à deux les divers éléments d'un ensemble. On appelle* ensemble ordonné *ou* ordre *un ensemble muni d'une relation d'ordre.*

Le terme « ordre » peut avoir, dans un contexte moins technique, un sens de succession d'évènements dans le temps, comme par exemple dans l'expression « ordre dans lequel des ressources ont été consultées ». Cependant, pour éviter toute confusion avec les autres types de relations d'ordre utilisées dans cette thèse, comme par exemple les relations d'ordre des modèles de Markov, on parlera dans ce contexte d'ordre chronologique ou temporel.

Transaction *Nous appelons* transaction *un ensemble de ressources consommées au cours d'une même opération commerciale.*

Un exemple de transaction peut être un ensemble d'articles de supermarché achetés ensemble, ou simplement un film acheté seul sur un site de vente en ligne.

Session *Nous appelons* session *la suite de ressources consultées entre la connexion et la déconnexion à un service d'un utilisateur.*

Une session est typiquement associée à la navigation Web. Cependant le terme peut être utilisé pour des suites d'actions sortant de ce cadre, comme par exemple une session de notation de films.

■ Prédiction de note et prédiction de ressource :

On rencontre parfois dans la littérature une séparation entre prédiction et recommandation (Sarwar *et al.*, 2001 ; Lousame et Sánchez, 2009). Dans ce cadre, la prédiction consiste à calculer la note la plus probable ou l'espérance de la note qu'un utilisateur va attribuer à une ressource. La recommandation quant à elle consiste à calculer une liste de ressources que l'utilisateur aimera le plus.

La prédiction de note est facilement exploitable pour effectuer de la recommandation : une fois que l'on a déterminé les notes de toutes les ressources qu'un utilisateur donné n'a pas encore consultées, il suffit de lui recommander les ressources ayant les meilleures notes prédites. Cependant, les notes ne sont pas le seul élément utilisable pour effectuer de la recommandation. Cette séparation met donc sur le même plan deux tâches d'un niveau de spécificité différent, et peut paraître inappropriée.

Le terme « prédiction » est également utilisé dans le cadre de la recommandation basée sur la détection de motifs, où il a un autre sens. Dans ce cadre, la prédiction consiste à prédire la

ou les prochaines ressources qu'un utilisateur donné est susceptible de consulter ou de consommer. Il est là aussi possible d'effectuer de la recommandation à partir de ces prédictions, simplement en recommandant les ressources prédites ayant la plus grande probabilité. Cela constitue une caractéristique intéressante, car une des limitations liées aux systèmes de recommandation utilisant des notes est que leur efficacité est fortement dépendante du nombre de notes attribuées aux ressources. Cependant, il est possible qu'une ou plusieurs des ressources qu'un utilisateur est susceptible de consulter ou de consommer soient également susceptibles de ne pas être appréciées. Par exemple, un utilisateur qui a apprécié le film *Matrix* verra probablement *Matrix 2* et *Matrix 3*, et ne les appréciera pas forcément. Face à cette problématique, certains travaux font l'hypothèse que si un utilisateur a consommé ou consulté une ressource, c'est qu'il l'a appréciée (Breese *et al.*, 1998 ; Zimdars *et al.*, 2001 ; Shani *et al.*, 2005), ce qui peut être légitime dans certains domaines. D'autres rares travaux considèrent à la fois les motifs et les notes (Trousse, 2000 ; Gündüz et Özsu, 2003 ; Sandvig *et al.*, 2007).

Le premier type de prédiction correspond donc à des prédictions de notes, quand le second correspond à des prédictions de ressources. Tous deux peuvent être utilisés pour la recommandation, et peuvent également être combinés.

■ Personnalisation :

Une notion proche de la recommandation est la personnalisation. C'est une notion souvent vague dans la littérature, et qui peut se rapporter à des concepts différents. La définition donnée par l'académie française est la suivante :

> **Personnalisation** *Action d'adapter quelque chose à une personne selon ses goûts, ses besoins ou ses moyens.*

Dans le domaine de la recherche d'information, la personnalisation correspond au tri des ressources correspondant à la requête de l'utilisateur en fonction des goûts ou des besoins de l'utilisateur (Lamprier, 2008). Dans le domaine des systèmes de recommandations, la personnalisation représente la part d'adaptation des recommandations aux goûts et aux besoins de l'utilisateur, une recommandation pouvant être plus ou moins personnalisée. En particulier, les recommandations basées sur la détection de motifs ne permettent pas de fournir des recommandations aussi personnalisées que d'autres approches de recommandation telles que le filtrage collaboratif. Aussi, la personnalisation peut prendre un sens plus général dans lequel le concept de système de recommandation est inclus, au même titre que d'autres concepts tels que l'adaptation du contenu ou de la structure d'un site Web au comportement des utilisateurs. Enfin, une recommandation peut sortir du cadre de la personnalisation. En particulier, un système de recommandation, contrairement à un système de personnalisation, peut fournir à l'utilisateur une recommandation qui sort du cadre de ses préoccupations et centre d'intérêts habituels. Par exemple, un système de recommandation peut recommander de la musique baroque à un habitué de la musique rock, pour tenter d'élargir le champ de ses recommandations. Un système de recommandation peut même créer de nouveaux besoins chez un utilisateur, en recommandant des objets dont il n'avait pas connaissance, et dont il ne peut plus se passer une fois qu'il les a considérés.

La recommandation peut donc représenter un moyen d'effectuer de la personnalisation, et une recommandation peut être plus ou moins personnalisée.

2 RECOMMANDATION BASÉE SUR LE CONTENU

La recommandation basée sur le contenu consiste à analyser le contenu des ressources ou des descriptions de ces ressources afin de déterminer quelles ressources sont susceptibles d'être utiles ou intéressantes pour un utilisateur donné. Ce sous-domaine est fortement similaire au domaine de la recherche d'information. En effet, les mêmes techniques sont utilisées, la différence se trouvant essentiellement dans l'absence de requêtes explicites formulées par l'utilisateur. Par conséquent, beaucoup de concepts généraux de la recommandation basée sur le contenu proviennent de la recherche d'information.

La plupart des systèmes de recommandation basée sur le contenu identifient les ressources similaires aux ressources qu'un utilisateur donné a appréciées (Balabanović et Shoham, 1997 ; Zhang *et al.*, 2002 ; Adomavicius et Tuzhilin, 2005 ; Pazzani et Billsus, 2007). Ainsi, quand de nouvelles ressources sont introduites dans le système, elles peuvent être recommandées directement sans que cela ne nécessite un temps d'intégration comme cela est le cas dans le cadre des systèmes de recommandation basée sur les usages.

Habituellement, la recommandation basée sur le contenu est séparée des autres formes de recommandation que nous présentons dans cette section : recommandation à partir de cas, recommandation basée sur la démographie, sur l'utilité et sur la connaissance. Comme mentionné précédemment, nous choisissons de considérer que ces dernières approches sont des formes particulières de recommandation basée sur le contenu. Dans cette section, nous nous intéressons donc dans un premier temps à la vision globale de la recommandation basée sur le contenu et aux approches habituellement présentées comme telles. Nous nous focalisons ensuite sur ce que nous considérons comme des formes particulières de recommandation basées sur le contenu.

2.1 Approche générale

Pour recommander des ressources en se basant sur le contenu, deux éléments doivent être constitués : les profils de ressource et les profils d'utilisateur. La notion de contenu ne se rapporte donc pas uniquement au contenu des ressources, mais également aux attributs descriptifs des utilisateurs.

2.1.1 Profils de ressource

Les profils de ressource consistent en un ensemble d'attributs décrivant les ressources, de façon analogue à l'index utilisé dans le domaine de la recherche d'information. Comme dans le domaine de la recherche d'information, la précision de cette approche est donc hautement dépendante de la nature des ressources : elle est beaucoup plus élevée pour des ressources

textuelles que pour des ressources telles que les images, les vidéos ou les ressources audio, dont il est difficile d'extraire des attributs. En général, quand cette approche est employée pour des ressources non textuelles, des méta-données sont utilisées [2]. Par conséquent, la plupart des recherches sur la recommandation basée sur le contenu porte sur des données textuelles (Adomavicius et Tuzhilin, 2005 ; Pazzani et Billsus, 2007).

Une étape importante de cette approche est la transformation des données textuelles sans restriction, c'est-à-dire écrites en langage naturel, en une représentation structurée. Une des techniques les plus répandues pour répondre à cette problématique est le stemming (Porter, 1997). Le stemming consiste à effectuer une transformation systématique des mots relatifs à un même concept en un même terme qui les représente tous. Ensuite un poids est attribué à chacun de ces termes en fonction de leur importance dans la ressource textuelle. Une façon classique de calculer ce poids est l'utilisation de la formule *term-frequency times inverse document-frequency* ou $tf \cdot idf$ (Salton et Buckley, 1987). Une limitation de cette technique est qu'elle ne prend pas en compte le contexte des termes. Ainsi, l'application de cette technique à des textes contenant par exemple des tournures négatives ou ironiques peut aboutir à de mauvaises représentations. C'est pourquoi d'autres méthodes ont été proposées : description de longueur minimale (Lang, 1995), utilisation de stems de plusieurs mots (Jacquemin *et al.*, 1997), etc.

Une fois cette étape finalisée, le système possède soit les listes des mots les plus importants ou les plus informatifs de chaque ressource, soit un ensemble de vecteurs de termes, c'est-à-dire un ensemble de poids associé à chaque terme de chaque ressource.

2.1.2 Profils d'utilisateur

Le profil d'un utilisateur définit ses centres d'intérêt. De tels profils consistent en un ensemble d'informations qui peuvent être entrées manuellement par l'utilisateur, ou extraites automatiquement à partir du contenu des ressources qu'il a consultées.

La première possibilité est donc de demander à l'utilisateur de fournir directement ses centres d'intérêts, à l'aide de formulaires, en lui demandant d'entrer une liste de termes, etc. Si un nombre restreint d'informations est demandé, cette approche rendra le système opérationnel rapidement, mais ne pourra pas fournir de recommandations précises. À l'inverse, en demandant un grand nombre d'informations, les recommandations seront plus précises mais le système sera trop contraignant pour l'utilisateur. De plus, les centre d'intérêts des utilisateurs peuvent évoluer au cours du temps, et une telle approche impose une actualisation manuelle régulière, ce qui est également contraignant. Un dernier inconvénient enfin, est que l'utilisateur peut ne pas remplir le formulaire honnêtement, auquel cas, les recommandations qui lui seront fournies ne pourront pas être pertinentes.

La seconde possibilité, l'extraction automatique à partir du contenu des ressources consultées par l'utilisateur, est donc souvent préférable. Une des méthodes les plus simples est de représenter les centres d'intérêt des utilisateurs par des vecteurs de termes à partir des vecteurs de termes représentant les ressources que l'utilisateur a appréciées. Les appréciations peuvent

2. Une exception se trouve dans les données musicales, où certains éléments tels que les mélodies ou le tempo peuvent être exploités avec une certaine efficacité (Yoshii *et al.*, 2006).

être obtenues de façon explicite en demandant directement aux utilisateurs de les fournir, ou implicitement en utilisant des algorithmes basés sur les usages (*cf.* section 1 de ce chapitre). Pour calculer les recommandations, il suffit alors de calculer la similarité entre les profils de ressource et les profils d'utilisateur. Cela peut être effectué selon diverses méthodes, comme la mesure de similarité cosinus. C'est dans ce cadre que cette approche est la plus similaire aux approches de la recherche d'information.

Beaucoup d'autres méthodes d'extraction automatique de profils qui se démarquent davantage de la recherche d'information ont été proposées. Dans ce cadre, les recommandations sont calculées selon la probabilité qu'un utilisateur donné appréciera une ressource. Cela peut être considéré comme un problème de classification où chaque classe représente un niveau d'appréciation (*e.g.* « aime » et « n'aime pas »). Trois des algorithmes de classification célèbres souvent utilisés dans ce contexte sont présentés dans cette section : les arbres de décision, le classificateur naïf de Bayes et les réseaux de neurones.

a) Arbres de décision

Un arbre de décision est obtenu en séparant de façon récursive les ressources en sous-groupes homogènes relativement à des variables déterminées au préalable. Dans le cas de la recommandation de ressources textuelles, ces variables sont en principe des variables booléennes sur la présence ou l'absence de termes. Ensuite, pour chaque sous-groupe, la probabilité que l'utilisateur appréciera une ressource de ce sous-groupe est conservée.

Le problème principal de l'application de cette approche à la recommandation basée sur le contenu est que la précision obtenue est dépendante du nombre de variables manipulées. Cette approche est simple et performante dans le cadre de recommandations portant sur des ressources ayant un nombre d'attributs limité, mais n'est pas appropriée dès que ces attributs sont en nombre élevé, ce qui est le cas des ressources textuelles sans restriction.

b) Classificateur naïf de Bayes

Le principe du classificateur naïf de Bayes est de déterminer la classe C pour laquelle la probabilité $P(C|\theta_1, ..., \theta_k)$ qu'une ressource r ayant pour attributs $(\theta_1, ..., \theta_k)$ appartienne à cette classe C soit maximale. Les attributs sont supposés indépendants, et maximiser $P(C|\theta_1, ..., \theta_k)$ revient à maximiser la formule suivante :

$$P(C) \prod_{i=1}^{k} P(\theta_i|C) \tag{1.1}$$

Les valeurs de $P(C)$ et de $P(\theta_i|C)$ sont estimées à partir d'un corpus d'apprentissage. Pour chaque ressource r, chaque valeur de la formule (1.1) est estimée pour chaque classe (ici chaque niveau d'appréciation). r est alors placée dans la classe pour laquelle cette valeur est la plus élevée.

En dépit du fait que les attributs des ressources sont en réalité inter-dépendants, le classificateur naïf de Bayes s'avère fournir une grande précision et représente un algorithme simple et ayant un temps de calcul réduit. De plus, contrairement aux arbres de décision, il est applicable aussi bien sur des données ayant un nombre d'attributs limité que sur des données sans restriction.

c) Réseaux de neurones

Dans un réseau de neurones, un neurone est simplement une fonction non linéaire, de variables réelles et bornée. Cette fonction est généralement définie comme suit :

$$f(x_1, ..., x_k; w_1, ..., w_k) = \left\lfloor \sum_{i=1}^{k} w_i x_i \right\rfloor$$

où les variables $w_1, ..., w_k$ correspondent à des poids à associer aux variables $x_1, ..., x_k$, qui sont déterminés à partir d'un corpus d'apprentissage. La fonction tangente hyperbolique est une fonction sigmoïde qui a certaines propriétés particulièrement appropriées pour l'apprentissage de réseaux de neurones (Kalman et Kwasny, 1992). De tels neurones sont associés en réseau selon deux types d'architecture : les réseaux bouclés qui correspondent à des graphes orientés avec circuit et les réseaux non-bouclés qui correspondent à des graphes orientés sans circuit.

Dans le cadre de la recommandation basée sur le contenu, les variables $x_1, ..., x_k$ correspondent à la fréquence des termes utilisés pour caractériser les ressources (qui peut être normalisée par rapport à la longueur du texte). L'architecture la plus fréquemment adoptée est l'architecture en réseaux non bouclés avec une structure de perceptron multicouche (Hornik, 1993). Plus précisément, cette structure consiste en général en k entrées (les k attributs d'une ressource), une couche d'un certain nombre de neurones cachés, et un certain nombre de neurones de sortie. Chaque neurone de sortie indique un score permettant de déterminer si une ressource appartient à la classe du niveau d'appréciation à laquelle il est associé. Un algorithme répandu pour effectuer l'apprentissage des poids est l'algorithme PLA (Perceptron Learning Algorithm) (Rosenblatt, 1958). Il consiste à initialiser les variables de façon aléatoire et à les ajuster itérativement de façon à minimiser le nombre de ressources disposées dans de mauvaises classes.

En plus de permettre un apprentissage rapide, l'utilisation de réseaux de neurones a l'avantage de permettre un ajustement particulièrement fin grâce à l'utilisation de la fonction sigmoïde. Selon le domaine d'application il peut s'avérer plus ou moins efficace que ses alternatives (Pazzani et Billsus, 1997).

2.2 Formes particulières de recommandation basée sur le contenu

Nous présentons à présent quatre approches habituellement présentées comme sortant du cadre de la recommandation basée sur le contenu, mais que nous considérons comme des points de vue particuliers de la recommandation basée sur le contenu. En outre, une approche de recommandation basée sur contenu peut relever de plusieurs de ces quatre approches.

2.2.1 Recommandation à partir de cas

La recommandation à partir de cas (Smyth, 2007) est basée sur le raisonnement à partir de cas (Althoff, 2001) :

> **Raisonnement à partir de cas** *Le raisonnement à partir de cas consiste à adapter des solutions concrètes à des problèmes spécifiques rencontrés dans le passé, appelés cas, pour résoudre des problèmes similaires.*

Pour effectuer des recommandations en utilisant le raisonnement à partir de cas, il suffit donc de considérer les ressources comme des cas et les recommandations comme des solutions à ces problèmes. Les deux critères qui différencient la recommandation à partir de cas des autres formes de recommandation basée sur le contenu, sont la façon dont les ressources sont représentées et la façon dont le concept de similarité est appréhendé.

Les cas consistent en un ensemble d'attributs décrivant les ressources. Ils sont donc *a priori* similaires aux profils de ressource présentés dans la section 2.1.1 de ce chapitre. La différence est que dans le domaine de la recommandation basée sur le contenu, les attributs considérés consistent généralement en des termes extraits de données textuelles, alors que les cas contiennent généralement des attributs qui sortent de ce cadre (par exemple le prix d'un livre dans le cadre d'une vente en ligne).

Dans le cadre de la recommandation basée sur le contenu, la similarité entre deux ressources est généralement calculée en fonction du nombre de termes qu'elles ont en commun. Dans le cadre de la recommandation à partir de cas, la similarité entre deux cas c_1 et c_2 est généralement calculée selon une somme pondérée des similarités des attributs correspondants de c_1 et c_2 (Smyth, 2007). L'avantage est donc que des attributs de types différents peuvent être considérés, et que pour chaque attribut, il est possible de déterminer une mesure de similarité spécifique.

2.2.2 Recommandation basée sur les données démographiques

La recommandation basée sur les données démographiques consiste à répartir les utilisateurs en plusieurs classes en fonction d'informations démographiques leur étant associées, telles que le sexe, l'âge, la profession, la localisation, etc. L'hypothèse sur laquelle repose cette approche est que deux utilisateurs ayant évolué dans un environnement similaire ont des codes esthétiques communs et sont donc plus susceptibles d'avoir des goûts communs que deux utilisateurs ayant évolué dans des environnements différents et ne partageant donc pas les mêmes codes [3]. Un des avantages principaux de cette technique est qu'elle est applicable dès que les informations nécessaires sont obtenues, et permet de fournir des recommandations relativement satisfaisantes dès qu'un utilisateur commence à utiliser le système (Nguyen *et al.*, 2006).

3. Cette hypothèse rejoint en un sens le concept de rationalité esthétique de (Rochlitz, 1992). En effet, selon Rochlitz, s'il n'existe pas de critères absolus et universels de jugement esthétique, des « critères objectifs [qui prétendent] à une validité intersubjective » peuvent apparaître « par accident ou par conformisme ». Or, deux utilisateurs fortement similaires selon des critères démographiques sont susceptibles de partager de tels critères objectifs qui auraient émergé dans leur environnement commun.

Cependant, pour des raisons de respect de la vie privée, il n'est pas toujours possible d'obtenir de telles informations.

2.2.3 Recommandation basée sur l'utilité

La recommandation basée sur l'utilité, parfois appelée recommandation basée sur les préférences, consiste à calculer les recommandations selon une fonction d'utilité pour l'utilisateur (Stolze et R., 2001). Toute la problématique est donc de définir une telle fonction d'utilité. Une façon de procéder est de demander aux utilisateurs de remplir des formulaires [4]. Par exemple, dans le cadre de vente en ligne de micro-ordinateurs, il est possible de demander des renseignements sur l'usage qu'en fera le client. Un certain nombre de méthodes alternatives sont décrites dans (Huang, 2008).

2.2.4 Recommandation basée sur la connaissance

La recommandation basée sur la connaissance consiste à accumuler des informations relativement élaborées sur un utilisateur pour pouvoir ensuite lui recommander des ressources (Towle et Quinn, 2000). Une analogie avec la vie réelle serait par exemple une recommandation faite par un ami qui nous connaît bien et se serait basé sur des informations précises nous concernant, plutôt que sur nos préférences. Cette approche permet également d'expliciter des liens entre les ressources : par exemple que la cuisine chinoise est proche de la cuisine vietnamienne. Une forme de recommandation basée sur la connaissance est la recommandation à partir de cas avec des attributs se rapportant à ce genre d'informations (Schmitt et Bergmann, 1999).

2.3 Limitations de la recommandation basée sur le contenu

La principale limitation de la recommandation basée sur le contenu est qu'elle nécessite l'acquisition d'un nombre suffisant d'attributs décrivant les ressources. C'est pourquoi elle est appropriée dans le cadre de ressources textuelles ou quand des descriptions textuelles des ressources ont été entrées manuellement. Dans le cadre de ressources textuelles, une des limitations provient des méthodes de classification de texte utilisées : en effet, deux ressources peuvent être similaires du point de vue de leurs attributs, mais avoir une qualité ou une pertinence incomparable.

Une autre limitation est que ces modèles ne peuvent recommander que des ressources similaires à celles qu'un utilisateur donné a appréciées, ce qui empêche de recommander d'autres ressources que ce même utilisateur pourrait également apprécier. Pour amoindrir ce problème, il est possible de fournir des recommandations aléatoires parmi les recommandations.

Enfin, une dernière limitation est qu'un nouvel utilisateur d'un tel système doit avoir consulté ou fourni des appréciations pour un certain nombre de ressources avant que le système ne puisse lui fournir des recommandations pertinentes. Ce problème est connu sous le nom

4. C'était en particulier la technique utilisée par le système PersonaLogic, dont le site a fermé aux environs des années 2001/2002.

de démarrage à froid. Une façon de réduire ce problème est de demander un certain nombre d'informations à l'utilisateur au moment de son arrivée (en nombre limité pour ne pas rendre le système trop contraignant) et d'utiliser un profil type correspondant aux informations qu'il a fournies.

3 RECOMMANDATION BASÉE SUR LES USAGES

Les systèmes de recommandation basée sur les usages calculent les recommandations en se basant sur les usages passés que les utilisateurs ont fait du système. Cette approche ne nécessite pas de considérer le contenu des ressources, ce qui présente plusieurs avantages. Le premier avantage est que cela évite l'extraction de profils de ressource et d'utilisateur. Le terme « profil » est utilisé ici dans le même sens que celui employé pour la recommandation basée sur le contenu de la section précédente. En réalité des profils d'utilisateurs peuvent être définis dans le cadre de la recommandation basée sur les usages, mais se rapportent à une forme différente de profil. Le second avantage est que les approches basées sur les usages ne sont pas aussi dépendantes de la nature des données que les approches basées sur le contenu. En particulier, elles sont applicables aussi bien aux données graphiques que sonores, deux types de données pour lesquels les recommandations basées sur le contenu ont une efficacité très limitée.

Parmi les critères exploitables pour effectuer des recommandations basées sur les usages, les deux critères principaux sont les appréciations et les motifs. L'utilisation d'appréciations correspond au filtrage collaboratif, et celle des motifs à différentes approches issues du domaine de la fouille de données.

Dans cette section, nous présentons les approches principales exploitant ces deux critères ; puis nous présentons les limitations de ces approches.

3.1 Filtrage collaboratif

La première méthode de recommandation basée sur les usages que nous présentons, le filtrage collaboratif, exploite les appréciations des utilisateurs sur les ressources. La représentation des appréciations se fait en règle générale par des notes. Comme mentionné dans la section 1 de ce chapitre, ces notes sont soit attribuées de façon explicite par les utilisateurs, ce qui représente une forme d'usage, soit de façon implicite à partir d'autres formes d'usage des utilisateurs. Le filtrage collaboratif est une des techniques les plus explorées du domaine de la recommandation (Das *et al.*, 2007).

Le premier système de recommandation ayant été désigné comme étant un système de filtrage collaboratif est le système Tapestry (Goldberg *et al.*, 1992). En réalité, ce système était à la fois un système de recommandation et un système de recherche d'information puisque les utilisateurs pouvaient accéder à des messages électroniques à la fois en fonction des appréciations des autres utilisateurs et en formulant des requêtes. Les auteurs avaient appelé cette approche « filtrage collaboratif » car les utilisateurs pouvaient collaborer afin de mettre de côté les messages électroniques indésirables.

Cette expression a depuis beaucoup été reprise au sein de la communauté des chercheurs de ce domaine. Si elle est aujourd'hui encore communément utilisée (Das *et al.*, 2007 ; Abernethy *et al.*, 2009 ; Koren, 2009 ; Huang et Jebara, 2010), les systèmes dits de « filtrage collaboratif » d'aujourd'hui sont bien différents du système Tapestry. En effet, dans la représentation commune de la littérature actuelle, le filtrage collaboratif consiste à fournir des recommandations en exploitant exclusivement les notes attribuées par les utilisateurs pour regrouper ces derniers en fonction de leurs goûts communs. Ces systèmes de recommandation sélectionnent donc des ressources plutôt qu'ils n'en filtrent (filtrer consiste davantage à mettre de côté les ressources indésirables), et les utilisateurs soumettent des appréciations indépendamment les uns des autres et ne collaborent pas directement. Cette terminologie peut donc sembler inappropriée. Cette imprécision était déjà mise en avant en 1997 par (Resnick et Varian, 1997). Cependant, elle est à ce point répandue qu'il ne serait pas judicieux de proposer une alternative dans cette thèse, cette approche se situant au contour de ma problématique.

Plus précisément, le filtrage collaboratif utilise une matrice dont les lignes correspondent aux utilisateurs et les colonnes aux ressources. Chaque cellule de la matrice correspond alors à une note fournie par l'utilisateur correspondant pour la ressource correspondante. Le but est alors de prédire les notes que les utilisateurs attribueraient aux ressources pour lesquelles ils n'ont pas encore fourni de note, pour ensuite recommander les ressources ayant les meilleures notes prédites. Le filtrage collaboratif est en général classé en deux approches : l'approche mémoire et l'approche modèle. Dans cette section, nous présentons tout d'abord la notation que nous utiliserons ; puis nous présentons les deux approches mémoire et modèle du filtrage collaboratif.

Notation

Soit $R = \{r_1, r_2, ..., r_N\}$ l'ensemble des ressources, $U = \{u_1, ..., u_M\}$ l'ensemble des utilisateurs. Nous désignons par u_a l'utilisateur actif pour lequel une recommandation doit être calculée. Chaque note de u_i pour chaque ressource r_j est une valeur numérique désignée par α_{ij}. La matrice de notes est alors la matrice suivante :

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{M1} & \alpha_{M2} & \cdots & \alpha_{MN} \end{pmatrix}$$

3.1.1 Approche mémoire

L'approche mémoire consiste à utiliser des algorithmes qui calculent chaque prédiction de note à partir de toute la matrice de notes. La méthode la plus couramment utilisée dans la littérature consiste à effectuer une somme pondérée de ces notes. Généralement, la formule

utilisée pour le calcul de la prédiction de la note α^*_{aj} d'une ressource r_j pour l'utilisateur actif u_a est la suivante :

$$\alpha^*_{aj} = \overline{\alpha_a} + \kappa \sum_{i=1}^{M} w(u_a, u_i) \cdot (\alpha_{ij} - \overline{\alpha_i}) \tag{1.2}$$

où $\overline{\alpha_i}$ désigne la moyenne des notes attribuées par l'utilisateur u_i, $w(u_a, u_i)$ est une fonction de pondération indiquant la similarité entre u_a et u_i, et κ est un coefficient de normalisation tel que la somme des valeurs absolues des poids soit égale à 1. L'utilisation des $\overline{\alpha_i}$ permet de recentrer les notes attribuées par les utilisateurs. En effet, il se peut que deux utilisateurs aient des goûts similaires, mais que l'un soit plus sévère que l'autre dans ses notations. Repositionner chaque note de chaque utilisateur relativement à la moyenne de ses notes permet de limiter les effets de ce phénomène.

L'élément déterminant de cette formule est la fonction de pondération $w(u_a, u_i)$, qui permet d'accorder plus d'importance aux notes des utilisateurs les plus similaires à l'utilisateur actif. Les valeurs de cette fonction sont en général calculées hors-ligne, et peuvent être obtenues de différentes façons. Parmi les fonctions de pondération les plus communes, on peut citer la similarité cosinus entre les vecteurs de notes (Breese *et al.*, 1998) :

$$w(u_a, u_i) = \cos(\overrightarrow{u_a}, \overrightarrow{u_i}) = \frac{\overrightarrow{u_a} \cdot \overrightarrow{u_i}}{\|\overrightarrow{u_a}\| \, \|\overrightarrow{u_i}\|} = \frac{\sum_j \alpha_{aj} \cdot \alpha_{ij}}{\sqrt{\sum_j \alpha^2_{aj}} \sqrt{\sum_j \alpha^2_{ij}}}$$

où les sommes sur j correspondent à la somme sur toutes les ressources notées à la fois par u_a et par u_i. C'est donc la même mesure de similarité que celle utilisée dans le cadre de la recommandation basée sur le contenu pour calculer la similarité entre les ressources (et entre le profil d'utilisateur et les ressources). La différence est que les dimensions du vecteur correspondent ici aux ressources et ont pour valeur les notes, alors que dans le domaine de la recommandation basée sur le contenu, les dimensions correspondent aux termes des ressources. Une autre fonction de pondération célèbre est le coefficient de corrélation de Pearson (Resnick *et al.*, 1994) :

$$w(u_a, u_i) = \frac{\sum_j (\alpha_{aj} - \overline{\alpha_a})(\alpha_{ij} - \overline{\alpha_i})}{\sqrt{\sum_j (\alpha_{aj} - \overline{\alpha_a})^2} \sqrt{\sum_j (\alpha_{ij} - \overline{\alpha_i})^2}}$$

La différence avec la mesure de la similarité cosinus est que les vecteurs $\overrightarrow{u_a}$ et $\overrightarrow{u_i}$ des utilisateurs sont centrés relativement à leurs notes moyennes, comme dans la formule 1.2. Le résultat est donc une valeur de corrélation comprise entre -1 et 1. Si $w(u_a, u_i)$ vaut 1, alors les utilisateurs sont considérés comme très similaires ; si $w(u_a, u_i)$ vaut 0, alors les utilisateurs sont indépendants ; si $w(u_a, u_i)$ vaut -1, alors les utilisateurs sont fortement opposés (Castagnos, 2008). Une façon d'utiliser cette mesure avec la formule 1.2 est d'ignorer les utilisateurs ayant une valeur de corrélation négative avec u_a.

L'approche mémoire a l'avantage d'être à la fois simple et performante, et de permettre une adaptation dynamique au fur et à mesure que de nouvelles notes sont entrées dans la

matrice ; cependant sa complexité est telle que son utilisation n'est possible que sur un espace de données relativement réduit. Cette dernière limitation est connue sous le nom de problème du passage à l'échelle.

3.1.2 Approche modèle

L'approche modèle répond à la problématique de la complexité de l'approche mémoire en utilisant des modèles d'utilisateur, de ressource, de communauté, de session, etc. L'avantage est que ces modèles peuvent être construits hors ligne à partir de corpus d'apprentissage et être utilisés rapidement en ligne pour calculer les recommandations. Par conséquent, la problématique de la complexité en temps de la construction de ces modèles peut être secondaire, selon le type d'application. Il est en effet souvent envisageable de construire un modèle en plusieurs heures voire en plusieurs jours, ce qui a l'avantage de permettre des approches de construction plus subtiles, mais l'inconvénient de rendre impossible une actualisation fréquente.

Dans le cadre de l'approche modèle, la prédiction d'une note peut être calculée de deux manières. La première manière consiste à construire un modèle probabiliste dans lequel sont stockées des estimations de probabilités. Ces probabilités peuvent alors être utilisées pour calculer l'espérance des notes ou de déterminer la note la plus probable :

1. en calculant l'espérance de la note (Breese *et al.*, 1998) :

$$p_{aj} = E(\alpha_{aj}) = \sum_{\alpha=\alpha_{\min}}^{\alpha_{\max}} \alpha \cdot p(\alpha_{aj} = \alpha)$$

   où α représente une valeur de note.

2. ou en recherchant la note la plus probable (Schafer *et al.*, 2007) :

$$p_{aj} = \arg\max_{\alpha} p(\alpha_{aj} = \alpha)$$

La problématique est alors de construire un modèle à partir duquel il sera possible d'obtenir les probabilités de ces deux équations.

Une autre possibilité généralement classée dans les approches modèle consiste à regrouper les utilisateurs ou les ressources en sous-groupes homogènes, et d'appliquer les approches mémoire sur les sous-groupes ainsi obtenus [5]. L'espace de données sur lequel sont appliqués les algorithmes peut ainsi être suffisamment réduit pour que les algorithmes de l'approche mémoire lui soient appliqués.

De nombreuses approches modèle ont été proposées, dont les principales sont présentées dans cette section.

5. Regrouper les utilisateurs (ou les ressources) en sous-groupes est une forme de modélisation en ce sens que cela engendre des modèles de communauté (ou de types de ressources).

a) Réseaux bayésiens

Un réseau bayésien, parfois appelé diagramme d'influence, consiste en un graphe orienté sans circuit $G = (X, A)$ dans lequel les nœuds $X = \{x_1, ..., x_N\}$ représentent des variables aléatoires et les arcs A les relations entre ces variables. Ces relations sont des relations de *dépendance conditionnelle* formulées sous forme de probabilités conditionnelles. Ces probabilités conditionnelles sont stockées sous forme de tableaux, chaque nœud du graphe possédant son propre tableau. La distribution jointe du réseau peut alors être formulée de la façon suivante :

$$P(x_1, ..., x_N) = \prod_{i=1}^{N} P\left(x_i | Par(x_i)\right) \tag{1.3}$$

où $Par(x_i)$ correspond à l'ensemble des valeurs des parents de x_i dans le graphe.

Dans le cadre du filtrage collaboratif, chaque nœud est associé à une ressource et chaque valeur de nœud à une note. De cette manière, chaque ressource possède un ensemble de parents dont ils sont grandement dépendants. Il est ainsi possible de stocker des probabilités de façon compacte et de les utiliser pour effectuer les prédictions de notes.

Un des travaux les plus célèbres utilisant des réseaux bayésiens pour le filtrage collaboratif est celui de (Chickering *et al.*, 1997). Dans ce travail, les auteurs proposent d'utiliser des arbres de décision (*cf.* section 2.1.2 de ce chapitre) pour représenter les dépendances conditionnelles de façon encore plus compacte. Dans (Breese *et al.*, 1998), cette dernière approche est comparée empiriquement à d'autres modèles de filtrage collaboratif. Les résultats montrent qu'elle est plus performante que l'approche mémoire utilisant la similarité cosinus et aussi performante que l'approche mémoire utilisant la corrélation de Pearson. Les réseaux bayésiens ont une complexité en temps et en mémoire plus faible pour le calcul des prédictions de note, mais une complexité en temps très élevée pour la phase d'apprentissage (plusieurs heures, voire plusieurs jours). Par conséquent, ils sont préférables quand une actualisation en temps réel n'est pas obligatoire et que l'espace de données est très important.

(Heckerman *et al.*, 2001) proposent de généraliser les réseaux bayésiens à des réseaux de dépendances, c'est-à-dire que le graphe utilisé pour modéliser les dépendances autorise les cycles. Les résultats obtenus sont légèrement inférieurs à ceux des réseaux bayésiens en termes de précision, mais l'occupation mémoire et le temps de calcul sont inférieurs. Cela est dû à ce que la cyclicité des réseaux de dépendance permet d'obtenir les probabilités directement (sans inférence, comme avec les réseaux bayésiens).

b) Classifieur naïf de Bayes

Le classificateur naïf de Bayes, dont le principe a été présenté dans la section 2.1.2 de ce chapitre, est une forme particulière, et très simple, de réseau bayésien (Friedman *et al.*, 1997). Il peut être représenté en tant que tel, et utilisé pour construire des communautés homogènes d'utilisateurs. Dans ce cadre, les attributs correspondent aux ressources, et leurs valeurs aux notes attribuées par l'utilisateur considéré. Par conséquent, au lieu de porter sur les attributs

descriptifs des ressources, l'hypothèse d'indépendance porte sur les notes attribuées par les utilisateurs, hypothèse qui n'est pas plus réaliste que la précédente. Relativement à cette considération, l'utilisation du classificateur naïf de Bayes fournit de bons résultats dans le cadre du filtrage collaboratif également, mais se révèle moins performante que les réseaux bayésiens dans leur version moins restreinte décrite ci-dessus (Breese *et al.*, 1998).

Un des problèmes liés à l'utilisation d'un tel classificateur est que le nombre de classes et leurs paramètres doivent être déterminés à l'avance. Une solution est d'utiliser l'algorithme *Expectation Maximization* (Dempster *et al.*, 1977) avec un nombre fixé de classes pour déterminer les paramètres qui fournissent le maximum de vraisemblance et la méthode de Cheeseman et Stutz (Cheeseman et Stutz, 1996) pour déterminer le nombre de classes qui fournit la meilleure vraisemblance marginale (Breese *et al.*, 1998).

Plutôt que de classer les utilisateurs en communautés homogènes, il est possible de classer les ressources en fonction de leur niveau d'appréciation. Un des premiers travaux ayant utilisé cette approche (Miyahara et Pazzani, 2002) utilise un nombre de classes fixé à deux : « aime » et « n'aime pas » en utilisant une transposition relativement à une note seuil. Cette approche a depuis été expérimentée avec un plus grand nombre de classes, en particulier dans (Su et Khoshgoftaar, 2006). Les résultats montrent qu'une telle classification des ressources permet un meilleur passage à l'échelle que les approches mémoire, mais fournit de moins bons résultats.

c) Clustering

Une alternative à la classification est le clustering, qui a l'avantage de ne pas imposer de connaître à l'avance les paramètres des classes. En effet, le clustering consiste à regrouper des ressources similaires et/ou à séparer les ressources dissimilaires (Han et Kamber, 2006), quand les classificateurs répartissent les ressources dans des classes dont le nombre et les paramètres sont prédéterminés. Nous présentons trois catégories classiques de clustering : le clustering hiérarchique, le clustering par partitionnement et le clustering basé sur la densité.

Un algorithme célèbre basé sur le clustering hiérarchique est l'algorithme BIRCH (Zhang *et al.*, 1996). Son principe est de partitionner les données de façon hiérarchique et de les disposer dans une structure d'arbre, puis d'utiliser diverses méthodes de clustering sur les feuilles de l'arbre afin de diviser les clusters trop épars et de regrouper les clusters denses en plus grands clusters. Les avantages principaux de cet algorithme sont qu'il a une complexité en temps très faible, et qu'il est applicable sur un grand espace de données, tout en fournissant une bonne qualité de clustering ; cependant, la notion de diamètre utilisée pour déterminer le contour des clusters a tendance à aboutir à des clusters sphériques, ce qui n'est pas forcément approprié selon les données considérées.

Un algorithme célèbre de clustering par partitionnement est l'algorithme k-means (MacQueen, 1967). Son principe est le suivant. Dans un premier temps, k éléments (*e.g.* utilisateurs, ressources) sont choisis aléatoirement. Ensuite, chaque élément parmi les $M - k$ restants est associé à l'élément le plus similaire (dans le cadre du filtrage collaboratif, en termes de similarité de notes) parmi les k qui ont été choisis à la première étape. Puis pour chaque cluster ainsi obtenu, le centre est ajusté en fonction des éléments qu'il contient. Le processus est alors

réitéré jusqu'à ce que les centres soient stabilisés. Le principal avantage de cet algorithme est sa simplicité et sa très faible complexité en temps ($\mathcal{O}(kMt)$ pour t itérations). Les principaux inconvénients sont que le nombre de clusters doit être prédéterminé, et que les clusters obtenus sont hautement dépendants des k centres choisis à l'initialisation. Cet algorithme est approprié pour détecter de petits groupes compacts relativement bien séparés, ce qui n'est pas forcément approprié selon les données considérées.

Un algorithme célèbre de clustering basé sur la densité est l'algorithme DBSCAN (Ester *et al.*, 1996). Son principe est le suivant. Dans un premier temps, le voisinage de chaque élément est recherché, et si un nombre suffisant d'éléments a été trouvé un nouveau cluster est créé. Les clusters ainsi obtenus sont alors agrandis en intégrant itérativement les éléments de leurs voisinage dont le nombre est suffisant pour respecter les contraintes de densité, jusqu'à ce qu'aucun élément ne puisse être ajouté à aucun cluster. Deux paramètres doivent donc être fixés à l'avance : un rayon de voisinage ε, et un nombre minimum d'éléments $MinPts$. Les avantages principaux de cet algorithme sont qu'il peut détecter des formes arbitraires et que le nombre de clusters ne doit pas être fixé à l'avance. Les principaux inconvénients sont que son efficacité est hautement dépendante des paramètres ε et $MinPts$ et que deux éléments très différents peuvent être placés dans le même cluster.

Généralement, quand le clustering est expérimenté pour le filtrage collaboratif, les résultats obtenus sont légèrement inférieurs aux résultats fournis par les approches mémoire. En effet, le clustering est utilisé pour séparer les utilisateurs en communautés homogènes sur lesquelles il sera possible d'appliquer les approches mémoire. Or, les communautés obtenues ne sont jamais parfaitement homogènes, et les utilisateurs appartenant à des communautés différentes pourraient apporter des informations pertinentes qui sont ignorées. Cependant, dans le cadre d'un espace de données important, le clustering permet un passage à l'échelle que ne permettent pas les approches mémoire.

3.2 Limitations du filtrage collaboratif

Une des principales limitations du filtrage collaboratif est le problème du manque de données. En effet, dans le cadre de notes explicites, le pourcentage moyen de ressources pour lesquelles les utilisateurs ont fourni une appréciation est très bas. Par exemple, une des bases de données fournies par MovieLens [6] consiste en 100 000 notes pour 1 642 films par 943 utilisateurs, soit 6,3% de notes fournies. Dans un tel cadre, la similarité entre deux utilisateurs ne peut être calculée que s'ils ont noté un minimum de ressources communes.

Tout comme les approches basées sur le contenu, le filtrage collaboratif souffre du problème du démarrage à froid : avant que le système puisse fournir des recommandations pertinentes à un utilisateur, il faut que ce dernier ait fourni, implicitement ou explicitement, des appréciations pour un nombre suffisant de ressources. Un problème supplémentaire par rapport aux

6. `http://www.grouplens.org/node/12`

recommandations basées sur le contenu est que cette limitation s'applique également aux nouvelles ressources introduites dans le système.

Des solutions à ces problèmes se trouvent dans les approches hybrides, présentées dans la section 4.

3.3 Détection de motifs

Une autre façon de faire de la recommandation basée sur les usages est de calculer et présenter à l'utilisateur les ressources qu'il est susceptible de consulter par la suite. Cette approche est en rapport direct avec le domaine de la fouille de données.

La fouille de données (Han et Kamber, 2006) est un domaine englobant un ensemble de techniques dont le but est de détecter des régularités au sein d'un espace important de données. C'est donc un domaine très général qui s'intéresse à de nombreux aspects relatifs au traitement de données tels que l'analyse de données, la détermination de tendances, la détection d'anomalie, etc. Toutes les approches présentées dans ce chapitre consistent en partie en des approches de fouille de données en ce sens qu'une partie de leur fonctionnement consiste à extraire de l'information de façon automatisée. Un exemple simple illustrant ce fait est le clustering : le clustering fait partie intégrante du domaine de la fouille de données (Han et Kamber, 2006), et peut être utilisé à la fois pour la recommandation basée sur le contenu et pour la recommandation basée sur les usages.

La détection de motifs et les modèles de Markov sont des approches de la fouille de données qui peuvent être utilisées pour calculer les recommandations. Dans cette section, après avoir introduit la notation, nous présentons ces approches et leur application à la recommandation. Nous présentons en tout trois approches de détection de motifs : les règles d'association, les motifs séquentiels et les modèles de Markov.

Notation

Soit $R = \{r_1, r_2, ..., r_m\}$ un ensemble de ressources, D un ensemble de transactions dans lequel chaque transaction $T \subseteq R$, A et B deux ensembles de ressources tels que $A \subset R$, $B \subset R$ et $A \cap B = \emptyset$.

3.3.1 Règles d'association

Une règle d'association est une implication de la forme $A \rightarrow B$ qui modélise le fait qu'un ensemble de ressources B est souvent consommé ou consulté quand un ensemble de ressources A a été consommé ou consulté. A est alors appelé *antécédent*, et B *conséquent*.

Une règle d'association $A \rightarrow B$ a une certaine capacité prédictive qui est mesurée selon deux critères, appelés *support* et *confiance*. Le support s d'une règle d'association $A \rightarrow B$ correspond au nombre d'occurrences des transactions de D qui contiennent $A \cup B$. Ce nombre

d'occurrences peut être normalisé par rapport au nombre de transactions de D. Le support peut donc prendre deux formes :

$$s(A \rightarrow B) = \begin{cases} P(A \cup B) & \text{dans le cas où une normalisation est effectuée} \\ C(A \cup B) & \text{sinon} \end{cases}$$

où $P(A \cup B)$ correspond à la probabilité de trouver $A \cup B$ dans D (pour une distribution uniforme sur D), et $C(A \cup B)$ correspond au nombre d'occurrences de $A \cup B$.

La normalisation n'a pas de véritable utilité, et est en général utilisée afin de permettre de parler en termes de probabilités ou de pourcentages. Cependant, face à un espace de données très important, les probabilités et pourcentages ainsi obtenus deviennent très faibles, et ne facilitent plus la lecture.

La confiance c d'une règle d'association $A \rightarrow B$ correspond à la probabilité conditionnelle des transactions contenant B sachant qu'elles contiennent A (pour une distribution uniforme sur D), c'est-à-dire :

$$c(A \rightarrow B) = P(B|A) = \frac{s(A \rightarrow B)}{s(A)}$$

La première tâche de l'utilisation de règles d'association est de découvrir ces règles en parcourant une base de données. Il est évidemment impossible de recenser toutes les règles qu'il est possible de constituer à partir de ces données, tant la combinatoire est importante. Par conséquent, seules les règles ayant la meilleure valeur prédictive sont conservées.

Cette première étape s'effectue en deux sous-étapes :

1. Recherche des règles ayant un support supérieur au seuil prédéterminé ;
2. Déduction des valeurs de confiance de ces règles et suppression des règles ayant une confiance inférieure au seuil prédéterminé.

Les valeurs de confiance pouvant être déduites des valeurs de support des règles, la seconde étape est beaucoup moins coûteuse en temps de calcul que la première, et l'essentiel de la problématique porte sur la recherche des règles. Cependant, l'extraction de ces règles ne peut être effectuée de façon naïve, car cela impliquerait un comptage des occurrences de chacune des règles qu'il est possible de construire avant de pouvoir effectuer les filtrages, et donc une grande complexité en temps et en mémoire. Par conséquent plusieurs techniques sont utilisées pour effectuer ce processus de façon plus efficace, en particulier l'algorithme *Apriori*.

L'algorithme *Apriori* est un algorithme célèbre de la fouille de données introduit en 1993 par (Agrawal *et al.*, 1993) pour extraire les règles d'associations d'un espace de données de façon incrémentale. Le principe général sur lequel est basé cet algorithme est que si un ensemble de ressources L_k de taille k n'a pas une valeur de support suffisante dans un espace de données D, alors tout sur-ensemble L_{k+1} de L_k de taille $k+1$ ne pourra avoir une valeur de support suffisante. Il suffit alors de procéder par valeur croissante de k, et de construire des *candidats*

à partir des règles obtenues à l'étape précédente pour obtenir les règles répondant aux contraintes de capacités prédictives fixées sans induire une trop grande complexité en temps et en mémoire.

Les principaux inconvénients de cet algorithme sont le nombre de candidats qu'il peut engendrer et le nombre important de passages sur la bases de données qu'il implique. D'autres algorithmes ont été proposés par la suite. Par exemple, l'algorithme FP-growth (Han *et al.*, 2000) permet de contourner le problème du nombre de candidats en utilisant une structure arborescente dont la construction se fait selon une technique « diviser pour régner » (*divide-and-conquer*).

Pour réduire encore le nombre de règles résultantes, les notions d'ensemble fermé et d'ensembles à fréquence maximale peuvent être utilisées.

Ensemble fermé *On appelle* ensemble fermé *un ensemble pour lequel il n'existe aucun sur-ensemble ayant le même support* (Han et Kamber, 2006).

Ensemble à fréquence maximale *On appelle* ensemble à fréquence maximale *un ensemble ayant une valeur de support supérieure au seuil prédéterminé et pour lequel il n'existe aucun sur-ensemble ayant une valeur de support supérieure au seuil prédéterminé* (Han et Kamber, 2006).

Utiliser la notion d'ensemble fermé permet d'extraire la même information de façon plus compacte, alors qu'utiliser la notion d'ensemble à fréquence maximale permet de réduire encore le nombre de règles mais implique une certaine perte d'information. Plusieurs algorithmes ont été proposés pour extraire des règles selon ces deux principes (Pasquier *et al.*, 1999 ; Pei *et al.*, 2000 ; Bayardo Jr, 1998 ; Burdick *et al.*, 2001).

a) Déduction des règles finales à partir des règles fréquentes

Comme mentionné précédemment, une fois que les règles fréquentes ont été extraites, la déduction des règles finales est une tâche relativement simple. Cette tâche se complique légèrement toutefois quand les notions d'ensemble fermé et d'ensemble à fréquence maximale sont utilisées. Il ne suffit alors pas de sélectionner les règles obtenues à la première étape en filtrant par rapport au seuil de confiance prédéterminé. En effet, dans le cas d'un ensemble fréquent mais ayant une valeur de confiance faible, une suppression ne serait pas appropriée car un sous-ensemble de cet ensemble aura peut-être une confiance répondant aux contraintes fixées. Il faut donc reconstruire la liste des sous-ensembles de chacune des règles obtenues à l'étape précédente et vérifier la confiance associée à chacun.

Les seuils de support et de confiance choisis sont généralement les mêmes quelle que soit la taille des motifs considérés. Cela permet à la fois une plus grande exigence dans la sélection des motifs les plus grands, et une plus grande couverture pour les motifs plus petits. À notre connaissance, une seule étude sur l'utilisation de seuils différents en fonction de la taille des motifs a été effectuée (Seno et Karypis, 2001).

b) Recommandation avec les règles d'association

Les règles obtenues en utilisant les méthodes présentées ci-dessus constituent alors un modèle, qui peut avoir de nombreuses utilisations. Parmi ces aplications, on peut citer la découverte de concepts ou de classes descriptives, les associations et corrélations, la classification, le clustering, l'analyse de tendances, l'analyse d'anomalies et de singularités, l'analyse de similarités, et en particulier, la prédiction de ressources (Han et Kamber, 2006).

Comme mentionné précédemment, la détection de motifs peut être utilisée pour recommander des ressources. Il suffit alors, étant donné un historique d'un utilisateur donné, de comparer tous les antécédents qu'il est possible de constituer à l'intérieur de cet historique aux antécédents des règles du modèle. Si un antécédent correspond, alors le conséquent qui lui est associé peut être recommandé.

Si plusieurs antécédents correspondent, il est possible de trier les recommandations obtenues en fonction de leur confiance et/ou de leur support. En revanche, quand plusieurs antécédents sont associés à un même conséquent, il faut combiner les valeurs de support et/ou de confiance correspondants. On associe alors à chaque recommandation possible un score qui peut être obtenu selon différentes politiques, les plus courantes étant les suivantes :

- politique de confiance maximale : seule la confiance maximale de chaque conséquent est conservée (Sarwar *et al.*, 2000 ; Wang *et al.*, 2005) ;
- politique de la somme : pour chaque conséquent, la somme des confiances associées à chaque antécédent est calculée (Kim et Kim, 2003) ;
- politique de la longueur maximale : les règles ayant les antécédents les plus longs sont d'abord utilisées pour fournir les recommandations (Nakagawa et Mobasher, 2003b). Cette politique peut être combinée avec les deux premières politiques.

Il est important de constater que les règles d'association sont extraites à partir de l'ensemble des historiques de consultation ou de consommation de tous les utilisateurs, ou d'un sous-groupe d'utilisateurs. En effet, exploiter les historiques d'un seul utilisateur ne permettrait pas de fournir des recommandations pertinentes, car leur nombre serait trop réduit, et les règles extraites n'auraient que peu d'intérêt du point de vue de la recommandation. Ainsi, cette approche peut être considérée comme une forme particulière de filtrage collaboratif en ce sens que les usages des utilisateurs sont mis en commun afin d'extraire des régularités intéressantes du point de vue de la recommandation.

3.3.2 Motifs séquentiels

Les motifs séquentiels sont la version séquentielle des règles d'association (Agrawal et Srikant, 1995 ; Lu *et al.*, 2005). Cette séquentialité représente une contrainte supplémentaire par rapport aux règles d'association, et permet de modéliser le comportement des utilisateurs de façon plus précise. Les séquences considérées peuvent être des suites d'ensembles de ressources (de transactions), ou plus simplement des suites de ressources (des suites de singletons *e.g.* $\langle a, b, c, d, e \rangle$).

Un motif séquentiel peut être dénoté par $A \circ B$, où $A \circ B$ est la concaténation de A et B, A est un antécédent séquentiel de taille quelconque, et B un conséquent séquentiel de taille quelconque.

Les notions de support et de confiance confiance utilisées pour les règles d'association sont également utilisées dans le cadre des motifs séquentiels. Le support s d'un motif séquentiel $A \circ B$ peut donc, ici encore, prendre les deux formes :

$$s(A \circ B) = \begin{cases} P(A \circ B) & \text{dans le cas où une normalisation est effectuée} \\ C(A \circ B) & \text{sinon} \end{cases}$$

où $C(A \circ B)$ correspond au nombre d'occurrences de $A \circ B$.

La confiance c d'un motif séquentiel $A \circ B$ est elle aussi définie de façon analogue à la confiance des règles d'association :

$$c(A \circ B) = P(B|A) = \frac{s(A \circ B)}{s(A)}$$

La recherche des motifs séquentiels au sein d'une base de données s'effectue de la même manière que la recherche de règles d'associations, c'est-à-dire en recherchant d'abord les motifs séquentiels ayant un certain support, puis en supprimant les motifs séquentiels ayant une confiance insuffisante.

La notion d'ensemble fermé des règles d'association peut, au même titre que la majorité des concepts utilisés pour les règles d'association, être étendue aux motifs séquentiels. Cependant, cette notion est également utilisée dans un autre sens. En effet, dans certains travaux, on parle de *motifs séquentiels ouverts* et de *motifs séquentiels fermés*. Dans ce cadre, un motif séquentiel ouvert correspond alors à une séquence non contiguë et un motif séquentiel fermé à une séquence contiguë (Mobasher, 2007). Par conséquent, un nouveau conflit terminologique apparaît. Dans cette thèse, on conservera le sens introduit dans la section précédente, et l'on évitera donc de parler de motifs séquentiels fermés quand il s'agit de motifs séquentiels contigus, et de motifs séquentiel ouverts quand il s'agit de motifs séquentiels non contigus.

Bien que les règles d'associations soient à l'origine des motifs séquentiels, l'adaptation aux motifs séquentiels des techniques de recherche de motifs des règles d'association engendre un temps de calcul trop important (Agrawal et Srikant, 1995). Un des premiers algorithmes pour extraire les motifs séquentiels est l'algorithme GSP pour *Generalized Sequential Pattern* (Srikant et Agrawal, 1996), qui est basé sur l'algorithme *Apriori*. Ici encore de nombreuses alternatives ont été proposées (Zaki, 1997 ; Masseglia *et al.*, 1998 ; Pei *et al.*, 2001 ; Yan *et al.*, 2003).

Enfin, les techniques des règles d'association utilisées pour calculer des recommandations peuvent également être utilisées pour les motifs séquentiels, selon les mêmes politiques de combinaison et de tri de recommandations.

3.3.3 Modèle de Markov

Les chaînes de Markov (Markov, 1971) modélisent des relations dans le temps selon une hypothèse d'indépendance telle que la probabilité d'un élément n'est dépendante que de l'élé-

ment précédent. Un modèle de Markov est défini par un ensemble d'*états* et un ensemble de *transitions* ayant chacune une probabilité. Une chaîne de Markov désigne les processus qui commencent dans l'un de ces états, et évoluent d'état en état selon les transitions. Un modèle de Markov d'ordre k est un modèle de Markov dans lequel les états sont constitués de k éléments. Dans ce cadre, les modèles de Markov constituent donc une forme de motifs séquentiels contigus de taille fixe.

Formellement, l'hypothèse d'indépendance entre les états d'un modèle de Markov d'ordre k peut être formulée de la façon suivante :

$$P(r_i \mid \langle r_1, ..., r_{i-1} \rangle) = P(r_i \mid \langle r_{i-k+1}, ..., r_{i-1} \rangle) \tag{1.4}$$

où $\langle r_1, ..., r_i \rangle$ est une suite de i ressources. Utilisés dans le cadre de la recommandation, la prédiction de la prochaine ressource r_i peut être estimée en fonction de l'état présent $\langle r_{i-k+1}, ..., r_{i-1} \rangle$. Les probabilités conditionnelles sont obtenues statistiquement sur un corpus d'apprentissage. Malgré leur grande simplicité, les modèles de Markov fournissent des prédictions étonnamment précises (Sarukkai, 2000).

L'utilisation des modèles de Markov implique un compromis entre précision et couverture. En effet, en filtrant les états les moins représentatifs, une meilleure précision peut généralement être obtenue. Cependant, plus ces états sont filtrés, moins on peut trouver de correspondances dans les historiques, ce qui résulte en une plus faible couverture. Une autre manière d'augmenter la précision est d'augmenter la valeur de k. Cependant, au-delà d'une certaine valeur de k il devient difficile de trouver des données suffisantes pour obtenir un modèle représentatif. Si les données sont insuffisantes, le modèle obtenu aura une plus faible couverture et fournira une couverture moindre que pour des valeurs plus petites de k. Si les données sont suffisantes, le modèle obtenu pourrait avoir une trop grande complexité en mémoire. C'est pourquoi la taille des états considérés est habituellement restreinte.

Un moyen de fournir une couverture quasi totale tout en conservant une bonne précision est d'utiliser des modèles de Markov de différents ordres variant de k à 1 : quand aucun historique de taille k ne correspond à l'historique de l'utilisateur actif, un historique de taille $k-1$ est utilisé, et ainsi de suite, jusqu'à ce qu'une correspondance soit trouvée, ou que k vaille 0 ce qui correspond à la probabilité d'une ressource indépendamment de l'état présent. En utilisant un tel schéma, il est possible d'obtenir à la fois une couverture totale et une bonne précision dans les prédictions. Ce schéma est appelé all-k^{th}-order Markov model (Pitkow et Pirolli, 1999). La complexité en temps d'un all-k^{th}-order est basse, mais la complexité en mémoire est grande, car il faut manipuler un modèle par valeur de k. Il est important de remarquer que dans les mêmes conditions de filtrage, le all-k^{th}-order Markov order est identique aux motifs séquentiels contigus où chaque élément des séquences est un singleton (Nakagawa et Mobasher, 2003b).

3.4 Limitations des approches basées sur la détection de motifs

Une des principales limitations de la recommandation basée sur la détection de motifs est qu'elle n'est applicable qu'à un espace de données relativement réduit. Une première solu-

tion pour traiter des espaces de données plus importants est de s'équiper d'une infrastructure conséquente, comme l'infrastructure distribuée utilisée par Google. Cependant cette solution n'est financièrement pas à la portée de tout un chacun (Hölzle, 2009). Une autre solution est de répartir les ressources ou les sessions de navigation en sous-groupes homogènes, de façon supervisée ou non (*cf.* section 3.1.2 de ce chapitre). Cette solution a un effet doublement positif : en plus de rendre possible l'application des approches basées sur la détection de motifs sur les sous-groupes obtenus, plus les données sur lesquelles sont appliquées ces approches sont homogènes, plus la taille mémoire nécessaire pour les modéliser est réduite.

Une seconde limitation est liée à la première : si le nombre distinct de ressources de l'espace de données est important, il devient difficile de trouver suffisamment de données d'apprentissage pour construire un modèle représentatif. En plus du problème de la complexité en temps et en mémoire, cela engendre à la fois un problème de précision et de couverture. Une solution face à ce problème est d'extraire des motifs de classes de ressources.

Tout comme le filtrage collaboratif, quand une nouvelle ressource est introduite dans le système, il faut qu'elle ait été consultée ou consommée un certain nombre de fois avant qu'elle puisse être recommandée de façon pertinente. Cependant, contrairement aux approches de la recommandation basée sur le contenu et au filtrage collaboratif, l'identification des utilisateurs n'est pas nécessaire. Ainsi, un nouvel utilisateur arrivant sur le système bénéficiera de recommandations de la même qualité que les autres utilisateurs.

D'une manière générale, les approches basées sur la détection de motifs impliquent un compromis entre précision, couverture et complexité en temps et en mémoire qui reste problématique.

* * *

Ayant donné un aperçu des approches principales des systèmes de recommandation, l'observation que l'on peut faire est qu'aucune approche n'est véritablement meilleure qu'une autre, et que beaucoup offrent des avantages complémentaires. Les approches basées sur le contenu sont précises, mais ne prennent en compte que le contenu des ressouces, quand d'autres critères sont déterminants des appréciations des utilisateurs. De plus, ces approches ne sont véritablement applicables que sur des ressources textuelles. Les approches basées sur les usages quant à elles, bien que pouvant être appliquées à n'importe quel type de données, possèdent également leurs limitations. En particulier, le filtrage collaboratif souffre entre autres du démarrage à froid et du manque de données, et les approches basées sur la détection de motifs ne sont applicables que sur un espace de données réduit. Dans un tel contexte, il est naturel d'envisager de combiner ces approches afin d'en retirer les avantages tout en tentant d'en limiter les inconvénients respectifs.

4 APPROCHES HYBRIDES

Dans cette section, après avoir brièvement présenté les différentes techniques d'hybridation, nous nous intéressons à l'efficacité des modèles hybrides en nous basant principalement

sur le travail de synthèse de (Burke, 2007). Puis nous présentons plus particulièrement une catégorie de méthodes hybrides peu explorée dans la littérature : la combinaison d'approches basées sur les motifs avec d'autres approches de recommandation.

4.1 Techniques d'hybridation

L'essentiel de la problématique de l'hybridation réside dans la combinaison des différentes approches de l'état de l'art. Robin Burke comptabilise sept techniques principales (Burke, 2007) :

1. *Weighted* : Interpolation des scores des différentes recommandations ;
2. *Switching* : Détermination de la technique de recommandation la plus appropriée au cas par cas ;
3. *Mixed* : Concaténation des recommandations issues des différentes techniques dans une même liste ;
4. *Feature Combination* : Utilisation de la combinaison d'attributs provenant de techniques différentes ;
5. *Feature Augmentation* : Utilisation d'une des techniques pour calculer un attribut ou un ensemble d'attributs, qui sont ensuite utilisés par la seconde technique ;
6. *Cascade* : instauration d'une hiérarchie au sein des modèles, les moins haut placés servant à renforcer les scores de ressources obtenus avec les modèles plus haut placés.
7. *Meta-level* : La première technique est utilisée pour construire un modèle, qui est utilisé par la deuxième technique.

4.2 Efficacité des approches hybrides

Bien que plusieurs approches aient été proposées depuis plusieurs années, l'efficacité des modèles hybrides a véritablement été mise en avant lors du concours Netflix. En effet, le candidat ayant obtenu les meilleurs résultats a proposé une approche hybride mélangeant pas moins de 107 modèles (Bell *et al.*, 2007). Ces 107 modèles consistent en réalité en des variantes très proches les unes des autres de 5 modèles de base.

Dans (Burke, 2007), différentes techniques d'hybridation sont présentées de façon rigoureuse. L'auteur classifie les approches de recommandation pure en 5 catégories : la recommandation basées sur le contenu, le filtrage collaboratif, la recommandation basée sur les données démographiques, la recommandation basée sur l'utilité et la recommandation basée sur la connaissance. Un certain nombre de techniques est mis en avant pour combiner ces 5 catégories d'approche, induisant 41 possibilités de modèles hybrides. Ces 41 modèles sont alors comparés expérimentalement sur un corpus de recommandation de restaurants. En dehors d'une très grande amélioration de la précision des recommandations par rapport aux modèles standards, les résultats montrent que les meilleures stratégies pour combiner les modèles sont les stratégies de la *feature augmentation* et de la *cascade*. Aussi, les combinaisons ayant fourni les

meilleurs résultats sont des combinaisons d'approches de recommandation basée sur le contenu avec l'approche utilisant la corrélation de Pearson du filtrage collaboratif.

Plusieurs éléments mettent un bémol à ces résultats : (1) les données considérées contiennent des notes attribuées implicitement, (2) la complexité en temps n'est pas considérée et (3) les approches basées sur la détection de motifs ne sont pas considérées.

Les approches basées sur la détection de motifs, et plus précisément de motifs séquentiels, sont celles qui nous intéressent particulièrement dans cette thèse. Dans le reste de cette section, nous présentons donc quelques approches hybrides intégrant la notion de motifs.

4.3 Hybridation impliquant la détection de motifs

Nous nous intéressons à présent aux travaux combinant les approches basées sur la détection de motifs avec d'autres approches.

4.3.1 Forme séquentielle de la recommandation à partir de cas

Dans (Trousse, 2000), un modèle basé sur le raisonnement à partir de cas (*cf.* section 2.2.1 de ce chapitre) est utilisé pour prédire le comportement d'utilisateurs navigant sur le Web. Le principal apport de ce travail est l'inclusion de critères temporels dans la définition des cas. En effet, les cas sont ici composés d'un contexte temporel, d'une liste de pages qui peuvent être recommandées dans ce contexte et d'un ensemble de données utilisées pour calculer la similarité entre les cas. Cette similarité dépend des trois dernières pages consultées et de l'ensemble des pages précédemment consultées dont une appréciation a pu être obtenue. Puis une liste de recommandations est obtenue en combinant les listes de pages obtenues à l'étape précédente, en fonction du nombre de cas ayant mené à la recommandation de chaque page.

Le modèle implique donc l'utilisation de notes pour considérer un grand historique. Si aucune note n'est disponible il se comporte alors comme un modèle de Markov d'ordre 3. Si des notes sont utilisées, alors il est proche d'un mélange entre un modèle de Markov d'ordre 3 et un modèle de motifs séquentiels.

4.3.2 Séquences de clusters

Dans (Yeong *et al.*, 2005), une approche de filtrage collaboratif est combinée avec des motifs séquentiels. L'idée derrière cette combinaison est que les opérations commerciales sont régies par une certaine périodicité qui peut être prise en compte pour affiner les recommandations du filtrage collaboratif. Dans un premier temps, l'algorithme *Self-Organizing Map* (Kohonen, 1990) est utilisé pour regrouper les ressources en clusters homogènes. Des motifs séquentiels de clusters sont alors extraits. Puis, pendant la phase de recommandation, étant donné un utilisateur, le cluster auquel appartiendront les prochaines ressources qu'il est susceptible de consulter est prédit. Enfin, les ressources les plus fréquemment consommées de ce cluster sont recommandées. Les résultats montrent que la qualité des recommandations peut être améliorée pour les acheteurs les plus actifs.

4.3.3 Forme séquentielle des réseaux de dépendance

(Zimdars *et al.*, 2001) proposent d'intégrer la notion de temporalité dans le filtrage collaboratif. Ils proposent trois techniques pour transformer des historiques de notes ordonnés chronologiquement en une représentation pouvant être utilisée avec les algorithmes traditionnels de filtrage collaboratif.

- Sacs de notes (*bag-of-votes*) : il s'agit d'une transformation telle que le résultat correspond au filtrage collaboratif standard, c'est-à-dire où l'ordre chronologique des notes n'est pas conservé. Plus précisément, pour chaque historique de notes, un ensemble de valeurs indiquant si la ressource a été appréciée est créé. On se retrouve ainsi avec une matrice de notes comme on a l'habitude d'en rencontrer pour le filtrage collaboratif [7].
- Mise en boîte (*binning transformation*) : cette technique consiste à séparer dans un premier temps les historiques dans des boîtes en fonction de leur taille puis de créer un modèle par boîte en utilisant le principe des sacs de notes. Ensuite, lors d'une prédiction, un des modèles est choisi en fonction de la taille de l'historique considéré. Cette transformation ne conserve pas l'ordre chronologique des notes non plus, mais exploite la structure temporelle pour fournir une forme de contextualisation des recommandations et un droit à l'oubli.
- Expansion des données (*data expansion*) : cette transformation semble rendre le modèle très similaire aux modèles de Markov. Pour chaque historique, chaque ressource k est associée à ce que les auteurs appellent un cas. Chaque cas consiste en : (1) une variable indiquant si la prochaine note est pour k, (2) j variables indiquant si la $j^{\text{ème}}$ précédente note était pour k et (3) une variable indiquant si k a déjà été notée (ce qui constitue un élément supplémentaire par rapport à une chaîne de Markov traditionnelle).

Ces trois approches sont alors combinées avec le modèle de réseau de dépendances (Heckerman *et al.*, 2001), présenté dans la section 1.1 de ce chapitre. Ces combinaisons entrent donc dans le cadre d'une hybridation d'approches par motifs et d'une approche de filtrage collaboratif. Une comparaison des performances respectives a été effectuée sur deux corpus de navigation Web. Deux mesures de la qualité ont été utilisées : la valeur de la précision du filtrage collaboratif selon le critère de Heckerman *et al.* (Heckerman *et al.*, 2000) et la probabilité logarithmique des notes. Pour les deux corpus, les deux dernières transformations ont fourni de meilleurs résultats que celle des sacs de votes. Cependant, les résultats du modèle d'expansion des données sont supérieurs selon la première mesure alors que ce sont ceux du modèle de mise en boîte qui le sont selon la seconde mesure.

Les détails fournis dans ce travail étant limités, il est difficile de savoir à quel point le modèle combinant l'expansion de données aux arbres de décision est proche d'un modèle de Markov standard. On remarquera que les auteurs font le choix de considérer qu'une ressource a été appréciée quand elle figure au moins une fois dans l'historique de l'utilisateur, ce qui est équivalent à utiliser un modèle de Markov sans utiliser de notes, et renforce le sentiment d'un modèle

7. Le fait que les données soient ici détemporalisées n'est pas contradictoire avec la proposition d'intégration de temporalité dans le filtrage collaboratif annoncée au début de la présentation de ce travail car elle ne porte que sur les données et non sur l'approche.

similaire à un simple modèle de Markov. De même, la mise en boîte semble correspondre à un modèle de règles d'associations dans lequel aucun filtrage sur les capacités prédictives n'est effectué.

4.3.4 Processus de décision markovien

Dans un travail successif au travail présenté précédemment, (Shani *et al.*, 2005) proposent de considérer la recommandation d'information non plus comme un procédé de prédiction sur des séries temporelles, mais comme un problème de décision sur des séries temporelles. Ils appliquent donc le principe des MDP (*Markov Decision Processes*, ou « processus de décision markovien »). Cela constitue une forme de système de recommandation hybride en ce sens qu'il combine le filtrage collaboratif avec la détection de motifs.

Ce travail ayant été prédestiné à être expérimenté en conditions réelles sur un site de vente en ligne de livres, il n'était pas envisageable de laisser le modèle s'initialiser directement en ligne par le biais d'utilisateurs réels et un premier modèle appelé *modèle prédictif* et basé sur les modèles de Markov a été développé. Dans cette section, nous présentons dans un premier temps le modèle prédictif, puis le modèle utilisant les MDP.

a) Le modèle prédictif

Le modèle prédictif est un modèle de Markov basique avec trois améliorations :

- skipping : possibilité de sauter des ressources. Par exemple, si un utilisateur a acheté trois ressources r_1, r_2 et r_3 successivement, il est possible que quelqu'un d'autre achète la ressource r_3 après avoir acheté la ressource r_1 ;
- regroupement des états similaires en fonction des éléments communs qui les composent ;
- combinaison de modèles de Markov de différents ordres selon une interpolation linéaire.

Comme mentionné précédemment, les auteurs avaient l'opportunité d'effectuer des évaluations sur des transactions réelles via un site de vente en ligne. Ils ont donc utilisé deux corpus : un corpus de transactions, et un corpus de navigation sur le site. Plusieurs versions du modèle prédictif ont été comparées en faisant varier l'ordre des modèles de Markov et en omettant une ou plusieurs des trois améliorations proposées. Un équivalent atemporel (le même modèle avec les mêmes variations, mais sans tenir compte de l'ordre des ressources) a également été évalué ainsi que le modèle de réseaux de dépendance issu de (Heckerman *et al.*, 2001), présenté dans la section 1.1 de ce chapitre et utilisé dans (Zimdars *et al.*, 2001). Ce dernier modèle a été lui aussi évalué selon les deux versions : séquentiel et non séquentiel, la version séquentielle correspondant probablement au modèle utilisant l'expansion de données de (Zimdars *et al.*, 2001).

Les résultats montrent que les modèles utilisant les modèles de Markov avec les trois améliorations en mode séquentiel sont les plus performants. De plus, le modèle de réseaux de dépendance dans sa forme non séquentielle s'est avéré le moins performant. Ce résultat est très in-

téressant, car il montre qu'une approche basée exclusivement sur la détection de motifs séquentiels peut s'avérer être plus précise à la fois qu'une approche hybride de filtrage collaboratif et de détection de motifs séquentiels.

De plus, ce modèle comporte des améliorations assez arbitraires. Par exemple, il existe dans la littérature de nombreuses variantes pour le skipping, et une version optimisée aurait pu être déterminée. En effet, la forme utilisée ajoute simplement 2^{-d} aux occurrences des transitions observées entre les états, d étant la distance à l'état futur. De même, les poids utilisés pour les interpolations linéaires sont fixés simplement à $1/k$.

b) Le modèle de MDP

Le formalisme MDP consiste en (1) un ensemble d'états, (2) un ensemble d'actions, (3) une fonction de transition entre les états (en utilisant les actions) et (4) une fonction de récompense associée à chaque action pour chaque état. Le but est de maximiser cette récompense selon une politique optimale (fonction associant une action à chaque état).

Dans ce modèle, les états correspondent à des k-tuples d'objets (livres, disques, etc.), les actions aux recommandations et les récompenses à l'utilité de vendre un objet. La politique optimale est obtenue en utilisant le principe de l'itération de la politique de Howard. Enfin, le modèle est mis à jour environ une fois par semaine, hors ligne. En effet une mise à jour en temps réel n'est pas envisageable au vu du temps de calcul que cela représente.

Le modèle a été évalué sur une période de presque deux ans sur le site de vente en ligne, proposant environ 15 000 objets pour les transactions, et recevant environ 5 000 visiteurs par jour. Le critère d'évaluation est le profit en shekels (monnaie israélienne). Une première évaluation sur des variations dans la politique montre que c'est leur politique optimale qui est la plus performante, bien que la différence de profit avec des politiques prédictives soit petite. Sur une courte période, ils comparent directement les profits obtenus avec le modèle de MDP à ceux d'avec le modèle prédictif (celui utilisé pour initialiser le modèle MDP), et obtiennent un profit supérieur de 28% avec le premier modèle.

Leur modèle semble donc effectivement plus performant pour ce type de problème qu'un modèle prédictif. Cependant le modèle prédictif expérimenté est loin d'être optimisé : d'abord, comme mentionné précédemment, les améliorations proposées pourraient elles même être optimisées. Également, il existe d'autres méthodes qui pourraient encore améliorer les résultats. Un second reproche est le critère d'évaluation utilisé. La maximisation du profit ne représente pas forcément le meilleur critère de satisfaction des utilisateurs. De plus, cela ne permet pas de comparaison de leur modèle de MDP avec les autres modèles de l'état de l'art selon les critères traditionnels.

5 CONCLUSION

Les approches des systèmes de recommandation sont particulièrement variées, et peuvent être classées de différentes manières. Dans ce chapitre, nous avons proposé une séparation

en deux approches principales : la recommandation basée sur le contenu et la recommandation basée sur les usages. Cette séparation simple nous a permis le regroupement d'approches habituellement présentées séparément, qui a son tour nous a permis de mettre en avant des similarités entre ces approches. En particulier, la recommandation à partir de cas, la recommandation basée sur la démographie, la recommandation basée sur l'utilité et la recommandation basée sur la connaissance peuvent être considérées comme des formes particulières de recommandation basée sur le contenu. De même, la recommandation basée sur la détection de motifs peut être considérée comme une forme particulière de filtrage collaboratif.

Toutes ces approches présentent néanmoins des caractéristiques complémentaires. Par conséquent un grand nombre de travaux se sont intéressé à différentes techniques d'hybridation, qui en plus de permettre de profiter des avantages respectifs de ces approches, s'avèrent fournir des recommandations plus précises. En particulier, comme le montrent les travaux présentés dans la section 4.3 de ce chapitre, la prise en compte de la notion de séquentialité, pourtant peu explorée dans le cadre des approches hybrides, peut améliorer la qualité des recommandations. Plus intéressant encore, cette notion utilisée seule peut fournir une précision supérieure à celle d'approches hybrides, comme le montrent (Shani *et al.*, 2005). C'est donc à cette notion particulière que nous nous intéresserons dans cette thèse.

Chapitre 2

Exploitation de la notion de séquentialité pour la recommandation Web

Comme nous l'avons vu dans le chapitre précédent, exploiter l'ordre chronologique de consultation ou de consommation des ressources peut améliorer la qualité des recommandations. Cependant cette stratégie semble plus appropriée pour certains types de données que pour d'autres. Les données pour lesquelles la recommandation est la plus utilisée sont les données cinématographiques (Miller *et al.*, 2003) et les ressources Web (Pitkow et Pirolli, 1999). D'autres travaux portent sur la recommandation de musique (Su *et al.*, 2010), de livres (Mooney et Roy, 2000), de blagues (Miyahara et Pazzani, 2000), de restaurants (Burke, 2007), de news (Das *et al.*, 2007), d'articles scientifiques (Pavlov *et al.*, 2004), etc.

Dans ce chapitre, nous discutons dans un premier temps de la séquentialité de deux types de données parmi les plus répandus, à savoir les données cinématographiques et les ressources Web. Dans cette discussion, nous verrons que la navigation Web se prête particulièrement à une modélisation basée sur la notion de séquentialité. Dans un deuxième temps, nous présentons le domaine de la modélisation statistique du langage (MSL), qui, de notre point de vue, a des caractéristiques similaires à celles de la navigation Web, et discutons de la meilleure façon d'utiliser les modèles de la MSL pour la recommandation. Enfin, nous présentons le protocole expérimental relatif à la problématique de cette thèse, qui concerne l'exploitation de la séquentialité pour la recommandation Web.

1 SÉQUENTIALITÉ DES TYPES DE DONNÉES

Dans cette section une discussion sur la séquentialité de deux types de données est présentée. Ces données sont les données cinématographiques et les ressources Web. On parle alors de recommandation de films et de recommandation Web.

1.1 Recommandation de films

S'il n'y a pas d'ordre chronologique fondamentalement cohérent dans le visionnage de films, et encore moins dans la notation de films, certaines notions séquentielles y résident. En particulier, certains films peuvent être davantage appréciés après en avoir vu certains autres. On peut par exemple apprécier de retrouver l'atmosphère d'un film dans le cadre de suites (*e.g.* les six épisodes de la *Guerre de étoiles*, *Schrek* et ses trois suites, etc.). Aussi, certains films consistent en des parodies d'autres films, ou contiennent des références qui ne sont appréciables que si l'on a vu les films en question (*e.g.* « *Top Gun* » pour « *Hot Shots* », « *Cliffhanger* » pour « *Ace Ventura en Afrique* », « *Terminator* » pour « *Last Action Hero* », etc.).

Un aspect psychologique lié à la notion d'ordre chronologique a été mis en avant par Gavin Potter, lors du concours Netflix (Ellenberg, 2008) : les notes attribuées par les utilisateurs dépendent des notes qu'ils viennent d'attribuer. Par exemple, un utilisateur qui vient de mettre quatre étoiles à un film, et doit ensuite noter un autre film qu'il a préféré au premier, mettra probablement cinq étoiles, alors que cela n'aurait peut-être pas été le cas si le second film lui avait été présenté directement. En se basant, entre autres, sur cette considération, Gavin Potter a longtemps figuré en bonne position lors du concours.

Ce dernier comportement est assimilable à ce qui est appelé « théorie de l'engagement » en sociologie (Joule et Beauvois, 1987) : en attribuant la première note, l'utilisateur s'est engagé dans une décision. Ainsi, mettre cinq étoiles au second film, ne contredit pas sa décision précédente, alors que tout autre note impliquerait une telle contradiction. Ce comportement peut être généralisé à d'autres critères : par exemple, il est fréquent que l'on apprécie un film d'un réalisateur particulier, et que l'on s'intéresse ensuite aux autres films qu'il a réalisés. Or, en faisant cela, il est possible que l'on s'engage dans l'appréciation dudit réalisateur, et l'on aura plus de mal à mal noter ses autres films, car cela impliquerait une contradiction avec notre engagement passé.

En dehors de ces considérations, le visionnage et la notation de films ne semblent pas suivre de motifs séquentiels particuliers : la plupart du temps, l'ordre chronologique dans lequel les films sont regardés semble grandement aléatoire, tout comme l'est celui dans lequel ils sont notés, sinon davantage puisque les systèmes de recommandations peuvent permettre de noter des listes aléatoires de films[1]. Par conséquent, la notion de séquence peut être utilisée pour améliorer des systèmes de recommandation d'œuvres cinématographiques exploitant d'autres critères, mais ne peut représenter un critère fondamental de tels systèmes.

1.2 Recommandation Web

Le Web peut être hiérarchisé selon deux niveaux : un ensemble de sites contenant un ensemble de ressources, telles que des pages Web, des documents à télécharger, etc. Il est donc possible de recommander des sites ou des ressources, dans le cadre restreint d'un site Web, ou dans le cadre du Web global.

1. `http://www.movielens.org/`

De par sa structure en hyperliens, la navigation Web est fondamentalement régie par des contraintes séquentielles : les utilisateurs naviguent de page en page en cliquant sur des hyperliens, et dessinent ainsi des chemins qui sont déterminants de leurs intentions. Il est ainsi possible, uniquement à partir de la suite de ressources qu'un utilisateur vient de consulter, d'estimer les ressources par lesquelles il est le plus susceptible d'être intéressé. Cette approche est également désignée par l'expression « aide à la navigation » (Trousse *et al.*, 1999) ou « *Clickstream-Based Collaborative Filtering* » (Kim *et al.*, 2004). L'amalgame avec le filtrage collaboratif est approprié en ce sens que, comme mentionné dans le chapitre 1, les approches par détection de motifs peuvent être considérées comme une forme particulière de filtrage collaboratif.

Une autre possibilité pour effectuer les recommandations est de mettre en place une interface dans laquelle les utilisateurs attribuent des notes explicites à une liste de sites ou ressources Web (Balabanović et Shoham, 1997). Dans ce cadre, le problème devient très similaire à celui de la notation de films, et la séquentialité est donc beaucoup moins exploitable. En revanche, dans le cadre d'une navigation, nous pensons que l'utilisation de notes explicites est beaucoup plus laborieuse, et que le nombre de notes récupérées serait trop faible pour être exploitable.

Une possibilité entre les deux consiste à calculer des notes implicites à partir des usages de l'utilisateur (Chan, 1999 ; Claypool *et al.*, 2001 ; Fox *et al.*, 2005 ; Castagnos, 2008 ; Esslimani *et al.*, 2009). En se plaçant dans ce cadre, l'ordre chronologique de consultation des ressources peut être exploité pour regrouper les utilisateurs en communautés homogènes, ce qui peut améliorer la précision de la prédiction de notes en comparaison des approches standard du filtrage collaboratif (Esslimani *et al.*, 2008).

La navigation Web est donc un domaine d'application des systèmes de recommandation où la notion de séquence est particulièrement appropriée. C'est donc à ce domaine que nous nous intéresserons dans cette thèse.

2 ÉTAT DE L'ART DE LA RECOMMANDATION POUR LA NAVIGATION WEB

Dans cette section, nous nous intéressons à l'applicabilité et à l'efficacité respectives des approches de recommandation présentées dans le chapitre 1 pour la navigation Web.

2.1 Recommandation basée sur le contenu

La recommandation basée sur le contenu n'est *a priori* pas appropriée dans le cadre de la navigation Web (Kim *et al.*, 2004). En effet, ces approches recommandent des ressources ayant un contenu similaire aux ressources appréciées par l'utilisateur. Or, le but d'un utilisateur naviguant sur le Web peut correspondre à un contenu totalement différent des pages qu'il est en train de consulter. Par exemple, un utilisateur naviguant sur le site Web d'un professeur peut être en train de rechercher une publication en particulier. Dans ce cas, on peut supposer qu'une approche par contenu recommanderait d'autres pages de professeurs, qui n'intéressent pas l'utilisateur.

2.2 Recommandation basée sur les usages

2.2.1 Filtrage collaboratif

Dans (Breese *et al.*, 1998), plusieurs modèles de filtrage collaboratif sont expérimentés sur un corpus de navigation Web. Le modèle fournissant les meilleurs résultats est un modèle de réseau bayésien avec des arbres de décision pour représenter les dépendances conditionnelles (*cf.* section 1.1 du chapitre 1). Les autres approches de filtrage collaboratif expérimentées sont un classifieur naïf de Bayes et plusieurs variantes d'approches mémoire.

Pour les mêmes raisons que les approches basées sur le contenu, les approches modèle du filtrage collaboratif utilisant la classification ou le clustering ne sont *a priori* pas optimales (Kim *et al.*, 2004 ; Linden *et al.*, 2003).

2.2.2 Détection de motifs

Dans le cadre de la navigation Web, la détection de motifs est en rapport direct avec le domaine du *Web Usage Mining* (Cooley *et al.*, 1997). Le Web Usage Mining est un sous-domaine de la fouille de données focalisé sur l'extraction de régularités à partir des traces d'usage des utilisateurs du Web. Un des aspects de ce domaine est la modélisation prédictive du Web (Pitkow et Pirolli, 1999).

La recommandation pour la navigation Web est une des applications possibles de la modélisation prédictive du Web. D'autres applications sont la recherche d'information (Brin et Page, 1998), la modélisation de sites Web afin d'en améliorer l'ergonomie (Chi *et al.*, 1998), la réduction de la latence (Kroeger *et al.*, 1997), etc. Par conséquent, tous les travaux liés à la problématique de la modélisation prédictive du Web sont intéressants du point de vue de la recommandation.

Un des travaux les plus retentissants de la modélisation prédictive du Web est celui de (Pitkow et Pirolli, 1999). Dans ce travail, un all-k^{th}-order Markov model est expérimenté afin de répondre au problème de compromis entre précision et couverture des modèles de Markov. Or, si cette technique permet de fournir une couverture totale, elle implique un compromis entre précision et complexité en mémoire. La problématique est alors de réduire cette complexité en mémoire sans réduire la précision. (Pitkow et Pirolli, 1999) proposent d'utiliser des *Longest Repeated Subsequences* (les sous-séquences les plus longues et les plus fréquentes), qui sont proches de motifs séquentiels contigus à fréquence maximale (*cf.* section 3.3 du chapitre 1). Cependant, la précision obtenue avec ce dernier modèle reste réduite. C'est pourquoi beaucoup de travaux ont été effectués par la suite pour trouver un meilleur compromis.

Dans (Deshpande et Karypis, 2004), un all-k^{th}-order Markov model est utilisé avec trois schémas du filtrage sur les états : un schéma de filtrage sur les occurrences dans lequel un même seuil d'occurrences est utilisé pour tous les ordres des modèles de Markov, un schéma de filtrage dans lequel les états sont supprimés si la différence de probabilité entre les deux ressources les plus proéminentes n'est pas statistiquement significative et un schéma de filtrage qui supprime des états en fonction des erreurs de prédictions observées sur un corpus

de validation. (Borges et Levene, 2005) propose de transformer des modèles de Markov du premier ordre en un modèle qui représente des états d'ordre supérieurs par des opérations de clonage.

Le principal inconvénient de l'utilisation des modèles de Markov pour la navigation Web, est que seules des suites contiguës de ressources sont considérées. Par conséquent des phénomènes tels que les erreurs de navigation ou les navigations parallèles ne peuvent pas être pris en compte. Considérons par exemple la suite de ressources $\langle a, b, c, x \rangle$, dans laquelle $\langle a, b, c \rangle$ correspond au début d'une navigation, et x correspond à une erreur. Pour prédire la prochaine ressource, un all-k^{th}-order Markov model recherchera la suite de ressources $\langle a, b, c, x \rangle$ dans la liste de ses états. Il est probable que, puisque x correspond à une erreur de navigation, aucune correspondance ne soit trouvée. Le modèle d'ordre inférieur sera alors utilisé, et le même problème se posera, et ainsi de suite, jusqu'à ce que l'état considéré soit réduit à x. Cet état se trouvera probablement dans la liste, et fournira probablement une plus mauvaise prédiction que si la suite de ressources $\langle a, b, c \rangle$ avait été exploitée.

Un façon de remédier à ce dernier problème est d'utiliser des motifs séquentiels non contigus, ou des règles d'association (*cf.* section 3.3 du chapitre 1). Dans (Nakagawa et Mobasher, 2003b), une comparaison empirique des règles d'association et des motifs séquentiels, contigus et non contigus, est présentée. Les résultats montrent que les règles d'association et les motifs séquentiels non contigus sont plus appropriés pour de petites sessions et les sites ayant une grande connectivité ; et que les motifs séquentiels contigus sont plus appropriés pour les plus grandes sessions. Cependant, ces expérimentations ont été réalisées avec des fenêtres glissantes de très petites tailles (3 et 4), et il est probable que l'utilisation de plus grandes fenêtres induise des résultats différents.

2.3 Approches hybrides

Deux travaux que nous avons classés parmi les approches hybrides dans le chapitre 1 (Zimdars *et al.*, 2001) et (Shani *et al.*, 2005) présentent des résultats d'expérimentations sur des données de navigation Web. Les modèles expérimentés sont, entre autres, un modèle hybride combinant le modèle de réseau de dépendances de (Heckerman *et al.*, 2001) avec un modèle de Markov et un modèle de Markov amélioré (sans hybridation), appelé modèle prédictif. Or, le modèle le plus performant s'avère être le modèle prédictif. Il semble donc que les modèles séquentiels de ce type soient particulièrement efficaces sur les données de navigation Web.

Il semble également que la qualité des recommandations Web peut être améliorée en combinant les approches basées sur la détection de motifs avec d'autres approches, en particulier avec les approches basées sur le contenu. Dans (Li et Zaiane, 2004), un modèle hybride de règles d'association et de recommandation basée sur le contenu est proposé. Les résultats montrent une nette amélioration de la précision en comparaison du modèle simple de règles d'association quand le modèle hybride est utilisé. Ce type d'approche semble également approprié pour les motifs séquentiels.

Quelques travaux combinent les modèles basés sur la détection de motifs avec la structure de sites Web (Zukerman *et al.*, 1999 ; Nakagawa et Mobasher, 2003a). Cela constitue une

forme de combinaison avec les approches basées sur le contenu car cela implique de considérer le contenu des pages. Le premier travail ne compare que des modèles à séquences contiguës (des variantes de modèles de Markov d'ordre 1 et 2), ce qui ne permet pas de comparaison avec d'autres modèles basés sur les motifs tels que les règles d'association et les motifs séquentiels. Le second ne montre pas d'amélioration significative. De plus, comme dans le travail des mêmes auteurs mentionné précédemment, les expérimentations ont été réalisées avec des fenêtres de petite taille, qui ne sont *a priori* pas optimales pour les règles d'association et les motifs séquentiels.

2.4 Conclusion

Dans cette section, nous nous sommes intéressé à la notion de séquentialité pour la recommandation. Après une discussion sur la séquentialité de deux des types de données parmi les plus répandus de la recommandation, nous avons décidé de nous focaliser sur la navigation Web.

Nous avons alors présenté l'état de l'art de la recommandation pour la navigation Web dans laquelle nous avons mis en avant la plus grande efficacité des approches séquentielles, telles que les motifs séquentiels et les modèles de Markov. Ces approches soulèvent deux grandes problématiques spécifiques de la navigation Web. La première est le compromis entre précision, couverture et complexité en temps et en mémoire ; la seconde est la résistance au bruit et la prise en compte des navigations parallèles.

La séquentialité est largement prise en compte en modélisation statistique du langage. Or, cette approche, qui a montré son efficacité dans de nombreuses applications, possède de nombreuses similitudes avec la modélisation prédictive du Web. C'est pourquoi nous étudions ses apports à la recommandation dans la suite de ce chapitre.

3 VERS L'EXPLOITATION DE LA MODÉLISATION STATISTIQUE DU LANGAGE

Dans cette thèse, nous prétendons que la navigation Web et le langage naturel ont de nombreuses similarités. Afin de pouvoir comparer les deux domaines, il convient de les présenter dans un premier temps. La navigation Web et sa modélisation ont été présentées dans la section précédente. La modélisation du langage est donc brièvement présentée dans celle-ci. L'accent est mis sur deux approches de modélisation statistique du langage prometteuses dans l'optique de la modélisation prédictive du Web.

Le langage naturel peut être modélisé selon deux approches : les approches linguistiques, et les approches statistiques. Étant donnés les progrès technologiques de ces dernières années, les techniques statistiques se sont révélées être plus efficaces que les techniques basées sur des règles de linguistique (Banko et Brill, 2001 ; Och et Ney, 2001 ; Fleischman *et al.*, 2003). Dans cette section, l'attention sera exclusivement portée sur les approches statistiques.

Modélisation Statistique du Langage *La Modélisation Statistique du Langage (MSL) tente de détecter les régularités du langage naturel dans le but d'améliorer les performances des différentes applications liées au langage naturel. En général, la Modélisation Statistique du Langage consiste à estimer la distribution de probabilités de différentes unités linguistiques, telles que les mots, les phrases et les documents entiers* (Rosenfeld, 2000).

Un modèle statistique de langage consiste donc simplement en une distribution de probabilités $P(s)$ sur des unités linguistiques s, l'unité linguistique la plus courante étant la phrase. Dans la plupart des modèles statistiques de langage, la probabilité d'une phrase est décomposée en un produit de probabilités conditionnelles :

$$P(s) = P(w_1, ..., w_n) = \prod_i P(w_i \mid h_i)$$

avec w_i le $i^{\text{ème}}$ mot de la phrase s (qui contient n mots) et $h_i = \langle w_1, ..., w_{i-1} \rangle$ l'historique de w_i.

La problématique est alors d'estimer pour chaque mot w_i et chaque historique h_i la probabilité de w_i étant donné h_i, $P(w_i \mid h_i)$. Ces probabilités sont estimées à partir de données d'apprentissage.

Dans cette section, nous nous intéressons à deux techniques classiques de MSL. Ces techniques sont les modèles de n-grammes avec skipping et les triggers.

3.1 Modèles de n-grammes avec skipping

Les modèles de n-grammes représentent la pierre angulaire de la MSL (Rosenfeld, 2000). Un modèle de n-grammes est exactement équivalent à un modèle de Markov d'ordre $n-1$, dont le principe a été présenté dans la section 3.3 du chapitre 1. Plus précisément, un modèle de n-grammes consiste en un ensemble de n-grammes, où chaque n-gramme est une séquence de taille n, équivalente à la concaténation d'un état présent (les $n-1$ éléments précédents) et du dernier élément de l'état futur d'une transition d'un modèle de Markov d'ordre $n-1$. Par conséquent, de même qu'une probabilité est associée à chaque transition entre deux états d'un modèle de Markov, une probabilité conditionnelle est associée à chaque n-gramme d'un modèle de n-grammes.

En pratique, la valeur de n doit être limitée. Généralement, $n = 3$ ou 4, rarement 5, ce qui engendre un problème similaire de couverture.

De nombreuses techniques ont été utilisées pour améliorer l'efficacité des modèles de n-grammes, comme le lissage, le clustering, les mixtures, etc. (Goodman, 2001). Une autre de ces améliorations est le skipping (Huang *et al.*, 1993 ; Ney *et al.*, 1994 ; Rosenfeld, 1994 ; Goodman, 2001). Le skipping consiste simplement à autoriser le saut de mots au sein d'un n-gramme. Par exemple, étant donnée la suite de mots $\langle a, b, x, y, z, c, d \rangle$ et pour $n = 3$, plutôt que de ne considérer que des triplets contigus tels que $\langle a, b, x \rangle$ ou $\langle y, z, c \rangle$ (comme le font les modèles de n-grammes standards), le skipping permet de considérer également des triplets tels que $\langle a, x, d \rangle$, $\langle a, b, c \rangle$ ou $\langle b, c, d \rangle$.

De cette manière, seule une partie de l'historique est utilisé lors de recherches de correspondances avec l'historique : les correspondances sont donc partielles. Ces correspondances partielles permettent une meilleure résistance au bruit et peuvent fournir des recommandations même quand l'historique de l'utilisateur ne correspond pas exactement aux données d'apprentissage. En effet, un des inconvénients des modèles de n-grammes standards est que plus n est grand, plus la probabilité de trouver une correspondance entre le modèle et les données d'application est petite. En revanche, trouver des correspondances partielles est plus probable.

Chaque n-gramme ainsi considéré est alors caractérisé par un ensemble de $n-1$ valeurs de distance entre les éléments qui le composent, que nous appelons configuration de skipping. Afin de combiner les n-grammes issus des différentes configurations de skipping, une interpolation est généralement effectuée. Une interpolation permet en effet d'approcher la probabilité $P(w_i|h_i)$ à partir d'autres probabilités connues. Par exemple, la probabilité d'un mot w_i étant donné un historique h_i peut être donnée par l'équation suivante :

$$\begin{aligned} P(w_i|h_i) &= P(w_i|\langle w_{i-3}, w_{i-2}, w_{i-1}\rangle) \\ &= \alpha P(w_i|\langle w_{i-2}, w_{i-1}\rangle) + \beta P(w_i|\langle w_{w-3}, w_{i-1}\rangle) + \gamma P(w_i|\langle w_{i-3}, w_{i-2}\rangle) \end{aligned} \tag{2.1}$$

où $0 \leqslant \alpha \leqslant 1$, $0 \leqslant \beta \leqslant 1$, $0 \leqslant \gamma \leqslant 1$ et $\alpha + \beta + \gamma = 1$. Utiliser un tel schéma permet donc d'approcher la probabilité d'un quadrigramme (n-gramme avec $n = 4$) en utilisant des sous-modèles de trigrammes (n-grammes avec $n = 3$). De cette façon, plus de correspondances sont trouvées dans l'historique que si un simple modèle de quadrigrammes contigus avait été utilisé. De plus, la complexité en mémoire est réduite, car le nombre de combinaisons d'un modèle de trigrammes est très inférieur au nombre de combinaisons d'un modèle de quadrigrammes. Les modèles de ce type ne fournissent pas d'amélioration significative (Huang *et al.*, 1993), mais représentent une bonne technique en cas de données d'apprentissage de taille limitée (Goodman, 2001).

Enfin, un inconvénient de ce type de modèle est l'augmentation de la complexité en temps au moment de leur utilisation. L'importance de ce problème est proportionnelle à la taille de l'historique considéré : plus le nombre d'éléments pris en compte dans l'historique est grand, plus le nombre de sous-modèles est grand.

3.2 Modèle de triggers

Les modèles de *triggers* (déclencheurs) (Rosenfeld, 1994) consistent en des couples de mots ayant une forte valeur informationnelle, le premier mot étant un déclencheur du second. Les triggers ont été imaginés pour compléter les modèles de n-grammes standards qui ont une portée limitée. Plus précisément, l'observation sur laquelle ils sont basés est que des informations déterminantes se trouvent au-delà des informations que peuvent traiter les modèles de n-grammes [2].

2. La notion de triggers est une forme spécifique de la notion de collocation issue de la linguistique (Polguere, 2003), et utilisée en traitement automatisé du langage naturel (Nerima *et al.*, 2006). En particulier, les méthodes

L'extraction des couples de mots des modèles de triggers est effectuée de façon analogue à l'extraction des n-grammes, en exploitant des données d'apprentissage. Cependant, les mots composant les couples de mots étant distants, il n'est pas envisageable de considérer touts les couples de mots qu'il est possible d'extraire à partir d'un corpus donné. Par conséquent, tout comme pour le skipping, une fenêtre glissante est généralement utilisée. De plus, seuls les couples ayant une grande valeur informationnelle sont retenus dans le modèle. Cette sélection est habituellement effectuée en utilisant l'information mutuelle (Resnik, 1993 ; Zhou et Lua, 1998 ; Chen et Chan, 2003), qui mesure la quantité d'information fournie par un déclencheur A à un mot déclenché B, même si d'autres critères peuvent être utilisés, tels que le $tf{\cdot}idf$ (Troncoso *et al.*, 2004) ou le concept de triggers de « haut niveau » et de « bas niveau » de (Tillmann et Ney, 1996). Dans le cadre de l'étude statistique du langage naturel en général, l'information mutuelle entre deux mots A et B est définie comme suit :

$$MI(A,B) = \frac{p(A,B)}{p(A)\,p(B)} \tag{2.2}$$

En considérant cette écriture, $\frac{p(A,B)}{p(A)\,p(B)}$, on peut faire l'interprétation suivante. La probabilité $p(A,B)$ seule donne une certaine mesure de corrélation entre A et B, mais est biaisée par la fréquence respective de A et B. En effet, si A et B sont tous les deux fréquents, alors il est probable qu'ils apparaissent fréquemment ensemble. La division par $p(A)\,p(B)$, qui correspond à la probabilité que A et B apparaissent ensemble s'ils étaient indépendants, permet donc de corriger ce biais.

Une autre définition de l'information mutuelle utilisée dans le cadre des triggers est directement liée à la théorie de l'information, et est parfois appelée *Average Mutual Information* (information mutuelle moyenne) (Zhou et Lua, 1998) :

$$\begin{aligned} AMI(A,B) = {} & p(A,B)\log\frac{p(B|A)}{p(B)} + p(A,\overline{B})\log\frac{p(\overline{B}|A)}{p(\overline{B})} \\ & + p(\overline{A},B)\log\frac{p(B|\overline{A})}{p(B)} + p(\overline{A,B})\log\frac{p(\overline{B}|\overline{A})}{p(\overline{B})} \end{aligned} \tag{2.3}$$

où A dénote la présence de A, $\overline{A}$ l'absence de A, $P(A)$ la probabilité de voir apparaître A, $p(A,B)$ la probabilité de voir apparaître A et B l'un après l'autre et $p(B|A)$ la probabilité de voir apparaître B quand A apparaît. Nous rappelons qu'une fenêtre glissante est généralement utilisée.

Une fois les triggers sélectionnés, ils sont utilisés pour améliorer la modélisation fournie par les modèles de n-grammes. Plusieurs politiques peuvent alors être utilisées. Le choix de ces politiques porte sur deux aspects : la combinaison des triggers présents dans l'historique, et la combinaison du modèle de triggers avec le modèle de n-grammes. Le traitement des triggers de l'historique peut être effectué en n'utilisant que le déclencheur qui donne la plus grande valeur

d'extraction de triggers sont identiques aux méthodes d'extraction des collocations. C'est dans l'utilisation qui en est faite que les triggers sont spécifiques. En effet, le but des triggers est de fournir de l'information distante pour affiner les distributions des modèles de n-grammes, ce qui constitue une des applications possibles des collocations.

d'information mutuelle, ou selon différentes politique de combinaison (Rosenfeld et Huang, 1992). La façon dont les triggers sont combinés avec le modèle de n-grammes peut se faire en utilisant une interpolation linéaire, en augmentant la probabilité des n-grammes quand un mot est déclenché, etc. (Rosenfeld et Huang, 1992).

Les modèles de triggers sont en général utilisés avec des couples de mots, mais sont facilement généralisables à plus de mots. À notre connaissance, le seul travail dans lequel cela a été expérimenté est (Chen et Chan, 2003). Cependant, cela n'a pas mené à une amélioration statistiquement significative des résultats.

3.3 Similarités entre la navigation Web et le langage naturel

La littérature a montré que la prise en compte de séquences pour la recommandation est particulièrement appropriée pour la navigation Web. Pour cette raison, nous nous sommes intéressés à l'étude de ce domaine d'application particulier. Or, sur de nombreux aspects, la modélisation de la navigation Web est similaire à celle du langage naturel. Cela devient particulièrement apparent quand les caractéristiques des corpus des deux domaines sont mises en avant :

- Dans le cas où l'on ignore leur contenu et qu'on ne les considère que comme des identifiants, les ressources peuvent être considérées comme étant similaires aux mots ;
- Les modèles statistiques de langage utilisent un vocabulaire constitué de mots (*cf.* section précédente) qui peut être considéré comme étant similaire à l'ensemble distinct des ressources du Web ou d'un site Web ;
- Une phrase est une suite de mots, et peut donc être considérée comme étant similaire à une session, qui est une suite de ressources ;
- La présence d'un mot dans une phrase dépend des mots qui le précèdent de la même façon que la consultation d'une ressource dans une session dépend fortement des précédentes consultations de ressources ;
- Dans les deux domaines, il est bien souvent possible de constituer de grands corpus qui peuvent être utilisés pour construire des modèles statistiques ;
- La même hypothèse d'indépendance est utilisable, comme le prouvent les travaux sur les modèles de Markov présentés dans les sections précédentes.

Étant données ces similarités, il semble intéressant d'exploiter les techniques de la MSL pour la recommandation Web. La MSL a été étudiée bien avant la recommandation Web, et beaucoup de modèles qui ont été étudiés ont prouvé leur efficacité, ce qui fournit des perspectives intéressantes. Cependant, trois grandes différences existent entre les deux domaines :

1. Il est possible que plusieurs navigations Web soient imbriquées, ce qui correspondrait à un mélange de phrases et n'existe pas dans le domaine du langage naturel ;
2. Le langage naturel est régi par de fortes contraintes : chaque mot et sa localisation dans la phrase est très important ; la navigation Web est moins contrainte et devrait être traitée avec des modèles plus permissifs.

3. Les vocabulaires utilisés pour le langage naturel ont une taille limitée, bien que déjà suffisamment importante pour être relativement problématique. Les ressources disponibles sur le Web sont quasiment illimitées. Dans le cadre du Web en général, cela est évident. Au sein d'un site Web, le nombre de ressources est variable et peut prendre de très grandes valeurs (*e.g.* Wikipédia).

Par conséquent, les approches de la MSL ne peuvent pas être directement appliquées à la recommandation Web. La première différence, les navigations parallèles, impose un traitement particulier dans lequel il faut pouvoir traiter séparément les ressources qui n'appartiennent pas à la même navigation.

La deuxième différence est également problématique. Plus les données sont contraintes, plus petits seront les modèles statistiques que l'on pourra construire. Par conséquent, le problème de la complexité en mémoire est de plus grande importance dans le domaine de la navigation Web. Ce problème est encore aggravé par la troisième différence, à savoir le plus grand nombre de ressources disponible sur le Web.

Par conséquent, l'exploitation de la MSL pour la navigation Web nécessite des ajustements des algorithmes pour fournir un modèle à la fois léger et permissif.

3.4 Discussion

Nous discutons à présent de la meilleure manière d'ajuster les approches de MSL mentionnées dans ce chapitre aux données de navigation Web. La première question à laquelle répondre est à quel type de données Web de tels modèles peuvent être appliqués. Peuvent-il être appliqués au Web en général, ou doit-on se contenter de ne les utiliser que dans un cadre plus restreint tel que l'espace d'un site Web ?

Ce type de modèle ne semble pas viable pour la quantité astronomique de ressources disponibles sur le Web. En effet, en supposant que le Web ne contienne qu'un billion de ressources (10^{12}), un simple modèle de bigrammes contigus engendrerait 10^{24} combinaisons, ce qui est rédhibitoire étant données les capacités des machines actuelles.

Une possibilité est de regrouper les ressources du Web en sous-groupes homogènes, de façon supervisée ou non supervisée. Dans la suite de cette thèse, on supposera qu'une telle répartition en classes a été effectuée, et que les données auxquelles sont appliquées les modèles sont relativement homogènes.

La seconde question à laquelle répondre est comment adapter les approches de MSL au Web. Rappelons que le but est de trouver un modèle fournissant une grande précision dans les recommandations, une grande couverture et une bonne résistance au bruit tout en ayant une complexité en temps et une complexité en mémoire réduites, ce qui ne peut être fourni en utilisant directement les approches de MSL classiques. Comme mentionné dans le chapitre 1, utiliser des séquences contiguës (*e.g.* des modèles de n-grammes) ne permet pas d'être robuste au bruit. Les données de navigation étant bruitées, le modèle devra donc prendre en compte des séquences non contiguës. C'est déjà ce que font les motifs séquentiels non contigus présentés

dans le chapitre 1. Cependant, ces derniers ont une complexité en mémoire et en temps trop importantes.

Nous abordons à présent les modèles de la MSL dont il est possible de s'inspirer pour améliorer la qualité des recommandation dans le cadre de la navigation Web.

■ Modèle de triggers

Une première possibilité serait d'utiliser les modèles de triggers. Les triggers permettent en effet de considérer des éléments distants, et donc de fournir une certaine résistance au bruit. En transposant les mots à des ressources Web, si une ressource correspond à du bruit, l'impact de toutes les autres ressources à l'intérieur de la fenêtre glissante compensera son impact. En utilisant des valeurs telles que le support ou la confiance ou des probabilités conditionnelles, les ressources les plus fréquentes sont susceptibles d'être recommandées. Or ces ressources ne s'avèrent pas forcément utiles ou intéressantes pour un utilisateur. Par exemple, la page d'accueil est généralement la page la plus visitée d'un site Web, mais pourrait ne pas être la page plus intéressante ou utile pour l'utilisateur. De même, une mauvaise ergonomie dans un site Web pourrait donner lieu à un égarement typique des utilisateurs que l'on pourrait vouloir éviter. De ce point de vue, l'information mutuelle utilisée par les triggers fournit une caractéristique intéressante, car les ressources les plus fréquentes sont moins susceptibles d'avoir une grande valeur d'information mutuelle. Cependant, de même que pour le langage naturel, les triggers ne peuvent pas être utilisés seuls. Par exemple, une ressource rare ayant une grande corrélation avec une ressource de l'historique n'est pas forcément utile pour l'utilisateur.

■ Combinaison d'un modèle de triggers avec un modèle de n-grammes

Il semble donc que les modèles de triggers doivent être combinés avec les modèles de n-grammes, comme cela est le cas dans le domaine de la MSL afin de tirer avantage des deux modèles. Une telle configuration a été expérimentée dans (Pavlov *et al.*, 2004). Les modèles présentés consistent en des interpolations de sous-modèles. En particulier, un modèle de bigrammes (n-grammes avec $n = 2$) est combiné avec un modèle de triggers. Les deux sous-modèles sont ensuite combinés selon une interpolation dont les coefficients sont calculés à l'aide de l'algorithme *Expectation Maximization* (Dempster *et al.*, 1977) sur un corpus de validation. Selon les données considérées, la complexité en temps de cet apprentissage de coefficients pourrait être trop grande. (Pavlov *et al.*, 2004) proposent donc d'utiliser un algorithme de clustering des ressources basé sur les séquences de navigation des utilisateurs. Le modèle permet ainsi de prendre en compte des ressources distantes et de fournir une bonne couverture tout en ayant une faible complexité en temps et en mémoire. Cependant, l'utilisation d'un modèle de bigrammes fournit une moins grande précision dans les recommandations, et si la dernière ressource consultée est du bruit, alors une mauvaise recommandation sera probablement fournie. Dans ce dernier cas de figure, le modèle de triggers utilisé en complément permet d'ignorer cette dernière ressource et d'utiliser les autres ressources pour calculer les recommandations. Cependant, ces recommandations ne seraient basées que sur des valeurs d'information mutuelle, ce qui n'est pas approprié.

■ Skipping

L'alternative que nous proposons est d'utiliser les modèles de n-grammes avec skipping. Ce modèle permet de prendre en compte des ressources distantes, tout en utilisant des probabilités conditionnelles. Cela est très proche des motifs séquentiels non contigus ; la principale différence est que les modèles de n-grammes avec skipping sont utilisés avec une valeur de n fixe, ce qui résulte en une plus faible complexité en mémoire et en temps. La couverture de ces modèles dépend des données considérées. Pour une valeur de n donnée, il est évident qu'un modèle de n-grammes avec skipping fournit une meilleure couverture qu'un modèle de n-grammes standard. Selon les données considérées et la valeur de n, il est cependant possible que la couverture ne soit pas totale, ce qui ne peut être vérifié qu'expérimentalement.

Un modèle de n-grammes avec skipping est utilisé dans (Shani *et al.*, 2005) et a été présenté dans le chapitre 1. Il s'agit du modèle prédictif utilisé pour initialiser le modèle de MDP. Dans ce travail, le skipping est utilisé d'une manière que nous n'avons vu dans aucun autre travail. Au lieu d'utiliser une interpolation de sous-modèles, les occurrences des n-grammes issus des différentes configurations de skipping sont intégrées à une même liste. Pour ce faire, les occurrences des n-grammes sont pondérées en fonction de la distance entre les éléments qui les composent et sont ajoutées au nombre d'occurrences correspondants. La pondération permet de diminuer l'importance accordée aux n-grammes quand les éléments qui les composent sont distants les uns des autres. De cette manière, chaque n-gramme n'est stocké qu'une fois, ce qui a l'avantage de diminuer la complexité en mémoire en comparaison de l'utilisation de l'interpolation.

Par exemple, étant donnée la suite de ressources $\langle a, x, b, c, a, b, y, c\rangle$, il est possible de détecter trois occurrences du trigramme $\langle a, b, c\rangle$. Ces occurrences sont alors stockées dans la même liste de trigrammes. Cependant, une fois stockées dans la liste, il est impossible de déterminer si un n-gramme a souvent été rencontré dans une configuration de skipping plutôt qu'une autre. Cette technique fournit donc une modélisation moins précise.

Ce dernier inconvénient peut tout de même s'avérer être un avantage : il est possible d'utiliser des n-grammes issus d'une configuration de skipping donnée pour chercher une correspondance dans l'historique selon une autre configuration de skipping. Par exemple, étant donné le trigramme $\langle a, b, c\rangle$, il est possible que ce trigramme ait été trouvé plusieurs fois avec un mot indésirable i entre a et b lors de la construction du modèle sur le corpus d'apprentissage ($\langle a, i, b, c\rangle$). Au moment de l'utilisation du modèle obtenu, il est possible que a et b soient rencontrés dans l'historique avant un autre mot indésirable j ($\langle a, b, j\rangle$). Dans ce cas, en utilisant une configuration de skipping différente pendant la phase d'apprentissage et pendant l'utilisation du modèle obtenu, le mot c peut être prédit, ce qui n'est pas possible en utilisant les interpolations. Savoir si cette technique est effectivement plus avantageuse que l'utilisation d'interpolations est une question qui n'a jamais été étudiée à ma connaissance. Cependant, il est certain qu'elle permet une meilleure couverture, et la couverture est un des principaux problèmes de l'utilisation de modèle de n-grammes.

Afin de réduire la complexité en mémoire, (Shani *et al.*, 2005) n'effectuent le skipping que pendant la phase d'apprentissage du modèle, et uniquement entre l'avant-dernière et la dernière ressource des n-grammes. Comme mentionné dans la section 4.3.4 du chapitre 1, ce modèle est

comparé à un des modèles les plus performants de l'état de l'art du FC (un modèle basé sur un réseau de dépendances dans lequel les distributions locales sont des arbres de décision), et fournit de meilleurs résultats. Cependant, les composantes de ce modèle sont assez arbitraires, et de nombreuses améliorations peuvent être apportées, telles que l'utilisation du skipping à la fois à l'apprentissage et pendant les recommandations, l'utilisation d'un schéma de pondération plus précis et l'utilisation d'une variante de skipping mieux appropriée.

4 PROTOCOLE EXPÉRIMENTAL

Dans cette section, nous présentons les corpus de navigation Web qui seront utilisés dans les expérimentations de cette thèse, ainsi que les mesures d'évaluation.

4.1 Corpus

Les études expérimentales de cette thèse sont effectuées sur deux corpus de navigation Web. Le premier a été fourni par le Crédit Agricole S.A. [3]. Il consiste en des logs de navigation sur un intranet par les employés. Cet intranet contient des espaces de travail, des pages d'actualité, des articles, etc. La banque nous a fourni des logs anonymisés contenant 3 391 pages Web distinctes (ressources) parcourues par 815 utilisateurs pendant les années 2007 et 2008. Un corpus de 123 470 consultations de pages a ainsi pu être extrait à partir de ces logs.

Le second corpus est le corpus du serveur Web CTI de l'université DePaul (`http://www.cs.depaul.edu`). Il contient 69 471 consultations de 683 pages par 5 446 utilisateurs sur une période de deux semaines en avril 2002 (*i.e.* environ 170 consultations par jour). Les données fournies ont déjà été pré-traitées et le corpus final ne contient que des sessions de taille supérieure à 1 et contenant des ressources suffisamment fréquentes.

La répartition des tailles de session des deux corpus est présentée dans la Figure 1. Comme on peut le constater, la plupart des sessions ont une taille inférieure à 10. La taille moyenne des sessions du corpus du Crédit Agricole S.A. est de $8,33$ et celle du corpus de l'université DePaul de $5,05$.

Afin d'évaluer expérimentalement la résistance au bruit de notre modèle, nous avons inséré un pourcentage croissant de bruit en extrayant l'ensemble distinct de ressources de chacun des corpus, et en insérant aléatoirement les ressources de ces ensembles dans les corpus correspondants. Bien évidemment, du bruit est déjà contenu dans les corpus initiaux, mais n'est pas quantifiable.

Deux traitements supplémentaires sont effectués sur les données. Le premier est l'élimination des sessions de taille 1 (ce qui était déjà fait pour le corpus de l'université DePaul, mais pas pour le corpus du Crédit Agricole S.A.) et des sessions de taille 2. En effet, dans nos évaluations, nous avons choisi une valeur de $n = 3$, et voulions que les recommandations fournies

3. Données fournies par Jean Philippe Blanchard, responsable du service de veille technologique et stratégique du Crédit Agricole S.A. qui relève du pôle de coordination des systèmes d'information du Crédit Agricole S.A., à qui nous adressons nos remerciements.

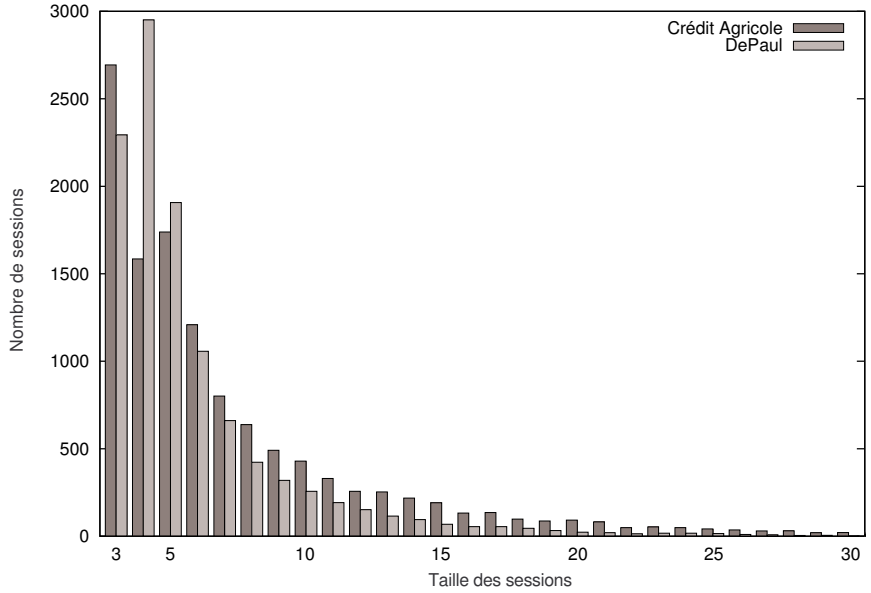

FIGURE 1 – Répartition de la taille des sessions

par les modèles de l'état de l'art portent sur les mêmes historiques. La seconde est la répartition des corpus résultants en un corpus d'entrainement et un corpus de test de 90% et 10% respectivement.

4.2 Mesures d'évaluation

Pour évaluer la précision des recommandations de nos modèles, nous utilisons le *hit ratio* (Jin *et al.*, 2005a ; Pavlov *et al.*, 2004 ; Gündüz et Özsu, 2003). Cette mesure évalue le pourcentage de cas où la ressource observée dans le corpus de test figure dans la liste de recommandation calculée. Cependant, il se peut que la précision soit élevée, mais que la couverture (le nombre de cas où une ressource a pu être recommandée) soit faible. Nous fournissons donc également la couverture, afin d'expliciter le compromis obtenu entre précision et couverture. La taille des listes de recommandation est fixée à 10. Dans certaines expérimentations, le temps d'exécution des algorithmes et les tailles des modèles sont également fournis. Les expérimentations ont toute été effectuées sur une machine ayant un processeur de 2.66 GHz et 4 Go de mémoire. Les temps d'exécution ont été obtenus en faisant fonctionner dix fois chaque modèle, et en retenant le temps de calcul le plus petit.

5 CONCLUSION

La navigation Web représente un type de données sur lequel les approches de recommandation prenant en compte la séquentialité semblent particulièrement efficaces. L'état de l'art de

la recommandation présenté dans ce chapitre confirme cette impression, et met en avant des problématiques spécifiques de ce type de données. La première de ces problématiques est la recherche d'un compromis entre précision, couverture et complexité en temps et en mémoire induite par la navigation Web ; la deuxième est la prise en compte du bruit et des navigations parallèles.

Pour répondre à ces problématiques, nous proposons de s'inspirer de la modélisation statistique du langage (MSL), qui a de nombreuses similarités avec la modélisation prédictive du Web. Nous avons présenté dans ce chapitre un état de l'art de la MSL qui a mis en avant une amélioration des modèles de n-grammes appelée skipping et qui a des caractéristiques intéressantes du point de vue de notre problématique. Nous proposons donc dans le chapitre suivant un modèle inspiré du principe du skipping et adapté aux contraintes spécifiques de la navigation Web.

Chapitre 3

Nouveau modèle inspiré de la modélisation statistique du langage

Comme nous l'avons montré dans le chapitre précédent, prédire le comportement des utilisateurs sur le Web implique un compromis entre complexité, précision, et couverture. Les motifs séquentiels permettent de manipuler des séquences non contiguës, mais la complexité est alors trop grande. Les all-k^{th}-order Markov models, quant à eux, induisent une faible complexité en temps, ont une couverture totale, mais leur complexité en mémoire reste élevée et ils ne permettent de prendre en compte ni le bruit contenu dans les navigations ni les navigations parallèles. Dans le domaine de la modélisation statistique du langage (MSL), les modèles de n-grammes avec skipping, bien que n'ayant qu'une efficacité relative pour le langage naturel, possèdent des caractéristiques intéressantes qui appliquées au domaine de la navigation Web permettent d'allier faible complexité en temps et en mémoire, prise en compte du bruit et des navigations parallèles, et bonne couverture.

Par conséquent, nous proposons dans ce chapitre un algorithme basé sur ce type de modèle. Ce modèle est appelé Skipping-Based Recommender (SBR) et est présenté dans un premier temps. Nous présentons ensuite une comparaison théorique et expérimentale du modèle SBR aux modèles de l'état de l'art.

1 LE SKIPPING-BASED RECOMMENDER (SBR)

Dans cette section, nous présentons le modèle SBR. Ce modèle a les caractéristiques suivantes :

- Les occurrences des n-grammes sont fusionnées comme dans (Shani *et al.*, 2005) ;
- Une fenêtre glissante est utilisée pour réduire la complexité en temps et en mémoire, comme pour les motifs séquentiels (Nakagawa et Mobasher, 2003b) ;
- Plusieurs variantes de skipping peuvent être utilisées ;
- Plusieurs schémas de pondération comme dans (Shani *et al.*, 2005) peuvent être utilisés afin d'accorder moins d'importance aux ressources les plus distantes.

Nous présentons d'abord trois variantes de skipping et quatre schémas de pondérations qui peuvent être utilisés avec le modèle SBR, puis nous présentons le fonctionnement général du modèle.

1.1 Variantes de Skipping

Une variante de skipping correspond à des contraintes sur les distances entre les éléments des n-grammes. Une telle contrainte peut par exemple consister à limiter la taille de la distance entre les deux premiers éléments à 0. Dans les variantes de skipping présentées dans cette section, nous considérons que quand le skipping est appliqué, sa taille est limitée à la taille de la fenêtre glissante utilisée.

1.1.1 Skipping de Shani

Une première variante de skipping possible est celle utilisée pour l'apprentissage du modèle prédictif de (Shani *et al.*, 2005). Dans cette variante, le skipping est autorisé entre l'avant dernier et le dernier élément du n-gramme ; tous les autres éléments étant contigus.

Par exemple, pour $n = 3$, et la séquence de navigation (a, b, x, y, z, c, d) où (a, b, c, d) et (x, y, z) correspondent à deux navigations imbriquées. Cette variante autorise la considération de triplets tels que (a, b, y) ou (a, b, c) en plus des triplets contigus tels que (a, b, x) et (z, c, d). Cette variante permet donc de prendre en compte des éléments distants d'une séquence si le dernier élément correspond à la continuation d'une navigation commencée auparavant comme pour le triplet (a, b, c). Les éléments entre (a, b) et c sont ici considérés comme des éléments d'une autre navigation, mais pourraient aussi être du bruit.

Cependant, cette variante de skipping ne permet pas de prendre en compte des imbrications de navigations si les deux derniers éléments correspondent à la continuation d'une navigation commencée auparavant (une étape après la configuration précédente) : par exemple, le triplet (b, c, d) ne peut être pris en compte puisque b et c ne sont pas contigus.

1.1.2 Skipping complet

Cette variante est la variante standard, qui consiste à ne pas fixer de contrainte sur les distances entre les éléments des n-grammes. Le modèle obtenu est alors comparable à un modèle de motifs séquentiels non contigus comme par exemple celui de (Nakagawa et Mobasher, 2003b). La principale différence est que le SBR considère uniquement des séquences de taille n alors que les motifs séquentiels manipulent généralement des séquences de taille variable. Cette variante a plusieurs avantages. Premièrement, un grand nombre de n-grammes est engendré, ce qui permet une plus grande couverture. Deuxièmement, cette variante de skipping est capable de prendre en compte les navigations parallèles, le bruit et les navigations plus approximatives, où qu'elles se trouvent dans la séquence, et quelle que soit leur taille. Cependant, cette variante pourrait être trop permissive, ce qui peut amoindrir la qualité des prédictions et engendre une plus grande complexité en temps et en mémoire en comparaison avec la variante de Shani.

1.1.3 Skipping multinavigationnel

Les deux variantes précédentes sont des variantes de l'état de l'art qui ont toutes deux leurs inconvénients. La première ne permet pas de prendre en compte le bruit et les navigations parallèles, alors que la seconde est trop permissive et a une plus grande combinatoire. Nous proposons donc une nouvelle variante qui possède les avantages de ces deux variantes, sans en avoir les inconvénients.

Cette variante autorise le skipping pour le premier ou le dernier élément du n-gramme, ce qui permet d'ignorer le bruit dans la première partie du n-gramme ou dans la dernière, mais pas dans les deux. L'hypothèse sur laquelle elle est basée est que les navigations parallèles et les erreurs de navigations sont peu fréquentes et que la plupart des ressources d'une même navigation sont en général contiguës.

Par exemple, étant donné l'exemple précédent, il devient possible de prendre en compte les deux configurations (a, b, c) et (b, c, d) mais pas (a, x, c).

1.2 Schémas de pondération

Ce paramètre du modèle SBR est fondé sur deux éléments. Tout d'abord, nous faisons l'hypothèse suivante :

> *Plus la distance qui sépare les éléments d'un n-gramme est grande, plus la probabilité que ce n-gramme soit représentatif du comportement de l'utilisateur est faible.*

En outre, comme le modèle SBR utilise une fenêtre glissante, les ressources sont soudainement ignorées dès qu'elles se retrouvent en dehors de cette fenêtre, ce qui est trop brutal. Pour ces deux raisons, nous proposons de pondérer les occurrences des n-grammes en fonction de la distance entre les éléments qui les composent. Ainsi, les n-grammes ayant les plus grandes distances entre les éléments qui les composent auront une influence amoindrie, qui diminuera progressivement au fur et à mesure que la fenêtre sera déplacée. Nous présentons dans cette section plusieurs façons de pondérer, appelées schémas de pondération qui peuvent être utilisés pour prendre en compte ces n-grammes.

On notera d_i la distance entre le $i^{\text{ème}}$ élément du n-gramme et le dernier élément du n-gramme et D la taille de la fenêtre.

Afin de montrer l'apport de la pondération des n-grammes issus du skipping, nous proposons tout d'abord de ne pas utiliser de pondération, comme cela est communément effectué pour les modèles de motifs séquentiels non contigus par exemple (Nakagawa et Mobasher, 2003b), ou encore pour les modèles de triggers dans le domaine des modèles de langage. Quelle que soit la distance entre les éléments qui les composent dans la limite de le fenêtre glissante, les n-grammes ont tous le même poids. Dans ce cas de figure, le schéma de pondération est

appelé le schéma **sans poids**. Le poids $w(d_1, ..., d_{n-1})$ de l'occurrence d'un n-gramme issu du skipping est alors calculé selon l'équation suivante :

$$w(d_1, ..., d_{n-1}) = \begin{cases} 1 & \text{si } d_1 \leqslant D \\ 0 & \text{sinon} \end{cases} \tag{3.1}$$

Les performances obtenues seront considérées comme les performances de référence.

Le second schéma que nous proposons consiste à appliquer une fonction de décroissance linéaire inversement proportionnelle à la taille de la fenêtre D en fonction de la distance d_1 entre le premier élément du n-gramme et du dernier élément du n-gramme. Dans ce cas de figure, le schéma de pondération est appelé le schéma de **décroissance linéaire**. Le poids devient alors :

$$w(d_1, ..., d_{n-1}) = \begin{cases} -\frac{d_1}{D} + 1 & \text{si } d_1 \leqslant D \\ 0 & \text{sinon} \end{cases} \tag{3.2}$$

Une autre façon d'effectuer cette pondération est d'appliquer une décroissance exponentielle comme dans (Shani *et al.*, 2005). En utilisant un tel schéma de pondération, la valeur diminue plus rapidement. Dans ce cas de figure, le schéma de pondération est appelé le schéma de **décroissance exponentielle simple**, et est défini comme suit :

$$w(d_1, ..., d_{n-1}) = \begin{cases} 2^{-d_1} & \text{si } d_1 \leqslant D \\ 0 & \text{sinon} \end{cases} \tag{3.3}$$

Ces trois schémas sont adaptés pour la variante du skipping de Shani. En effet, comme dans cette variante le skipping n'est autorisé que pour la dernière ressource, il n'est pas nécessaire de considérer les distances entre tous les éléments des n-grammes. Dans les deux autres variantes de skipping en revanche, le skipping peut être effectué entre d'autres ressources, et les différentes distances entre les éléments devraient être considérées pour calculer les poids. Nous proposons donc un nouveau schéma de pondération qui dépend de la distance entre tous les éléments des n-grammes.

Nous proposons donc de pondérer les occurrences selon une moyenne de valeurs exponentielles dépendantes des distances entre tous les éléments des n-grammes. Dans ce cas de figure, le schéma de pondération est appelé le schéma de **décroissance exponentielle multiple**. Le poids final est calculé selon l'équation suivante :

$$w(d_1, ..., d_{n-1}) = \begin{cases} \frac{1}{n-1} \sum_{i=1}^{n-1} 2^{-d_i} & \text{si } d_1 \leqslant D \\ 0 & \text{sinon} \end{cases} \tag{3.4}$$

Par exemple, étant donnée la séquence $\langle a, b, x, y, z, c, d \rangle$ et $n = 3$, les triplets $\langle a, b, d \rangle$ et $\langle a, c, d \rangle$ ne devraient pas avoir le même poids, même si le premier élément qui les compose est équidistant du dernier élément du triplet. Plus précisément, le poids de $\langle a, c, d \rangle$ devrait être plus grand que celui de $\langle a, b, d \rangle$ puisque la ressource du milieu du premier triplet (c) est plus proche de d que celle du second triplet (b), ce qui est le cas lorsque cette variante est utilisée.

1.3 Fonctionnement du SBR

Le SBR fonctionne en 3 étapes, similaires aux étapes de construction d'un modèle statistique de langage, ainsi qu'aux étapes de construction d'un modèle de recommandation basé sur la détection de motifs.

1. Apprentissage des occurrences de n-grammes sur un corpus d'apprentissage en fonction de la variante de skipping et du schéma de pondération choisis ;
2. Estimation des probabilités conditionnelles à partir des occurrences obtenues à l'étape précédente. Le résultat est le modèle SBR ;
3. Utilisation du modèle résultant pour recommander des ressources en fonction de la session active d'un utilisateur donné, de la variante de skipping et du schéma de pondération choisis.

Notation

On notera R l'ensemble des ressources, avec $|R| = N$. Le corpus est constitué d'un ensemble de sessions S. Chaque session s contient une suite de ressources $\langle r_1, ..., r_{|s|} \rangle$ dont chaque élément $r_i \in R$. Le nombre d'occurrences d'un n-gramme $\langle r_1, ..., r_n \rangle$ est dénoté par $C(\langle r_1, ..., r_n \rangle)$.

1.3.1 Apprentissage des occurrences

Pendant la construction du modèle, les n-grammes et leurs occurrences sont extraits du corpus d'apprentissage. Les occurrences sont pondérées selon le schéma de pondération choisi et sont ajoutées à une même liste d'occurrences de n-grammes, comme cela est fait dans (Shani *et al.*, 2005). L'algorithme 1 présente la façon dont cette étape est effectuée en utilisant la variante du skipping complet et $n = 3$.

1.3.2 Calcul des probabilités conditionnelles des n-grammes

Étant données les occurrences de l'étape précédente, les probabilités conditionnelles doivent être calculées. Soit le n-gramme $\langle r_1, ..., r_n \rangle$. La probabilité conditionnelle de la ressource r_n sachant $\langle r_1, ..., r_{n-1} \rangle$ est calculée de la façon suivante :

$$P(r_n \mid \langle r_1, ..., r_{n-1} \rangle) = \frac{C(\langle r_1, ..., r_n \rangle)}{C(\langle r_1, ..., r_{n-1} \rangle)} \tag{3.5}$$

Le nombre d'occurrences du $(n-1)$-gramme $\langle r_1, ..., r_{n-1} \rangle$ peut être obtenu en faisant la somme du nombre d'occurrences des n-grammes $\langle r_1, ..., r_{n-1}, \rho \rangle$ pour chaque ressource $\rho \in R$.

Données : Un ensemble S de sessions de navigation
Résultat : Une liste de trigrammes avec leurs occurrences
$trigramlist \leftarrow \langle\rangle$;
pour chaque session s dans S **faire**
  pour $i \leftarrow 1$ à $\|s\| - 2$ **faire**
    pour $j \leftarrow i + 1$ à $\min(i + D, \|s\| - 1)$ **faire**
      pour $k \leftarrow j + 1$ à $\min(j + 1 + D, \|s\|)$ **faire**
        $trigram \leftarrow (r_i, r_j, r_k)$;
        $d_1 \leftarrow k - i - 1$;
        $d_2 \leftarrow k - j - 1$;
        si *trigram* est dans *trigramlist* **alors**
          $C(r_i, r_j, r_k) \leftarrow C(r_i, r_j, r_k) + w(d_1, d_2)$;
        sinon
          $C(r_i, r_j, r_k) \leftarrow w(d_1, d_2)$;
        fin
      fin
    fin
  fin
fin

Algorithme 1 : Calcul du nombre d'occurrences des trigrammes issus du skipping en utilisant la variante du skipping complet.

1.3.3 Recommandation

L'étape de recommandation consiste à prédire la prochaine ressource que l'utilisateur consultera étant données les $D - 1$ ressources précédentes dans la session $\langle r_{i-D+1}, ..., r_{i-1} \rangle$. Pour chaque ressource ρ dans l'ensemble des ressources R, le score q est calculé en fonction de chaque état possible σ. σ correspond aux sous-historiques ayant des correspondances avec les états du modèle. Le score correspond à la probabilité pondérée qu'au moins un des états de l'historique donnera la ressource ρ. Il est calculé selon la formule suivante :

$$q(\rho, h) = 1 - \prod_{\sigma} (1 - P(\rho \mid \sigma) \cdot w(d_1, ..., d_{n-1})) \tag{3.6}$$

où $P(\rho|\sigma)$ est la probabilité de ρ sachant l'état σ, apprise lors de la seconde étape. Les états σ considérés dépendent de la variante de skipping choisie. Par exemple, si un utilisateur a consulté les ressources suivantes `A B C D E F G H I J K`, alors si la taille de la fenêtre est $D = 5$, les ressources considérées pour calculer les recommandations sont les suivantes : `H`, `I`, `J` et `K` (les 4 ressources précédentes). Si la variante de skipping utilisée est le skipping multinavigationnel et $n = 3$, alors des états de taille 2 doivent être considérés, donc $1 + 2 \times 2 = 5$ états. Ces états sont présentés dans la Figure 1.

Session	A	B	C	D	E	F	G	H	I	J	K
États										J	K
									I		K
									I	J	
								H			K
								H	I		

FIGURE 1 – Exemple d'états obtenus pour la variante du skipping multinavigationnel et $n = 3$.

Ensuite les n-grammes correspondants sont recherchés dans le modèle et les entrées correspondantes sont incluses dans la liste de recommandation en utilisant l'équation (3.6).

Il est important de constater qu'en utilisant un all-k^{th}-order Markov model, quand aucune correspondance entre un sous-historique et le modèle n'est trouvée, la taille du sous-historique est réduite, et la ressource la plus distante est ignorée. Or, il est possible que cette ressource soit plus pertinente que certaines ressources plus proches de l'historique, en particulier si ces ressources correspondent à du bruit. Le SBR présente donc l'avantage d'offrir la possibilité d'accorder moins d'importance ou d'ignorer certaines ressources proches qui sont moins pertinentes, et ainsi de mettre en valeur certaines ressources éloignées qui sont plus utiles pour calculer les recommandations.

2 COMPARAISON DU MODÈLE SBR À L'ÉTAT DE L'ART

Dans cette section, le modèle SBR est comparé à deux modèles de l'état de l'art mentionnés dans le chapitre 2, à savoir un all-k^{th}-order Markov model et un modèle de motifs séquentiels non contigus. Dans une première partie, la comparaison porte sur la complexité en temps et en mémoire des modèles. La seconde partie est consacrée à la résistance au bruit.

2.1 Complexité en temps et en mémoire

Cette section est dédiée à l'étude théorique et expérimentale du modèle SBR que nous proposons, du all-k^{th}-order Markov model et du modèle de motifs séquentiels non contigus. Deux aspects sont considérés : la complexité en temps dans le pire des cas et la complexité en mémoire dans le pire des cas. Comme l'apprentissage peut être effectué hors-ligne, la complexité en temps n'est étudiée que pour la phase de recommandation.

2.1.1 Étude théorique

Théoriquement, la complexité en mémoire dépend du nombre d'éléments distincts N des données considérées. Par exemple, si un modèle doit stocker toutes les séquences de taille 1 et 2, alors le nombre d'éléments à stocker est $N + N^2$.

La complexité en temps dépend du nombre de sous-historiques considérés pour chaque recommandation, et du temps nécessaire pour trouver une correspondance avec les états du modèle.

a) Motifs séquentiels non contigus

■ **Complexité en mémoire :**

Les motifs séquentiels non contigus induisent un très grand nombre de séquences à stocker. Soit D la taille de la fenêtre, la borne supérieure du nombre d'éléments à stocker est :

$$\sum_{k=1}^{D} N^k = N \cdot \frac{1 - N^{D+1}}{1 - N} = \mathcal{O}(N^D) \tag{3.7}$$

En utilisant un algorithme du type GSP (cf. section 3.3.2 du chapitre 1) la complexité en mémoire peut être réduite ; cependant, cela réduit également la couverture des séquences les plus longues.

■ **Complexité en temps :**

Les modèles de motifs séquentiels non contigus considèrent des séquences de taille variable à l'intérieur d'une fenêtre de taille D. Le dernier élément du motif (le conséquent) est toujours l'élément le plus à droite dans la fenêtre. Le nombre de combinaisons induit est donc :

$$\sum_{k=1}^{D-1} C_{D-1}^{k} = 2^{D-1} - 1 \tag{3.8}$$

La recherche des motifs correspondants dans le modèle peut être effectuée en $\mathcal{O}(k)$ en utilisant une structure arborescente, où k est la longueur du motif pour lequel une correspondance est recherchée. La complexité en temps est donc :

$$\begin{aligned} \sum_{k=1}^{D-1} \mathcal{O}(k) \cdot C_{D-1}^{k} &\leqslant \sum_{k=1}^{D-1} \mathcal{O}(D-1) \cdot C_{D-1}^{k} \\ &\leqslant \mathcal{O}(D-1) \sum_{k=1}^{D-1} C_{D-1}^{k} \\ &\leqslant \mathcal{O}(D-1) \cdot (2^{D-1} - 1) = \mathcal{O}(D \cdot 2^D) \end{aligned} \tag{3.9}$$

b) All-k^{th}-order Markov models

■ **Complexité en mémoire :**

Le nombre maximal d'éléments induits par un all-k^{th}-order Markov model est le même que celui des motifs séquentiels. Cependant, en pratique, considérer des éléments contigus induit beaucoup moins d'éléments, et la complexité en mémoire est très inférieure. La différence dépend de la taille des données d'apprentissage et du nombre distinct de ressources. Comme pour les motifs séquentiels, supprimer les séquences les moins fréquentes peut réduire la complexité en mémoire, comme cela est fait dans (Deshpande et Karypis, 2004), mais peut aussi induire une plus petite couverture des séquences les plus longues.

■ **Complexité en temps :**

Les all-k^{th}-order Markov models ont une complexité en temps inférieure à celle des motifs séquentiels non contigus. En effet, étant donné que les motifs considérés sont contigus, seulement $D-1$ séquences sont induites pour chaque recommandation. La complexité en temps est donc :

$$\sum_{k=1}^{D-1} \mathcal{O}(k) = \mathcal{O}(D^2) \tag{3.10}$$

c) Modèle SBR

■ **Complexité en mémoire :**

La borne supérieure du nombre d'éléments à stocker en utilisant le SBR est N^n où $n \leqslant D$, ce qui représente une borne très inférieure aux deux modèles précédents.

■ **Complexité en temps :**

La complexité du modèle SBR dépend de la variante de skipping avec laquelle il est utilisé. En utilisant la variante du skipping complet, la variante ayant la plus grande complexité, C_{D-1}^{n} séquences sont induites à chaque recommandation. Comme la recherche d'une correspondance peut être faite en $\mathcal{O}(n)$ en utilisant une structure arborescente, la complexité en temps est :

$$\begin{array}{ll} \mathcal{O}(n \cdot D^{n-1}) & \text{si } n \leqslant \frac{D}{2} \\ \mathcal{O}(n \cdot D^{D-n+1}) & \text{si } \frac{D}{2} \leqslant n \leqslant D \\ \mathcal{O}(n \cdot D^{D/2}) & \text{pour } n \leqslant D \end{array} \tag{3.11}$$

En utilisant la variante du skipping de Shani ou du skipping multinavigationnel, la complexité n'est plus que de $\mathcal{O}(n \cdot D)$.

Ainsi, selon la valeur de n, la variante du skipping complet peut avoir une grande complexité en temps. Cependant, en utilisant de petites valeurs de n, telles que 3 ou 4, une complexité acceptable est fournie. Pour $n = 3$ et $D \geqslant 6$, la complexité en temps est $\mathcal{O}(D^2)$. Pour $n = 4$ et $D \geqslant 8$, la complexité en temps est $\mathcal{O}(D^3)$.

2.1.2 Étude expérimentale

Dans cette section, les résultats expérimentaux fournis par les trois modèles sont comparés en termes de taille de modèle et de temps d'exécution. Afin de rendre les modèles comparables, tous les seuils de supports et de confiance sont fixés à 0. La taille des listes de recommandation est fixée à 10. Nous avons choisi cette valeur pour deux raisons : (1) Un utilisateur prend rarement en considération les ressources recommandées au-delà de cette limite (2) les listes de recommandation de taille 10 sont très répandues dans l'état de l'art ce qui permet une comparaison directe des résultats. La taille de la fenêtre est $D = 10$. Le modèle SBR est utilisé avec une valeur de $n = 3$ et avec les trois variantes de skipping : le skipping de Shani, le skipping multinavigationnel et le skipping complet. Les résultats sont présentés dans le tableau 1 pour les corpus de l'université DePaul et du Crédit Agricole S.A.

	Crédit Agricole S.A.		DePaul	
	taille	temps	taille	temps
SBR + Shani	2,3 Mo	3mn05s	1,5 Mo	14s
SBR + Multinavigationnel	4,1 Mo	3mn20s	2,6 Mo	18s
SBR + Complet	8,1 Mo	5mn51s	5,3 Mo	25s
all-k^{th}-order Markov model	8,3 Mo	17mn02s	3,3 Mo	3mn06s
Motifs séquentiels	289,6 Mo	10mn50s	108,7 Mo	1mn08s

TABLE 1 – Taille et temps d'exécution des modèles.

■ Taille des modèles

Le tableau 1 montre que le modèle SBR avec les variantes des skippings de Shani et multinavigationnel fournit les modèles les plus petits sur les deux corpus. Sur le corpus de l'université DePaul, la variante du skipping complet induit un plus grand modèle que le all-k^{th}-order Markov model. Cependant, sur le corpus du Crédit Agricole S.A. la taille du modèle SBR avec la variante du skipping complet est légèrement inférieure à celle du all-k^{th}-order Markov model.

La grande complexité en mémoire des motifs séquentiels est clairement confirmée : le modèle obtenu est plus de 20 fois plus grand que les autres modèles sur le corpus du Crédit Agricole S.A., et plus de 30 fois sur le corpus de l'université DePaul. Arrivé à ce point, le modèle SBR et le all-k^{th}-order Markov model sont donc presque équivalents.

Comme prévu, la variante du skipping multinavigationnel engendre une taille de modèle supérieure à celle de la variante du skipping de Shani et inférieure à celle du skipping complet. Enfin, sur les deux corpus, cette taille est inférieure à celles des deux modèles de l'état de l'art.

- **Temps d'exécution**

Étonnamment, les motifs séquentiels ont un temps d'exécution inférieur à celui du all-k^{th}-order Markov model : il est 2,7 fois plus rapide sur le premier corpus, et 1,6 fois sur le second. Cela est dû à deux éléments. Le premier est que le modèle de motifs séquentiels considère des séquences non contiguës et contient beaucoup plus d'éléments. En effet, le modèle de motifs séquentiels occupe 33 fois plus de mémoire que le all-k^{th}-order Markov model sur le premier corpus et 35 fois plus de mémoire sur le second. Ainsi, en utilisant les motifs séquentiels, il est beaucoup plus fréquent de trouver des correspondances, et donc de trouver des ressources à recommander, et des listes peuvent donc être construites plus rapidement. Le second élément est l'utilisation de la politique de la longueur maximale. En utilisant cette politique, si un nombre de ressources à recommander suffisant est obtenu, il n'est pas nécessaire d'inspecter les séquences de taille inférieure. Comme il est plus difficile de trouver des correspondances avec les longues séquences contiguës qu'avec de longues séquences non contiguës, le processus du modèle de motifs séquentiel est plus court.

Le temps d'exécution du modèle SBR est clairement inférieur à celui des deux autres modèles. En utilisant la variante du skipping complet, qui est celle qui a le plus grand temps d'exécution des trois variantes de skipping, ce temps est quatre fois inférieur à celui du modèle de motifs séquentiels sur le corpus de l'université DePaul, et presque deux fois inférieur sur le corpus du Crédit Agricole S.A. Le SBR représente donc l'alternative la plus efficace en terme de temps de calcul. Enfin, comme pour la taille des modèles, la variante du skipping multinavigationnel a un temps de calcul plus grand que celui de la variante du skipping de Shani et inférieur à celui du skipping complet.

2.2 Précision et résistance au bruit

Comme mentionné précédemment, la présence de bruit dans les navigations peut avoir des effets désastreux sur les recommandations. Notre modèle est conçu dans l'optique d'une bonne résistance au bruit. Dans cette section, nous le comparons à l'état de l'art en termes de précision et de résistance au bruit.

2.2.1 Étude théorique

Les all-k^{th}-order Markov models sont souvent considérés comme figurant parmi les modèles les plus performants de l'état de l'art. Cependant, l'utilisation de séquences contiguës ne permet pas d'ignorer les ressources correspondant à du bruit. Quand, pour une taille de séquence donnée, aucune correspondance n'est trouvée entre l'historique et les séquences du modèle, la taille des séquences considérées est réduite pas à pas, jusqu'à ce qu'une ressource

puisse être recommandée. Après une réduction, la ressource qui est ignorée est celle qui est la plus éloignée. Donc si du bruit est apparu dans un passé récent, et qu'aucune correspondance n'a été trouvée si ce bruit n'est pas ignoré, la taille des séquences considérées sera réduite jusqu'à ce que le bruit ne figure plus dans l'historique considéré. Par conséquent, peu de ressources sont considérées pour calculer la recommandation. De plus, quand la dernière ressource consultée est du bruit, aucune information fiable n'est utilisée pour fournir la recommandation. Pour ces raisons, nous pensons que les séquences contiguës, et en particulier les all-k^{th}-order Markov models, ne représentent pas la meilleure configuration.

Les motifs séquentiels non contigus possèdent de bonnes caractéristiques qui les rendent plus robustes à ce genre de problèmes. Comme toutes les $2^{D-1} - 1$ séquences possibles dans l'historique sont considérées, si du bruit est apparu dans un passé récent, des séquences plus longues qui n'incluent pas ce bruit pourront être utilisées pour calculer les recommandations. Cependant, la plupart des séquences considérées dans ce cas de figure sont construites à partir de ressources distantes les unes des autres, et nous pensons que de telles séquences pourraient être moins représentatives. En effet, nous faisons l'hypothèse suivante :

> *Les navigations comportant des erreurs, des retours vers les pages précédentes, etc. entre chaque page sont relativement rares et la plupart des consultations contiguës correspondent à des transitions cohérentes.*

En accord avec cette hypothèse, le modèle SBR possède plusieurs avantages en ce qui concerne la résistance au bruit. Le premier avantage est le même que celui des motifs séquentiels, à savoir le fait que plusieurs configurations de skipping portant sur plusieurs parties de l'historique sont combinées et pondérées pour calculer les recommandations. Le second avantage est qu'en utilisant les variantes du skipping de Shani ou du skipping multinavigationnel, parmi les n éléments de chaque n-gramme, $n - 1$ éléments sont toujours contigus, ce qui amoindrit le problème lié au phénomène des transitions incohérentes. Cependant, pour la variante du skipping de Shani, le problème inverse peut se poser. La variante du skipping multinavigationnel a quant à lui des caractéristiques lui permettant une certaine résistance au bruit et une certaine cohérence dans les transitions utilisées pour calculer les recommandations. Enfin, quelle que soit la configuration de skipping utilisée, le modèle a une complexité moindre que celle des motifs séquentiels. Il semble donc représenter la meilleure alternative.

2.2.2 Étude expérimentale

Nous nous intéressons à présent à l'étude expérimentale de la résistance au bruit des trois modèles. Les expérimentations sont effectuées sur des versions des corpus du Crédit Agricole S.A. et de l'université DePaul dans lesquels 0%, 15% et 30% de bruit a été inséré. Il est important de constater que quand aucun bruit n'a été inséré, les corpus en contiennent tout de même. Par conséquent, la valeur de 0% de bruit ne signifie pas qu'il n'y a pas de bruit dans le corpus, mais qu'aucun bruit supplémentaire n'a été ajouté. Pour cette raison, nous n'avons pas inséré plus de 30% de bruit.

Dans cette section, nous commençons par rechercher quelle configuration du SBR est la plus appropriée. Ensuite nous nous concentrons sur la comparaison du modèle SBR dans cette configuration aux deux modèles de l'état de l'art. Ici encore, la taille des listes de recommandation est fixée à 10. Les résultats sont fournis en termes de hit ratio et de couverture.

a) Paramétrage du SBR

Cette section est consacrée à l'étude expérimentale du modèle SBR en fonction de ses paramètres : variantes de skipping et schémas de pondération. Ces expérimentations sont effectuées sur les versions non bruitées des corpus du Crédit Agricole S.A. et de l'université DePaul, dans lesquelles aucun bruit supplémentaire n'a été inséré. En effet, le but ici est de déterminer quelle configuration est la plus performante, indépendamment de la résistance au bruit.

Les paramètres étudiés ici sont les suivants :

- Effets de l'utilisation du skipping uniquement à l'apprentissage, ou à la fois à l'apprentissage et pendant les recommandations ;
- Taille de la fenêtre glissante ;
- Variantes de skipping ;
- Schémas de pondération.

■ Skipping effectué uniquement à l'apprentissage

Dans un premier temps, nous nous intéresserons aux résultats obtenus lorsque le skipping n'est utilisé qu'à l'apprentissage. La figure 2 et la figure 3 montrent les hit ratio obtenus respectivement sur le corpus du Crédit Agricole S.A. et celui de l'université DePaul, et les tableaux 2 et 3 les couvertures respectives. Nous rappelons que la taille des listes de recommandation est fixée à 10.

N'utiliser le skipping qu'à l'apprentissage permet une comparaison du SBR au modèle prédictif utilisé dans (Shani *et al.*, 2005). Dans ce cas, le skipping permet d'augmenter le nombre de n-grammes du modèle et donc d'augmenter la couverture. La précision peut également être augmentée, car des n-grammes non contigus, correspondant à des navigations entre les éléments desquelles du bruit a pu apparaître, pourront être inclus dans le modèle, le rendant ainsi plus représentatif.

Comme on peut le constater sur les figures 2 et 3, augmenter la taille de la fenêtre fournit une amélioration en termes de hit ratio. Sur le corpus du Crédit Agricole S.A., elle passe de $68,2$ pour une taille de fenêtre de 3 (modèle de trigrammes standard) à $70,8$ pour une fenêtre de 10 avec la variante du skipping complet et le schéma de pondération selon la décroissance exponentielle multiple, soit une amélioration de $3,8\%$. Sur le corpus de l'université DePaul, elle passe de $57,6$ pour une taille de fenêtre de 3 à $59,6$ pour une fenêtre de 10, soit une amélioration de $3,5\%$. Une détérioration de la précision par rapport au modèle de trigrammes standard peut même être constatée quand aucune pondération n'est utilisée. Par conséquent, utiliser le

skipping uniquement à l'apprentissage permet une certaine amélioration de la précision des résultats. En se basant sur ces seuls histogrammes, aucune configuration ne semble véritablement meilleure qu'une autre. Il est important de constater que la complexité en temps est en $\mathcal{O}(1)$ pendant la recommandation (puisque le skipping n'y est pas utilisé), et que par conséquent ces améliorations ne se font pas au prix d'une augmentation du temps de calcul.

Si on se tourne vers les les tableaux 2 et 3, on peut constater que, alors que l'on aurait pu s'attendre à une augmentation, même minimale, l'utilisation de la variante du skipping de Shani n'augmente pas la couverture, quelle que soit la taille de la fenêtre utilisée. En revanche, les deux autres variantes fournissent une meilleure couverture au fur et à mesure que la taille de la fenêtre augmente. Sur le premier corpus, 4,7% de couverture supplémentaires sont obtenus ; sur le second, 6% de couverture supplémentaires sont obtenus. Bien que ces couvertures ne soient pas totales, elles atteignent tout de même 89,2% et 90,9% respectivement, alors que le modèle a une complexité en $\mathcal{O}(1)$ pendant la recommandation. Ces résultats ne confirment donc la capacité de cette configuration (l'utilisation du skipping à l'apprentissage) à augmenter la couverture que pour les variantes du skipping complet et du skipping multinavigationnel. Étant donné que la variante du skipping de Shani n'a fourni aucune augmentation de la couverture, on peut déduire que cette configuration ne correspond pas au comportement réel de l'utilisateur. Une explication possible est qu'avec cette variante, chaque fois que la taille de la fenêtre est augmentée, les seules ressources considérées sont les plus éloignées, et donc celles qui ont la plus petite probabilité d'avoir une influence sur la prochaine action de l'utilisateur.

Relativement aux couvertures fournies par les deux autres variantes de skipping, ces dernières représentent à présent les meilleures alternatives.

Taille de la fenêtre	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Shani	84,5	84,5	84,5	84,5	84,5	84,5	84,5	84,5
Multinavigationnel	84,5	87,1	87,9	88,4	88,7	88,9	89,1	89,2
Complet	84,5	87,1	87,9	88,4	88,7	88,9	89,1	89,2

TABLE 2 – Couverture sur le corpus du Crédit Agricole S.A. quand le skipping n'est effectué que pendant l'apprentissage.

Taille de la fenêtre	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Shani	84,9	84,9	84,9	84,9	84,9	84,9	84,9	84,9
Multinavigationnel	84,9	87,8	88,8	89,4	89,9	90,4	90,7	90,9
Complet	84,9	87,8	88,8	89,4	89,9	90,4	90,7	90,9

TABLE 3 – Couverture sur le corpus de l'université DePaul quand le skipping n'est effectué que pendant l'apprentissage.

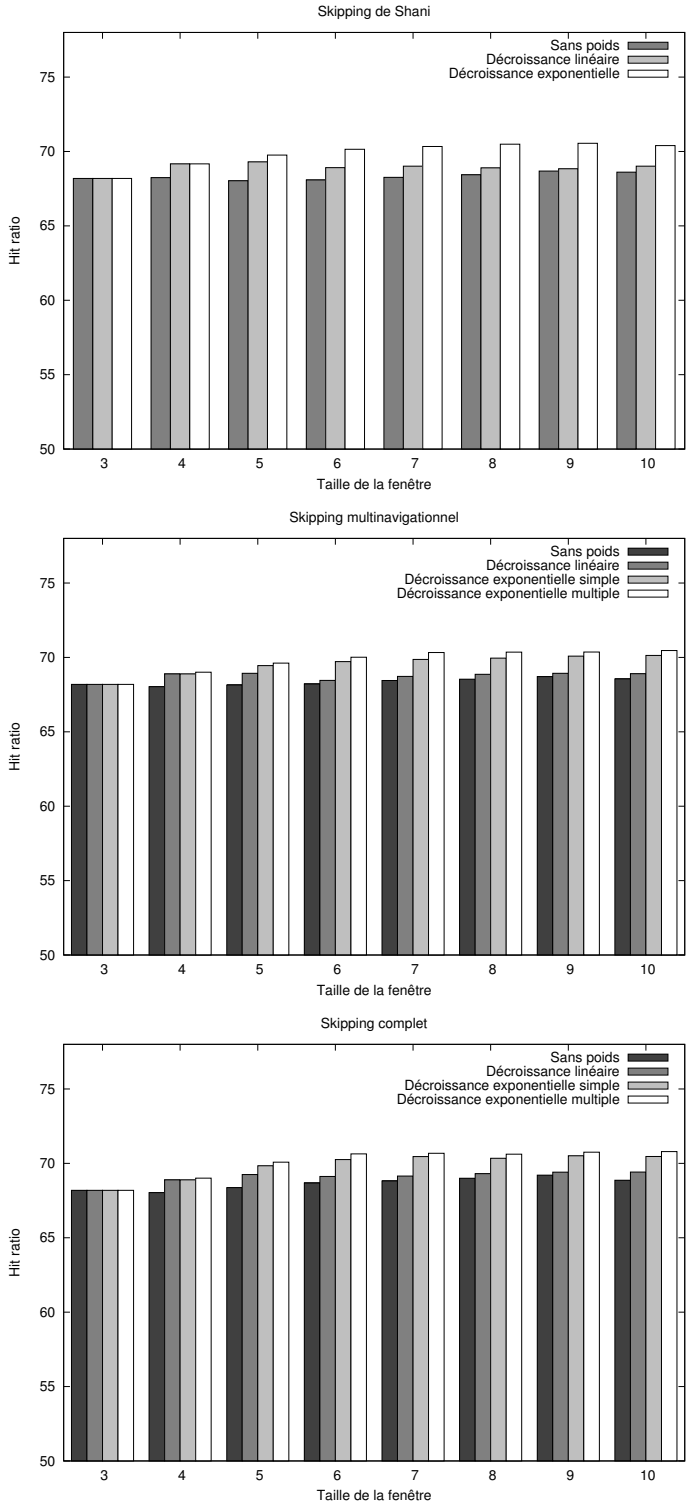

FIGURE 2 – Hit ratio du modèle SBR sur le corpus du Crédit Agricole S.A. en fonction de la variante de skipping et du schéma de pondération quand le skipping n'est appliqué qu'à l'apprentissage.

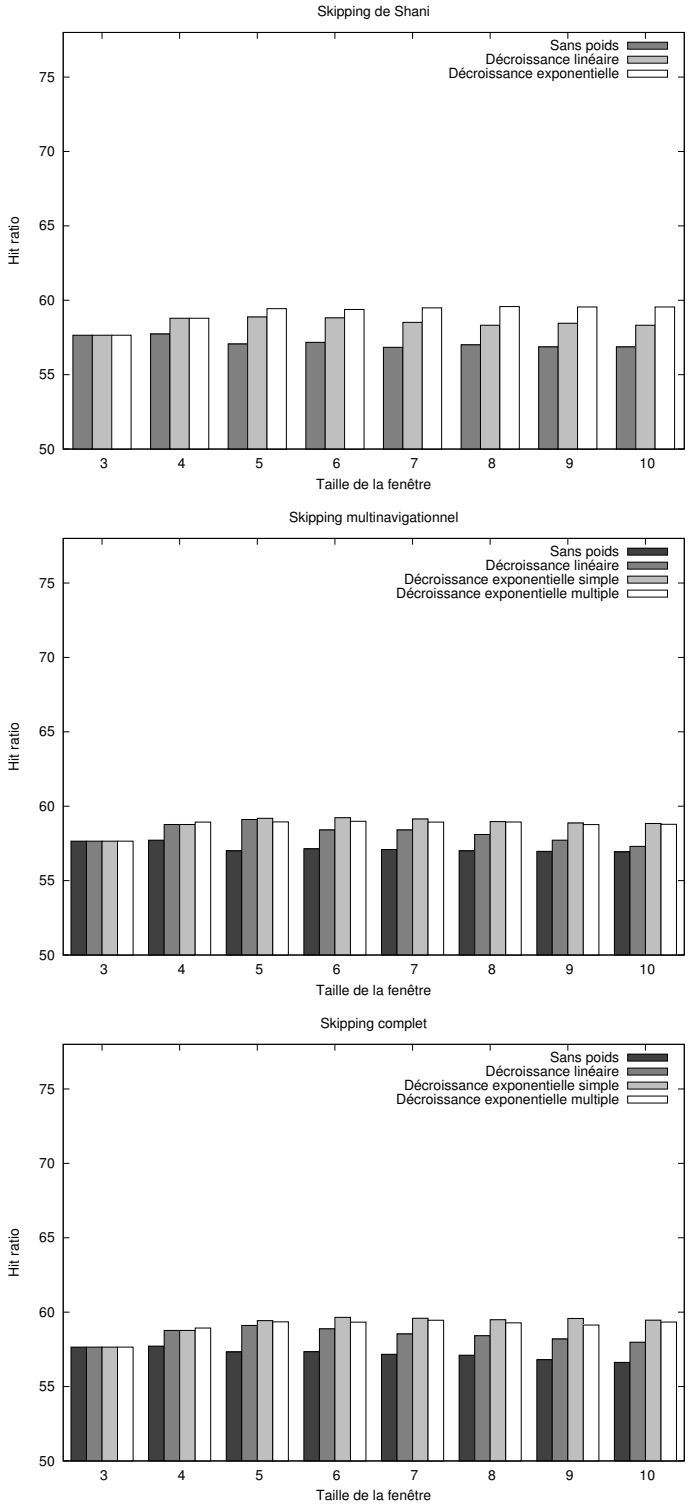

FIGURE 3 – Hit ratio du modèle SBR sur le corpus de l'université DePaul en fonction de la variante de skipping et du schéma de pondération quand le skipping n'est appliqué qu'à l'apprentissage.

■ Skipping effectué à l'apprentissage et pour le calcul des recommandations

La configuration étudiée est maintenant celle où le skipping est utilisé à l'apprentissage ainsi que pendant la phase de recommandation. La figure 4 et la figure 5 montrent les hit ratio obtenus respectivement sur le corpus du Crédit Agricole S.A. et celui de l'université DePaul, et les tableaux 4 et 5 les couvertures respectives. Ici encore, le skipping est utilisé avec une fenêtre glissante de taille allant de 3 à 10.

Taille de la fenêtre	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Shani	84,5	93,1	96,0	97,3	97,9	98,3	98,6	98,8
Multinavigationnel	84,5	96,2	97,9	98,6	98,9	99,2	99,3	99,5
Complet	84,5	96,2	98,3	98,9	99,2	99,5	99,6	99,7

TABLE 4 – Couverture sur le corpus du Crédit Agricole S.A. quand le skipping est effectué pendant l'apprentissage et pendant les recommandations.

Taille de la fenêtre	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Shani	84,9	93,6	96,4	97,4	98,0	98,5	98,7	98,9
Multinavigationnel	84,9	96,4	98,2	99,1	99,4	99,5	99,6	99,7
Complet	84,9	96,4	98,5	99,4	99,7	99,7	99,7	99,8

TABLE 5 – Couverture sur le corpus de l'université DePaul quand le skipping est effectué pendant l'apprentissage et pendant les recommandations.

On peut remarquer que les résultats fournis par la variante de Shani sont particulièrement mauvais sur le corpus de l'université DePaul. Quelle que soit la configuration choisie, augmenter la taille de la fenêtre diminue les valeurs de hit ratio obtenues. Sur le corpus du Crédit Agricole S.A., les résultats obtenus avec cette variante de skipping sont similaires à ceux d'un modèle de trigrammes standard, et même inférieurs dans certaines configurations. Considérer des paires d'éléments contigus dans un historique lointain ne semble donc pas approprié pour ce genre de tâche. Cela semble confirmer l'explication fournie pour expliquer la moindre performance de cette variante dans le jeu de tests précédents : ici, non seulement les n-grammes considérés à l'apprentissage ne sont pas représentatifs du comportement de l'utilisateur, mais en plus les éléments de l'historique considérés pour calculer les recommandations ne sont pas non plus représentatifs du comportement de l'utilisateur.

Sur les deux corpus, le meilleur hit ratio est fourni par la variante du skipping multinavigationnel combiné à la pondération selon la décroissance exponentielle multiple. En effet, sur le premier corpus, il passe de 68,2 à 73,5, soit une amélioration de 7,7%, et sur le second corpus, il passe de 57,6 à 60,1, soit une amélioration de 4,3%. L'amélioration en termes de précision est donc plus flagrante sur le corpus du Crédit Agricole S.A. que sur le corpus de l'université DePaul. Les résultats fournis par la variante du skipping complet fournissent des résultats

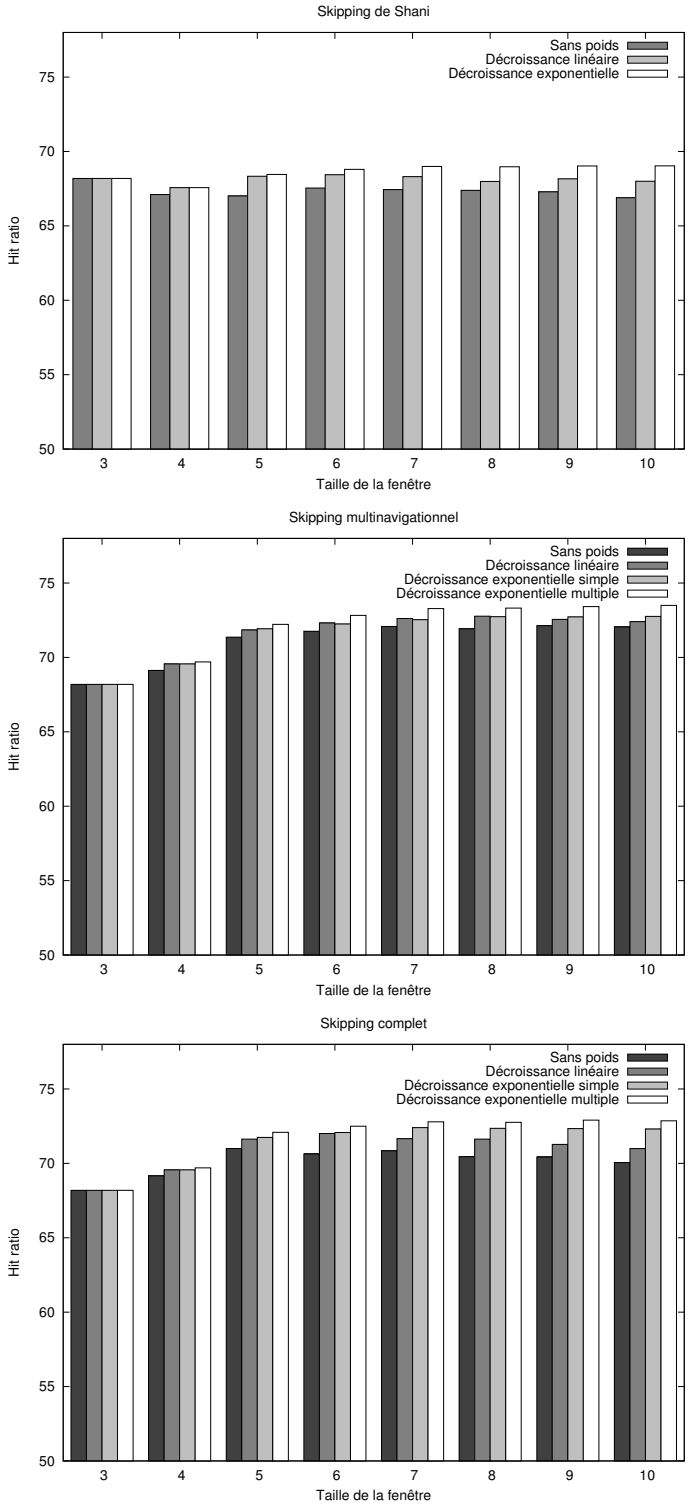

FIGURE 4 – Hit ratio du modèle SBR sur le corpus du Crédit Agricole S.A. en fonction de la variante de skipping et du schéma de pondération.

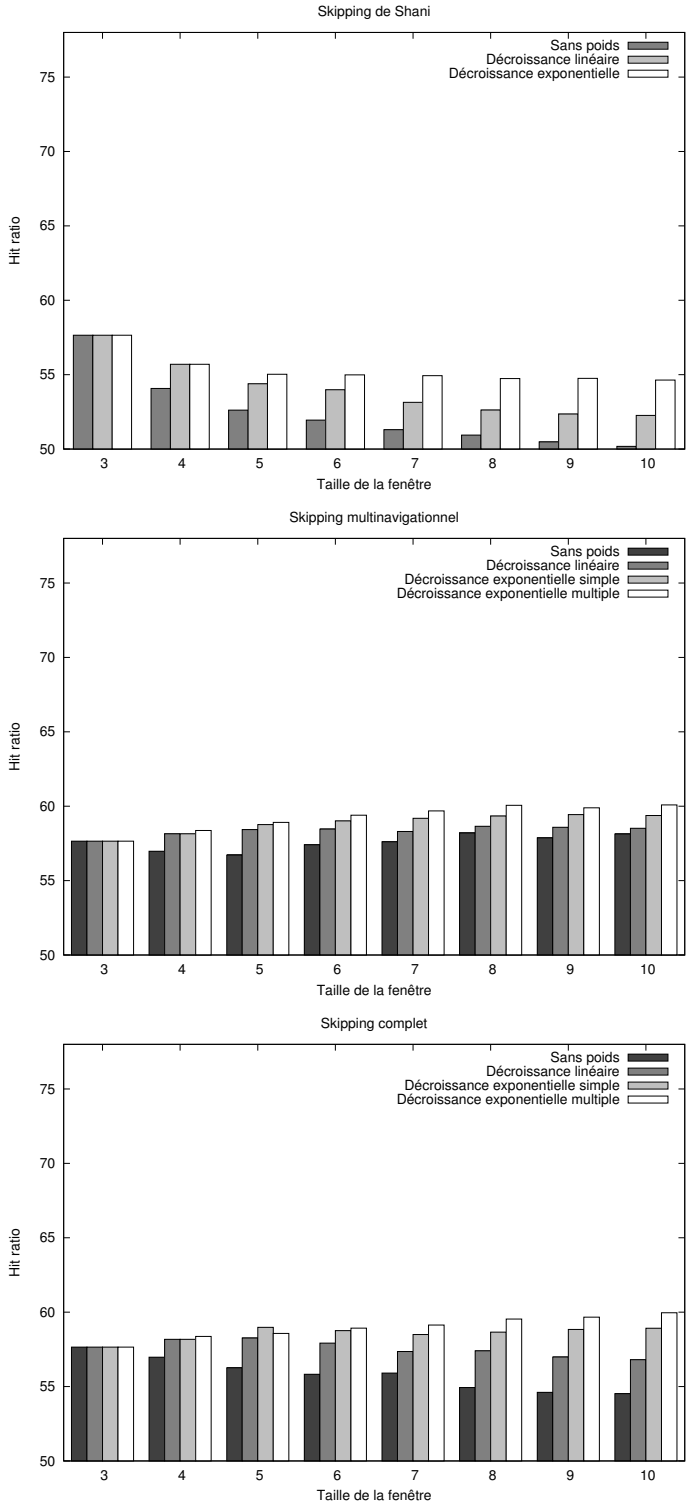

FIGURE 5 – Hit ratio du modèle SBR sur le corpus de l'université DePaul en fonction de la variante de skipping et du schéma de pondération.

comparables, mais avec une complexité en temps et en mémoire supérieure. Par conséquent, la variante du skipping multinavigationnel semble, ici encore, représenter la meilleure alternative.

Pour les trois variantes de skipping, ne pas utiliser de pondération, comme cela est communément le cas pour les modèles de motifs séquentiels non contigus, fournit de moins bons résultats. Le schéma de pondération qui fournit les meilleures valeurs de hit ratio est toujours la décroissance exponentielle multiple, quelle que soit la taille de la fenêtre, quelle que soit la variante de skipping utilisée (pour la variante du skipping de Shani, ce schéma de pondération est équivalent à la décroissance exponentielle simple).

Par conséquent, la configuration qui semble à présent la plus efficace est la variante du skipping multinavigationnel combinée avec le schéma de pondération selon une décroissance exponentielle multiple.

Les tableaux 4 et 5 montrent qu'augmenter la taille de la fenêtre à l'apprentissage et pendant la recommandation augmente la couverture. Cette fois, des couvertures presque totales sont fournies par les trois variantes. Étant donné les hit ratio présentés ci-dessus, la meilleure configuration pour le SBR est la variante du skipping multinavigationnel combinée avec le schéma de pondération selon une décroissance exponentielle multiple, ce qui valide les arguments théoriques mis en avant dans les sections précédentes.

b) Résistance au bruit

Cette section est dédiée à la comparaison de la résistance au bruit du modèle SBR aux deux modèles de l'état de l'art auquel il a été comparé jusqu'à présent dans cette thèse. La configuration utilisée pour le modèle SBR est celle qui a fourni les meilleurs résultats dans les expérimentations précédentes, à savoir la variante du skipping multinavigationnel et la pondération selon la décroissance exponentielle multiple.

Les résultats du all-k^{th}-order Markov model et du modèle de motifs séquentiels correspondent aux seuils de filtrage optimaux sur le support et la confiance. Il est important de constater que les motifs séquentiels ne peuvent pas être utilisés avec un seuil de support de 0 sur les versions des corpus où du bruit supplémentaire a été inséré, car les besoins en matière de capacité de stockage étaient trop importants.

Le hit ratio des différents modèles en fonction du taux de bruit inséré est présenté dans la figure 6 et la figure 7. On peut remarquer dans un premier temps que sur le corpus du Crédit Agricole S.A., les meilleurs résultats sont fournis par le modèle SBR. Quand aucun bruit supplémentaire n'est inséré, le modèle SBR fournit une amélioration significative en comparaison des motifs séquentiels. Si la différence est moins importante quand 15% de bruit est inséré, le modèle SBR fournit toujours une meilleure précision que le modèle de motifs séquentiels. Avec 30% de bruit supplémentaire, les deux modèles fournissent des résultats comparables.

Sur le corpus de l'université DePaul, le modèle de motifs séquentiels fournit des résultats légèrement supérieurs à ceux du modèle SBR, quelle que soit la quantité de bruit insérée. Cette amélioration n'est pas suffisante cependant pour compenser la plus grande complexité du modèle de motifs séquentiels.

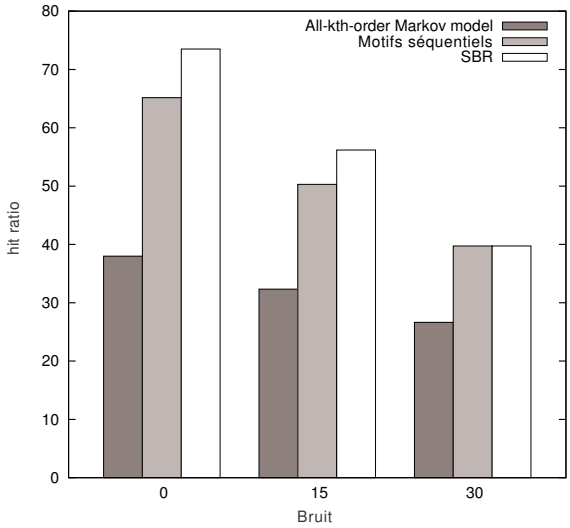

FIGURE 6 – Hit ratio des trois modèles sur les corpus bruités du Crédit Agricole S.A.

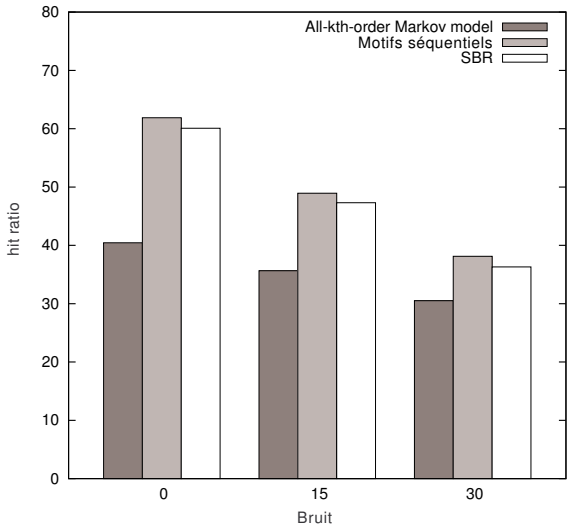

FIGURE 7 – Hit ratio des trois modèles sur les corpus bruités de l'université DePaul.

Par conséquent, la résistance au bruit du modèle SBR est soit similaire, soit légèrement inférieure à celle des motifs séquentiels. Cependant, cette infériorité est négligeable étant données sa plus grande précision et sa plus faible complexité en temps et en mémoire.

Le all-k^{th}-order Markov model fournit les recommandations les moins précises, ce qui confirme que l'utilisation de séquences contiguës n'est pas appropriée dans un environnement bruité. Cependant, la baisse de performance en fonction de la quantité de bruit est moins rapide que pour les deux autre modèles. Nous pensons que cela montre que ce modèle n'est pas adapté pour de longues séquences et est plus précis quand des séquences de taille inférieure sont considérées. En effet, plus la quantité de bruit insérée est grande, moins les états pour lesquels des correspondances seront trouvées seront grands.

Les études théoriques et expérimentales présentées dans cette section convergent donc vers la même conclusion. L'utilisation de séquences non contiguës est plus efficace que l'utilisation de séquences contiguës, et le modèle SBR que nous proposons représente une meilleure alternative à l'état de l'art : il a une complexité en temps et en mémoire plus basse, et fournit soit une meilleure précision, soit une précision comparable à celle d'un des modèles de l'état de l'art, tout en ayant une couverture comparable. De plus, ils montent que la variante de skipping et le schéma de pondération que nous avons proposés représentent une meilleure alternative aux variantes de skipping et aux schémas de pondération de l'état de l'art.

3 CONCLUSION

Dans ce chapitre, nous avons présenté le modèle de base de cette thèse. Nous avons dans un premier temps présenté brièvement le domaine de la modélisation statistique du langage. Nous avons ensuite montré que ce domaine possède de nombreuses similarités avec le domaine de la recommandation Web. Le domaine de la modélisation statistique du langage étant plus ancien que celui de la recommandation Web, de nombreuses approches ont été proposées et expérimentées, dont certaines peuvent être exploitées pour la recommandation Web. Cependant, si les deux domaines ont de nombreuses similarités, la navigation Web possède ses propres caractéristiques et spécificités. Nous avons donc décidé d'adapter à la navigation Web un modèle statistique de langage appelé modèle de n-grammes avec skipping qui présente des caractéristiques intéressantes du point de vue de la navigation Web.

Le modèle que nous proposons est appelé SBR pour Skipping-Based Recommender et exploite des séquences non contiguës pour calculer des recommandations, tout en ayant une faible complexité en temps et en mémoire. Ce modèle peut être utilisé selon plusieurs variantes de skipping, dont une que nous proposons et qui s'avère fournir de meilleurs résultats que les variantes de l'état de l'art. Pour accorder moins d'importance aux ressources distantes, le SBR peut être utilisé avec plusieurs schémas de pondération, dont un que nous proposons et qui s'avère fournir de meilleurs résultats que les schémas de pondération de l'état de l'art.

Dans la dernière section, le SBR est comparé de façon théorique et expérimentale à l'état de l'art en termes de complexité en temps et en mémoire ainsi que de précision et de résistance au bruit. Les études expérimentales sont effectuées sur les deux corpus de navigation Web

présentés dans le chapitre 2. Les résultats montrent que sur les deux corpus, considérer des séquences non contiguës comme le font le SBR et les modèles de motifs séquentiels fournit de meilleurs résultats. Enfin, ces résultats montrent que le modèle SBR représente une meilleure alternative à l'état de l'art : il a une complexité en temps et en mémoire plus basse, et fournit soit une meilleure précision, soit une précision comparable à un des modèles de l'état de l'art, tout en ayant une couverture comparable.

Chapitre 4

Vers un modèle précis, rapide et adaptatif

Dans le chapitre précédent, nous avons montré que la meilleure configuration du modèle SBR est celle qui utilise la variante du skipping multinavigationnel couplée avec le schéma de pondération de la décroissance exponentielle multiple. Cette configuration sera utilisée comme configuration de base dans ce chapitre. Par conséquent, quand nous parlerons du modèle SBR dans le reste de cette thèse, nous sous-entendrons que sa configuration est cette configuration de référence.

Dans ce chapitre nous nous attachons à améliorer notre modèle selon les trois critères communs de la navigation Web : la précision, la complexité et la couverture. Dans un premier temps, nous proposons différentes alternatives au modèle de référence permettant un apprentissage de jeux de pondérations plus élaborés, afin de valider le schéma de pondération de référence utilisé pour le SBR. Dans un deuxième temps nous proposons d'exploiter une caractéristique de la politique de combinaison des sous-historiques utilisée par le SBR pour calculer les recommandations selon un algorithme incrémental, afin d'améliorer la rapidité du modèle SBR. Enfin, nous proposons de nous inspirer du principe des all-k^{th}-order Markov model pour fournir une couverture totale.

1 PRÉCISION : MISE EN VALEUR DES ACTIONS INFORMATIVES DE L'UTILISATEUR

Le modèle SBR utilise le skipping selon l'approche de la fusion des occurrences de n-grammes. Comme mentionné dans la section 3.1 du chapitre 2, une alternative est de construire un sous-modèle par configuration de skipping. Cette alternative a l'avantage de permettre un apprentissage automatique de poids optimaux à affecter aux différentes configurations de skipping, plutôt que d'utiliser des valeurs analytiques relativement arbitraires. En poussant cette possibilité plus loin, il serait intéressant de pouvoir pondérer les n-grammes non pas simplement en fonction de leur configuration de skipping, mais plutôt en fonction de leur configura-

tion de prédilection. En effet, il se peut que certains n-grammes soient rencontrés plus souvent dans certaines configurations de skipping que d'autres (leurs configurations de prédilection), et qu'il soit judicieux de prendre en compte ce critère dans le calcul de la pondération.

Cette section a pour but de valider le schéma de pondération de référence utilisé pour le SBR relativement à des alternatives plus élaborées. Nous proposons quatre alternatives : deux interpolations de sous-modèles utilisant l'algorithme EM, une pondération en fonction de la fréquence des n-grammes par configuration de skipping, et un clustering des suites de ressources en fonction de leurs configurations de skipping de prédilection. Une étude expérimentale est ensuite présentée, dans laquelle ces modèles sont comparés au SBR.

1.1 Alternatives à la décroissance exponentielle

Comme montré dans le Tableau 1, quand le schéma de référence est appliqué à la variante du skipping multinavigationnel (la variante de référence), il fournit une décroissance en dents de scie. Dans ce tableau, ces pondérations sont montrées pour $n = 3$ et les triplets sont symbolisés en utilisant des croix. $\times\times\times$ correspond aux triplets de la forme (r_{i-2}, r_{i-1}, r_i), $\times - \times\times$ aux triplets de la forme (r_{i-3}, r_{i-1}, r_i), $\times\times -\times$ aux triplets de la forme (r_{i-3}, r_{i-2}, r_i) et ainsi de suite.

La raison pour laquelle un tel schéma a été choisi est qu'il semble plus approprié qu'une décroissance monotone. Par exemple, la configuration $\times - - \times\times$ devrait avoir plus d'importance que la configuration $\times\times -\times$, parce que dans le premier cas la deuxième ressource est proche de la navigation en cours alors que dans le second, le deux premières ressources sont loin dans l'historique.

Configuration de skipping	**Pondération**
$\times\times\times$	1,0
$\times - \times\times$	0,75
$\times\times -\times$	0,5
$\times - - \times\times$	0,63
$\times\times - - \times$	0,25
$\times - - - \times\times$	0,56
$\times\times - - -\times$	0,13

TABLE 1 – Décroissance en dents de scie obtenue avec le schéma de décroissance exponentielle multiple.

Ayant explicité le schéma de pondération de référence, nous pouvons présenter les quatre alternatives auquel il sera comparé.

1.1.1 Interpolation simple de sous-modèles (SBR 2)

La première alternative que nous présentons est la forme classique du modèle de skipping de la littérature (Goodman, 2001), mais utilisant la variante du skipping multinavigationnel. Ce modèle consiste en une interpolation de sous-modèles : un sous-modèle par configuration de skipping est construit et les sous-modèles obtenus sont combinés en utilisant une interpolation linéaire. Par exemple, la probabilité d'une ressource r_i en fonction de l'historique h peut être donnée par la formule suivante :

$$P(r_i|h) = \alpha P(r_i|r_{i-2}, r_{i-1}) + \beta P(r_i|r_{i-3}, r_{i-1}) + \gamma P(r_i|r_{i-3}, r_{i-2})$$

où $0 \leqslant \alpha \leqslant 1$, $0 \leqslant \beta \leqslant 1$, $0 \leqslant \gamma \leqslant 1$ et $\alpha + \beta + \gamma = 1$.

Cette approche est une des deux alternatives d'utilisation du skipping possibles présentées dans le chapitre 2. Nous rappelons qu'un inconvénient lié à l'utilisation d'un sous-modèle par configuration de skipping est que cela implique une plus grande complexité en mémoire, car les n-grammes stockés peuvent être redondants d'un sous-modèle à l'autre. Plus il y a de sous-modèles, plus cette complexité est grande, ce qui limite la taille de l'historique pris en compte par rapport au modèle SBR.

L'apprentissage du modèle se fait en trois étapes :

1. Construction des listes de n-grammes dans chacune des configurations de skipping à partir du corpus d'apprentissage ;
2. Calcul des probabilités conditionnelles ;
3. Calcul des coefficients de l'interpolation.

Au moment de l'utilisation des modèles, il y a correspondance entre la configuration de skipping associée à l'état considéré et le sous-modèle utilisé. Par exemple, étant donné l'historique $h = (a, b, c, x)$, et la ressource candidate r, il est possible d'utiliser l'interpolation suivante :

$$P(r|h) = \alpha P_1(r|b, c, x) + \beta P_2(r|a, c, x) + \gamma P_3(r|a, b, c)$$

P_1 correspond alors au sous-modèle de quadrigrammes contigus, P_2 au modèle de quadrigrammes avec un saut entre la première et la deuxième ressource, et P_3 à celui avec un saut entre la troisième et la quatrième ressource.

Un désavantage par rapport au modèle SBR, est qu'une telle configuration entraîne une diminution de la couverture. En effet, le SBR de par sa configuration autorise l'utilisation de n-grammes d'une configuration de skipping à l'apprentissage différente de celle rencontrée dans l'historique lors d'une recommandation. De cette manière, plus de correspondances peuvent être trouvées et donc la couverture est plus grande. Pour calculer les coefficients de l'interpolation, nous utilisons l'algorithme EM sur un corpus de validation. Dans nos expérimentations, nous nommons ce modèle **SBR 2**.

1.1.2 Interpolation mixte de sous-modèles (SBR 3)

Afin d'utiliser des configurations de skipping différentes à l'apprentissage et à la recommandation, nous proposons d'effectuer une interpolation mixte des sous-modèles : chaque configuration de skipping peut être utilisée avec une configuration différente. À notre connaissance, ce type d'interpolation n'a jamais été expérimenté.

Le modèle obtenu est alors également très proche du SBR, à la différence que les n-grammes ne sont pas stockés dans une même liste, ce qui permet de retrouver dans quelle configuration de skipping quels n-grammes ont été rencontrés lors de la phase d'apprentissage. La complexité en mémoire engendrée est encore supérieure à celle de la configuration précédente, et la complexité en temps est plus grande : pour $n = 3$ et en utilisant la variante du skipping multi-navigationnel, avec une fenêtre de taille D, la complexité pour chaque recommandation est $\mathcal{O}(D^2)$ au lieu de $\mathcal{O}(D)$.

Pour calculer les coefficients de l'interpolation, nous utilisons également l'algorithme EM sur un corpus de validation. Dans nos expérimentations, nous nommons ce modèle **SBR 3**.

1.1.3 Pondération en fonction des fréquences par configuration de skipping (SBR 4)

Certains n-grammes peuvent apparaître plus souvent dans certaines configurations de skipping que d'autres. Nous proposons donc de mettre en valeur ces n-grammes quand ils apparaissent dans ces configurations, et de limiter leur importance dans les autres cas, ce que ne permettent pas les modèles présentés auparavant.

Le modèle que nous proposons est très proche du SBR : la différence se situe au niveau de la pondération, qui au lieu d'être effectuée en fonction des distances, est effectuée en fonction de la fréquence des configurations de skipping des n-grammes. Pour cela, une étape supplémentaire d'apprentissage de ces fréquences est effectuée. Par exemple, pour une fenêtre de taille 4 et $n = 3$, les occurrences des trigrammes dans chacune des configurations $\times\times\times$, $\times-\times\times$ et $\times\times-\times$ sont comptées. Ces occurrences sont alors normalisées.

Par exemple, pour le trigramme $\langle abc \rangle$, qui apparait 600 fois dans la configuration $\times\times\times$, 300 fois dans la configuration $\times-\times\times$ et 100 fois dans la configuration $\times\times-\times$, les poids des configurations respectives de ce trigramme spécifique seront $0,6$, $0,3$ et $0,1$.

Ces poids sont alors utilisées comme dans le SBR présenté dans le chapitre 3. L'inconvénient de ce modèle est qu'il a un plus grand nombre de paramètres que les modèles précédents. Par conséquent, plus de données d'apprentissage sont nécessaires pour que le modèle soit fiable.

Par la suite, nous nommerons ce modèle **SBR 4**.

1.1.4 Classes de n-grammes (SBR 5)

Le modèle précédent pouvant comporter trop de paramètres, nous proposons une configuration intermédiaire. Elle consiste à utiliser des classes de n-grammes en fonction de leurs

configurations de skipping de prédilection. Nous appelons configuration de prédilection la configuration dans laquelle un n-gramme est le plus souvent rencontré.

Nous faisons correspondre une classe à chaque configuration de skipping, et y regroupons les n-grammes qui apparaissent le plus souvent dans la configuration correspondante. Puis, pour chaque classe, un modèle d'interpolation mixte (SBR 3) est construit. Ainsi, plusieurs jeux de pondération sont disponibles et peuvent être utilisés en fonction de la classe à laquelle appartient un n-gramme donné. Le modèle obtenu représente donc une situation intermédiaire entre le modèle SBR et le modèle SBR 4.

Le processus d'apprentissage se fait donc en trois étapes :

1. Construction des listes de n-grammes ;
2. Détermination des classes de n-grammes ;
3. Calcul des poids pour chaque classe avec l'algorithme EM.

La phase de recommandation fonctionne comme suit :

1. Détermination de la classe à laquelle appartient le n-gramme ;
2. Application du modèle et du poids correspondants.

Par la suite, nous nommerons ce modèle **SBR 5**.

1.2 Étude expérimentale

Dans cette section, nous validons expérimentalement le schéma de pondération de référence utilisé pour le SBR relativement aux modèles alternatifs permettant des pondérations plus élaborées. Les expérimentations sont effectuées avec la variante du skipping multinavigationnel pour tous les modèles et des listes de recommandation de taille 10. Contrairement aux expérimentations du chapitre précédent, un corpus de validation doit être utilisé pour l'estimation des poids. Par conséquent, les corpus sont divisés en un corpus d'apprentissage, un corpus de validation et un corpus de test selon une répartition de 70%, 20% et 10% respectivement. Les corpus utilisés sont les corpus de navigation Web de l'université DePaul et du Crédit Agricole S.A. utilisés dans les expérimentations du chapitre précédent. Bien que la répartition des données soit différente, cette différence ne concerne que le corpus d'apprentissage, et les expérimentations sont effectuées sur les mêmes données que les expérimentations du chapitre 3.

1.2.1 Interpolations

Nous nous intéresserons dans un premier temps à la comparaison du modèle SBR aux interpolations de sous-modèles présentés dans les sections 1.1.1 et 1.1.2. La variante de skipping

utilisée étant la même pour toutes les expérimentations, les sous-modèles considérés correspondent aux sous-modèles du tableau 1. Pour des raisons de clarté, les résultats ne sont pas fournis au-delà d'une fenêtre de taille $D = 6$.

Les tableaux 2 et 3 présentent les résultats obtenus quand un seul des sous-modèles utilisés par le SBR 2 est utilisé. En comparant deux sous-modèles à portée égale (*e.g.* le sous-modèle 4 et le sous-modèle 5), on peut remarquer que la configuration où le deuxième élément est contigu au dernier fournit une meilleure précision que l'autre configuration (qui est celle utilisée par (Shani *et al.*, 2005)). Cela confirme l'hypothèse sur laquelle est basée le choix de la décroissance en dents de scie du schéma de la décroissance exponentielle multiple, à savoir qu'un n-gramme construit à partir d'éléments plus proches dans l'historique a une meilleure capacité prédictive. En revanche, la seconde configuration fournit une meilleure couverture. En effet, les états correspondent toujours à des suites contiguës de ressources et sont donc moins épars. Par conséquent, un plus grand nombre de correspondances peut être trouvé dans le modèle. Ces résultats illustrent donc la complémentarité des deux configurations, et explicitent l'origine de la supériorité de la variante du skipping multinavigationnel sur la variante du skipping de Shani.

Sous-modèles		**Préc.**	**Couv.**
1	$\times\times\times$	66,9	83,3
2	$\times-\times\times$	56,8	73,6
3	$\times\times-\times$	48,4	81,4
4	$\times--\times\times$	55,8	66,2
5	$\times\times--\times$	40,3	79,4
6	$\times---\times\times$	49,3	62,3
7	$\times\times---\times$	37,0	77,1

TABLE 2 – Précision et couverture des sous-modèles de skipping du SBR 2 sur le corpus du Crédit Agricole S.A.

Sous-modèles		**Préc.**	**Couv.**
1	$\times\times\times$	56,3	82,8
2	$\times-\times\times$	46,2	68,9
3	$\times\times-\times$	30,3	80,4
4	$\times--\times\times$	40,4	59,2
5	$\times\times--\times$	21,0	76,9
6	$\times---\times\times$	33,3	49,5
7	$\times\times---\times$	16,4	72,0

TABLE 3 – Précision et couverture des sous-modèles de skipping du SBR 2 sur le corpus de l'université DePaul

Les poids calculés par l'algorithme EM pour le modèle SBR 2 sont présentés dans les tableaux 4 et 5. Afin de les rendre comparables, les valeurs des poids du SBR de référence ont été normalisées, de sorte de sommer à 1. Comme on peut le constater, à part la plus grande importance accordée à la configuration contiguë correspondant au modèle de trigrammes standard (sous-modèle 1), les deux jeux de poids sont très différents. Les deux schémas attribuent une pondération qui décroit en fonction de la distance ; cependant, à portée égale, l'importance accordée aux configurations de skipping est différente. Sur le corpus du Crédit Agricole S.A., on observe une décroissance monotone. Sur le corpus de l'université DePaul, on observe une décroissance en dents de scie inverse à celle de la décroissance exponentielle.

		$D = 4$		$D = 5$		$D = 6$	
Sous-modèles		**SBR 2**	**SBR**	**SBR 2**	**SBR**	**SBR 2**	**SBR**
1	$\times\times\times$	0,75	0,44	0,71	0,32	0,69	0,26
2	$\times-\times\times$	0,15	0,33	0,11	0,24	0,10	0,20
3	$\times\times-\times$	0,10	0,22	0,07	0,16	0,07	0,13
4	$\times---\times\times$			0,05	0,20	0,04	0,16
5	$\times\times---\times$			0,05	0,08	0,03	0,07
6	$\times----\times\times$					0,03	0,15
7	$\times\times----\times$					0,03	0,03

TABLE 4 – Comparaison des poids du modèle SBR 2 et du modèle SBR sur le corpus du Crédit Agricole S.A.

		$D = 4$		$D = 5$		$D = 6$	
Sous-modèles		**SBR 2**	**SBR**	**SBR 2**	**SBR**	**SBR 2**	**SBR**
1	$\times\times\times$	0,77	0,44	0,72	0,32	0,69	0,26
2	$\times-\times\times$	0,10	0,33	0,08	0,24	0,08	0,20
3	$\times\times-\times$	0,12	0,22	0,10	0,16	0,10	0,13
4	$\times---\times\times$			0,04	0,20	0,03	0,16
5	$\times\times---\times$			0,06	0,08	0,05	0,07
6	$\times----\times\times$					0,02	0,15
7	$\times\times----\times$					0,04	0,03

TABLE 5 – Comparaison des poids du modèle SBR 2 et du modèle SBR sur le corpus de l'université DePaul.

Les tableaux 6 et 7 présentent une comparaison en termes de hit ratio et de couverture des interpolations de sous-modèles, simples (SBR 2) et mixtes (SBR 3), et du SBR sur les corpus du Crédit Agricole S.A. et de l'université DePaul respectivement. La partie droite des tableaux correspond donc aux résultats présentés dans le chapitre 3, à la différence que les données d'apprentissage sont plus petites de 22%.

Nous pouvons voir que malgré le calcul de poids optimaux avec l'algorithme EM, les interpolations simples fournissent une moins bonne précision que le SBR. Les interpolations mixtes quant à elles fournissent des résultats soit légèrement supérieurs à ceux du SBR (tableau 6) soit légèrement inférieurs, la différence n'étant pas significative (tableau 7).

Plusieurs raisons peuvent expliquer ce résultat : (1) La décroissance exponentielle multiple est proche de la pondération optimale, (2) les poids calculés sur le corpus de validation étaient optimaux pour les données de validation, mais pas pour les données de test (3) L'algorithme EM maximise la vraisemblance des sessions ; il n'est donc pas garanti qu'il maximise le hit ratio.

Au vu de ces résultats, les interpolations simples et mixtes ayant une plus grande complexité en temps et en mémoire (*cf.* sections 1.1.1 et 1.1.2), le SBR représente une meilleure alternative.

	SBR 2		**SBR 3**		**SBR**	
D	Préc.	Couv.	Préc.	Couv.	Préc.	Couv.
3	66,9	83,3	66,9	83,3	66,9	83,3
4	65,7	93,2	66,9	95,0	66,9	95,0
5	66,6	95,2	69.0	97,1	69,3	97,1
6	**66,8**	**96,5**	**69,7**	**98,0**	**70,0**	**98,0**

TABLE 6 – Précision et couverture des interpolations de sous-modèles et du SBR en fonction de la taille de la fenêtre *D* sur le corpus du Crédit Agricole S.A.

	SBR 2		**SBR 3**		**SBR**	
D	Préc.	Couv.	Préc.	Couv.	Préc.	Couv.
3	56,3	82,8	56,3	82,8	56,3	82,8
4	54,3	93,4	57,5	95,4	57,3	95,4
5	54,3	96,1	58,0	97,8	57,5	97,8
6	**54,3**	**97,1**	**58,8**	**98,9**	**58,5**	**98,9**

TABLE 7 – Précision et couverture des interpolations de sous-modèles et du SBR en fonction de la taille de la fenêtre *D* sur le corpus de l'université DePaul

Comme nous l'avons déjà mentionné, les interpolations ont l'inconvénient de ne fournir qu'un jeu de pondération par configuration de skipping. Une pondération dépendante à la fois des configurations de skipping et des n-grammes considérés constitue une modélisation plus précise encore, et pourrait améliorer la précision des recommandations. C'est à cet aspect que nous allons maintenant nous intéresser.

1.2.2 Pondération en fonction des fréquences par configuration de Skipping

Nous nous intéressons à présent à la possibilité de manipuler plusieurs jeux de pondération par configuration de skipping. Les tableaux 8 et 9 montrent les valeurs de hit ratio obtenues par le modèle de pondération en fonction des fréquences par configuration de skipping (SBR 4), le modèle utilisant des classes de configurations de prédilection (SBR 5) et le SBR standard. La couverture ne figure pas dans les tableaux car les trois modèles ont exactement la même.

	SBR 4	**SBR 5**	**SBR**
3	66,9	66,9	66,9
4	66,2	63,8	66,9
5	67,8	64,5	69,3
6	68,4	63,7	**70,0**

TABLE 8 – Précision des modèles SBR 4 et 5 et du SBR standard en fonction de la taille de la fenêtre D sur le corpus du Crédit Agricole S.A.

	SBR 4	**SBR 5**	**SBR**
3	56,3	56,3	56,3
4	52,6	54,4	57,3
5	51,0	53,4	57,5
6	51,5	52,9	**58,5**

TABLE 9 – Précision des modèles SBR 4 et 5 et du SBR standard en fonction de la taille de la fenêtre D sur le corpus de l'université DePaul.

Comme on peut le voir, prendre en compte les fréquences par configuration de skipping n'améliore pas les résultats. Il semble donc que seule la distance doive être prise en compte pour la mise en valeur des différents états de l'historique, et que les n-grammes n'ont pas de configuration de skipping de prédilection à exploiter.

La décroissance exponentielle multiple présentée dans le chapitre 3 s'avère donc représenter la façon la plus adaptée de pondérer les états de l'historique.

1.3 Conclusion

Ayant comparé expérimentalement le SBR de référence à des alternatives de plus en plus élaborées, nous avons montré que le schéma de pondération selon une décroissance exponentielle fournit des résultats soit meilleurs, soit comparables. Par conséquent ce schéma de pondération, bien qu'étant analytique et relativement arbitraire, reste la meilleure alternative à notre connaissance.

2 RAPIDITÉ : INTÉGRATION INCRÉMENTALE DE SOUS-HISTORIQUES

Le chapitre précédent nous a montré que la prise en compte d'information distante améliore la précision des recommandations, mais augmente également la complexité en temps. Le modèle SBR a l'avantage de pouvoir prendre en compte une telle information distante tout en ayant une faible complexité en temps. Cependant, il se peut que cette complexité soit trop grande dans un contexte de recommandation en temps réel. Par exemple dans le contexte du m-commerce [1] (Brun et Boyer, 2010), prendre en compte toutes les combinaisons à $n-1$ éléments pourrait s'avérer trop coûteux. Pour répondre à ce problème, nous proposons dans cette section une adaptation du SBR qui permette de fournir les meilleures recommandations possibles dans un temps limité. Nous présentons ensuite des expérimentations de cette adaptation.

2.1 Approche proposée

Nous nous focalisons sur la réduction du nombre nécessaire d'itérations pour délivrer les recommandations à l'utilisateur. Rappelons que pour calculer les recommandations, le modèle SBR combine les probabilités conditionnelles associées aux états ayant des correspondances avec l'historique selon la formule 3.6 de la section 1.3.3 du chapitre 3 page 60 :

$$q(\rho, h) = 1 - \prod_{\sigma} (1 - P(\rho \mid \sigma))$$

Une particularité de cette formule est qu'elle permet d'incorporer progressivement les sous-historiques σ dans le calcul. Par exemple, supposons que deux sous-historiques σ_1 et σ_2 soient trouvés dans h, et que ces sous-historiques aient une probabilité respective p_1 et p_2 de mener à la ressource ρ ; le score $q(\rho, h)$ associé à la ressource ρ sera alors le suivant :

$$q(\rho, h) = 1 - (1 - p_1 w_1)(1 - p_2 w_2)$$

Supposons à présent qu'un troisième sous-historique plus distant σ_3 avec une probabilité p_3 soit trouvé. Le score $q'(\rho, h)$ associé à la ressource ρ devrait être le suivant :

$$q'(\rho, h) = 1 - (1 - p_1 w_1)(1 - p_2 w_2)(1 - p_3 w_3)$$

Or, il est possible de calculer le score $q'(\rho, h)$ associé à ces trois sous-historiques à partir du score $q(\rho, h)$ associé aux deux premiers sous-historiques :

1. Commerce mobile.

$$\begin{aligned} q'(\rho,h) &= 1-(1-p_1w_1)(1-p_2w_2)(1-p_3w_3) \\ &= 1-(1-(1-(1-p_1w_1)(1-p_2w_2)))(1-p_3w_3) \\ &= 1-(1-q(\rho,h))(1-p_3w_3) \end{aligned} \tag{4.1}$$

Cette propriété n'est pas valable pour les autres politiques de combinaison de l'état de l'art. Par conséquent nous proposons d'exploiter cette caractéristique afin de calculer le score de chaque ressource de façon incrémentale, et permettre le calcul de recommandations dans un temps limité. Nous proposons de débuter le calcul des recommandations en utilisant le sous-historique le plus proche (constituant une séquence contiguë), car c'est la configuration qui, utilisée seule, permet d'obtenir les meilleures performances, puis de raffiner ces recommandations tant que le délai prédéterminé n'est pas passé en incorporant des sous-historiques de plus en plus distants.

Cela représente donc un fonctionnement inverse par rapport au fonctionnement habituel des approches de la fouille de données : en particulier, le all-k^{th}-order Markov model et les motifs séquentiels utilisés avec la politique de la longueur maximale (*cf.* chapitre 1 section 3.3). Ces approches commencent par considérer le plus grand sous-historique possible, et si aucune correspondance n'est trouvé entre ce sous-historique et le modèle, la taille de ce sous-historique est réduite, jusqu'à ce qu'une recommandation puisse être calculée. Or, la couverture fournie par les sous-historiques de grande taille est très inférieure à celle fournie par les sous-historiques de taille réduite. Par conséquent, pour un délai de calcul de recommandation donné, le SBR a une plus grande probabilité de pouvoir fournir une recommandation que les modèles traditionnels.

Cependant, en limitant le nombre de sous-historiques considérés pour calculer les recommandations, la couverture est amoindrie. Une manière de fournir la même couverture que lorsque tous les sous-historiques utiles sont utilisés, est de limiter non pas le nombre de sous-historiques, mais le nombre de *sous-historiques informatifs*, c'est-à-dire le nombre de sous-historiques pour lesquels une correspondance dans le modèle est trouvée. Ainsi, en limitant par exemple le nombre de sous-historiques informatifs à 1, il est possible de fournir la même couverture que quand tous les sous-historiques sont considérés ; cependant, la précision devrait être moins élevée que lorsque plusieurs sous-historiques sont combinés. La perte en précision peut cependant être suffisamment faible en fonction du nombre de sous-historiques informatifs considérés pour être compensée par le gain en temps de calcul.

2.2 Expérimentation

Nous nous intéressons ici aux performances obtenues lorsqu'un nombre limité de sous-historiques informatifs est utilisé. Comme dans nos autres expérimentations, nous utilisons une fenêtre de taille $D = 10$. La variante du skipping multinavigationnel implique donc la considération de 13 sous-historiques pour effectuer chaque recommandation. Notons que ce

nombre n'est valable que quand un historique suffisamment long est disponible : par exemple quand un utilisateur commence une nouvelle session, aucun historique n'est disponible. Le nombre moyen d'itérations est donc inférieur à 13.

Nous nous intéressons dans un premier temps aux résultats obtenus lorsqu'au maximum 1 sous-historique informatif est utilisé, c'est-à-dire que le processus de recommandation est réitéré jusqu'à ce qu'un historique informatif soit trouvé. Ce modèle est appelé SBR minimal. Le tableau 10 montre le nombre moyen d'itérations, le hit ratio et la couverture obtenus sur les corpus de l'université DePaul et du Crédit Agricole S.A.

	SBR de référence		SBR minimal	
	CA	DePaul	CA	DePaul
Itérations	7,9	6,1	1,2	1,2
Hit ratio	73,5	60,1	66,2	54,5
Couverture	98,9	99,7	98,9	99,7

TABLE 10 – Recommandations avec un nombre maximum de sous-historiques informatifs de 1.

Le tableau 10 montre que le nombre d'itérations du SBR minimal est très inférieur à celui du SBR de référence (il est plus de 5 fois inférieur sur chacun des corpus) tout en fournissant la même couverture (99,7% et 98,9%). Cependant, comme nous l'avons escompté, les recommandations ne sont pas aussi précises : une réduction du hit ratio de 9,3% sur le corpus de l'université DePaul et de 7,9% sur le corpus de Crédit Agricole S.A. est obtenue.

Nous rappelons que quand le skipping n'est utilisé qu'à l'apprentissage (section 2.2.2 chapitre 3), les valeurs de hit ratio obtenues sur chaque corpus étaient de 58,8 et 70,1% respectivement, avec une couverture de 89,2% et 91,5% respectivement. Cela signifie que les 10,2% et 7,4% de nouveaux cas de recommandation que permet le SBR minimal induisent une réduction de la précision globale de 7,5% et 5,6% respectivement. Cependant, quand tous les sous-historiques à l'intérieur de la fenêtre sont pris en compte (SBR de référence), le hit ratio atteint 60,1 et 73,5. Par conséquent, quand une recommandation ne peut être déduite du sous-historique contigu, le calcul des recommandations selon un nombre maximum de sous-historiques informatifs de 1 entraîne une baisse de précision ; mais quand plus de sous-historiques sont considérés, les recommandations sont raffinées et la précision augmente. C'est pourquoi il nous semble intéressant d'observer les performances quand la situation intermédiaire est utilisée : lorsqu'un nombre maximum de sous-historiques informatifs variable i est utilisé pour fournir les recommandations. Nous rappelons qu'en réalité, le nombre moyen de sous-historiques pris en compte se situe en-dessous de ce seuil (i) car un nombre inférieur de sous-historiques est disponible en début de session.

Les figures 1 et 2 montrent les valeurs de hit ratio et le nombre moyen d'itérations en fonction du nombre maximum de sous-historiques informatifs i sur le corpus du Crédit Agricole S.A., et les figures 3 et 4 les mêmes mesures sur le corpus de l'université DePaul.

Sur les deux corpus, les valeurs de hit ratio convergent asymptotiquement aux environs de 6 sous-historiques informatifs. À l'inverse, le nombre moyen d'itérations augmente presque

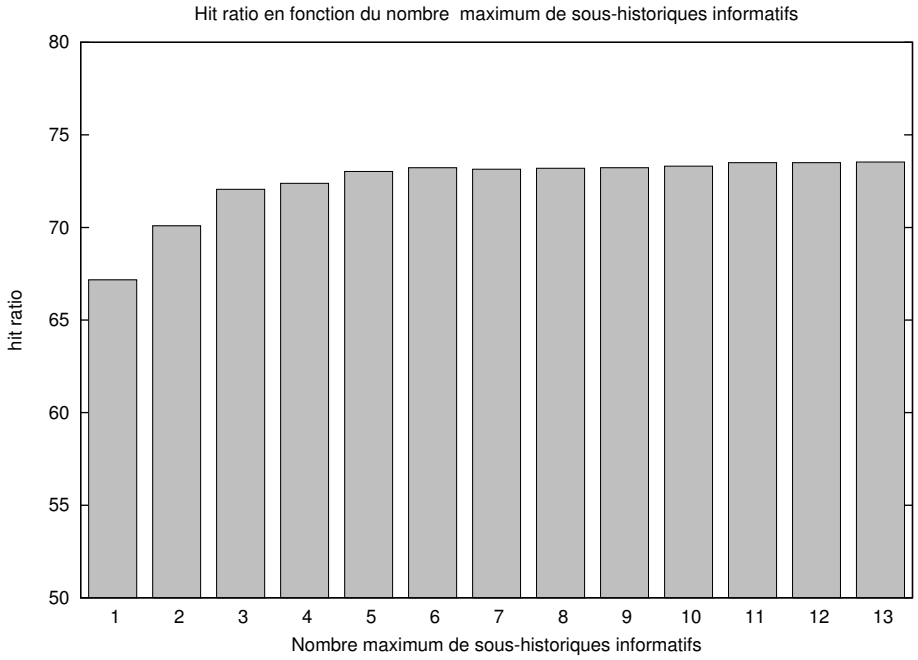

FIGURE 1 – Hit ratio en fonction du nombre maximum de sous-historiques informatifs sur le corpus du Crédit Agricole S.A.

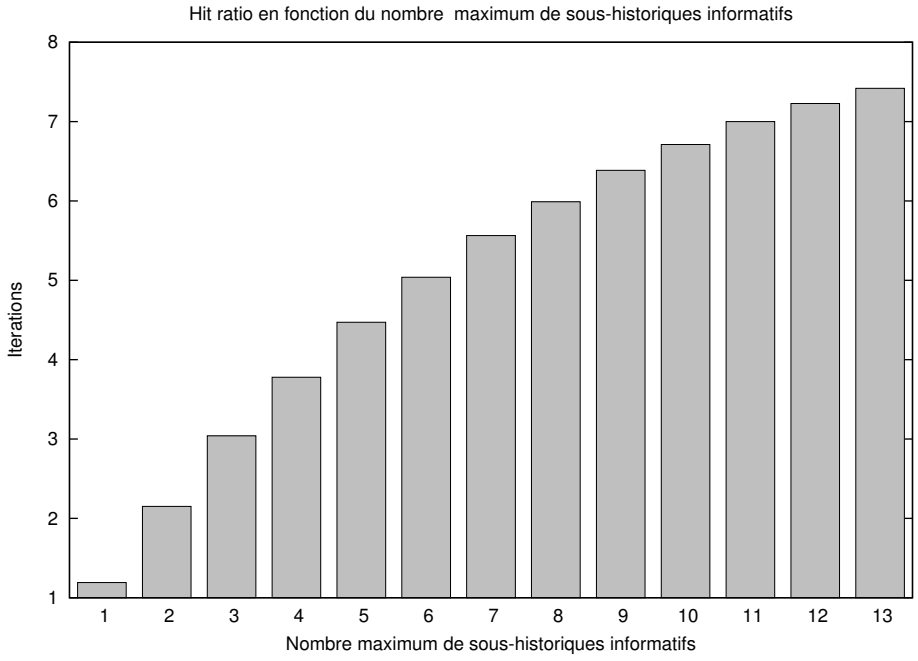

FIGURE 2 – Nombre moyen d'itérations en fonction du nombre maximum de sous-historiques informatifs sur le corpus du crédit Agricole S.A.

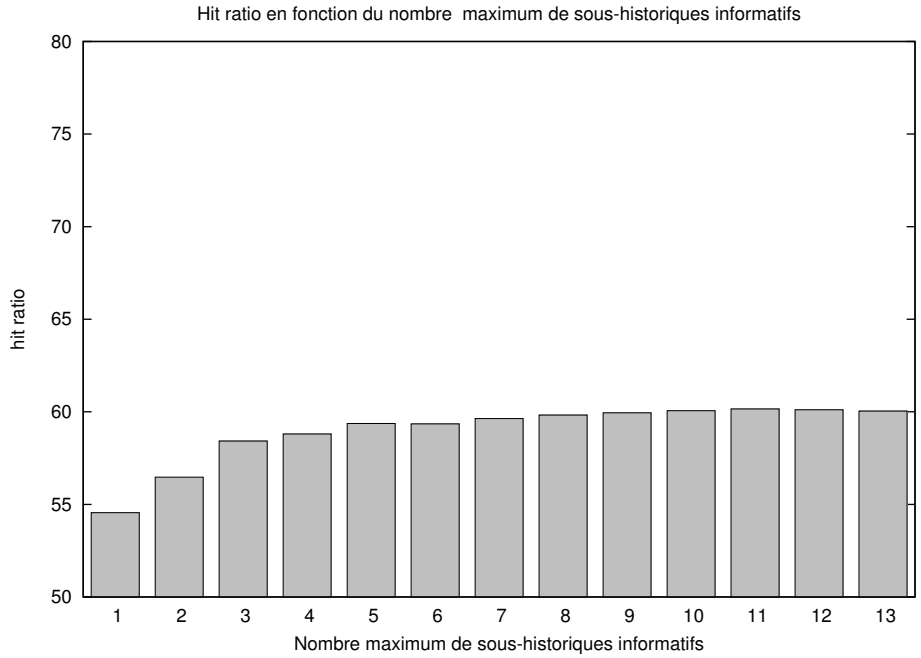

FIGURE 3 – Hit ratio en fonction du nombre maximum de sous-historiques informatifs sur le corpus de l'université DePaul.

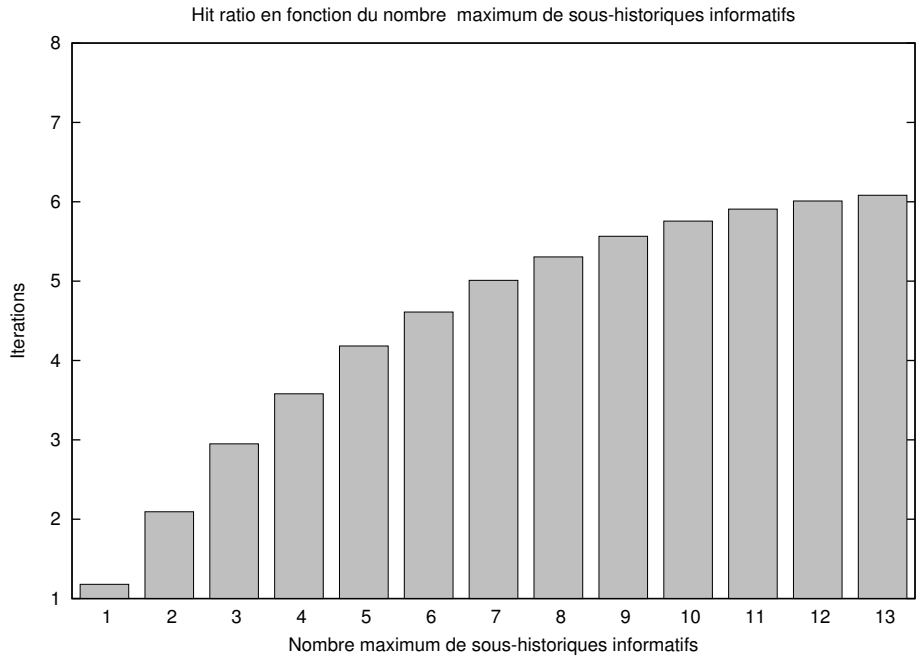

FIGURE 4 – Nombre moyen d'itérations en fonction du nombre maximum de sous-historiques informatifs sur le corpus de l'université DePaul.

linéairement. Les nombres moyens d'itérations correspondant à $i = 6$ sont respectivement de $4,6$ et $5,0$ alors qu'ils sont respectivement de $6,1$ et $7,9$ pour le SBR. Par conséquent, le nombre d'itérations est très inférieur à celui du SBR standard, avec une valeur de hit ratio similaire. Sur ces graphiques, les valeurs qui apparaissent pour $i = 1$ et $i = 13$ correspondent aux valeurs du tableau 10.

Par conséquent, le SBR peut non seulement être utilisé dans un temps limité, mais peut également fournir des recommandations ayant une précision similaire à celle fournie dans sa version standard en presque deux fois moins de temps, en utilisant la notion de sous-historiques informatifs.

2.3 Conclusion

Afin d'améliorer la rapidité des recommandations, nous avons proposé d'exploiter une caractéristique de la politique de combinaison des sous-historiques utilisée par le SBR pour calculer les recommandations selon un algorithme incrémental. Nous avons étudié ces caractéristiques expérimentalement, et avons observé que le SBR peut non seulement être utilisé dans un temps limité, mais peut également fournir des recommandations ayant une précision similaire à celle fournie dans sa version standard en utilisant la notion de sous-historiques informatifs.

3 ADAPTABILITÉ : ADAPTATION DYNAMIQUE À L'HISTORIQUE DE L'UTILISATEUR

Bien que fournissant une couverture quasi totale sur les corpus que nous avons utilisés, certains cas subsistent où le modèle SBR ne peut pas trouver de ressource à recommander. En effet, le modèle SBR fournit à la fois une grande précision, une grande résistance au bruit et une faible complexité en temps et en mémoire, mais sa couverture n'atteint pas 100% de couverture. Ce problème de couverture est résolu dans l'état de l'art en utilisant le all-k^{th}-order Markov model, dont le principe est d'appliquer des modèles de Markov d'ordre décroissant, jusqu'à ce qu'une recommandation puisse être fournie (cf. chapitre 1, section 3.3.3).

Dans cette section nous étudions l'application de ce principe au modèle SBR.

3.1 Approche proposée

Le modèle que nous proposons est appelé AKO-SBR pour All-k^{th}-Order SBR. Comme mentionné dans le chapitre 2, prendre en compte des modèles d'ordres supérieurs augmente la précision des recommandations, à condition que des données d'apprentissage suffisamment importantes soient disponibles pour que le modèle obtenu soit représentatif. Il convient donc de déterminer pour quel ordre k le modèle SBR est le plus précis, et de construire k sous-modèles d'ordres variant de 0 à k. Pendant la phase de recommandation, le sous-modèle d'ordre k est utilisé en premier, et si aucune ressource n'a pu être recommandée, alors le sous-modèle d'ordre inférieur est utilisé, et ainsi de suite, jusqu'à ce qu'une ressource puisse être recommandée

(dans le pire des cas en utilisant le sous-modèle d'ordre 0, qui fournit toujours la même liste de recommandation quel que soit l'historique).

3.2 Expérimentation

Nous nous intéressons à présent à l'étude expérimentale du modèle AKO-SBR. Nous présentons dans les tableaux 11 et 12 les valeurs de hit ratio et de couverture du modèle SBR et du modèle AKO-SBR en fonction de l'ordre k, avec une taille de fenêtre de $D = 10$.

	SBR		AKO-SBR	
	hit ratio	couv.	hit ratio	couv.
$k = 0$	31,8	100,0	31,8	100,0
$k = 1$	71,3	100,0	71,3	100,0
$k = 2$	**73,5**	**99,5**	**73,6**	**100,0**
$k = 3$	63,3	95,2	65,5	100,0
$k = 4$	57,2	78,9	61,9	100,0

TABLE 11 – Précision et couverture du modèle SBR et du modèle AKO-SBR sur le corpus du Crédit Agricole S.A.

	SBR		AKO-SBR	
	hit ratio	couv.	hit ratio	couv.
$k = 0$	22,2	100,0	22,2	100,0
$k = 1$	58,9	100,0	58,9	100,0
$k = 2$	**60,1**	**99,7**	**60,0**	**100,0**
$k = 3$	45,6	92,4	46,9	100,0
$k = 4$	37,8	63,7	41,6	100,0

TABLE 12 – Précision et couverture du modèle SBR et du modèle AKO-SBR sur le corpus de l'université DePaul.

Le tableaux 11 et 12 montrent que le hit ratio du modèle SBR et du modèle AKO-SBR ont une valeur optimale pour $k = 2$. En considérant le tableau dans son ensemble, nous pouvons remarquer que les modèles d'ordre inférieurs ne sont pas assez spécifiques et fournissent des précisions moindres ; au contraire, les modèles d'ordres supérieurs sont trop spécifiques et les données d'apprentissage ne sont pas suffisantes pour construire de modèles représentatifs, ce qui aboutit également à une précision moindre. Par conséquent, une valeur de $k = 2$ est optimale.

Un élément intéressant est que pour $k = 2$, le AKO-SBR a un hit ratio légèrement supérieur à celui du SBR sur le corpus du Crédit Agricole S.A., et légèrement inférieur sur le corpus de l'université DePaul. Cela signifie que dans le cas du premier corpus, le SBR d'ordre 1 est plus

performant sur les cas pour lesquels aucune recommandation n'a pu être fournie par le SBR d'ordre 2, et moins performant sur les autres cas. Cependant, ce phénomène ne se reproduit pas sur l'autre corpus, et étant donné le faible nombre de cas sur lesquels il porte (moins de 0,5%), il est probablement dû au hasard.

Le principe des all-k^{th}-order Markov models peut donc être appliqué au modèle SBR afin de fournir une couverture totale. De plus, le nombre de cas pour lesquels son utilisation est nécessaire est extrêmement faible, et le temps de calcul très peu augmenté.

3.3 Conclusion

Afin d'améliorer la couverture, nous avons proposé de nous inspirer du principe des all-k^{th}-order Markov model pour le modèle SBR. Le modèle obtenu, le AKO-SBR, fournit une couverture totale tout en fournissant une précision globale et un temps de calcul comparables à ceux du modèle SBR de référence.

4 CONCLUSION

La navigation Web implique un compromis entre précision, complexité et couverture. Le modèle SBR, bien que représentant une meilleure alternative que l'état de l'art, utilise un schéma de pondération analytique relativement arbitraire, a une complexité en temps qui peut être trop importante dans le cadre d'une application temps réel, et ne fournit pas 100% de couverture.

Dans ce chapitre, nous avons dans un premier temps validé le schéma de pondération en comparant le modèle SBR a des alternatives permettant des schémas de pondération plus élaborés. Le SBR fournissant des résultats similaires ou supérieurs, nous avons conclu que de par sa plus faible complexité, il reste la meilleure alternative.

Nous avons ensuite proposé d'exploiter une caractéristique de la politique de combinaison des sous-historiques du SBR pour effectuer des recommandations dans un temps limité selon un algorithme incrémental. Nous avons proposé d'utiliser le principe des sous-historiques informatifs pour fournir la même couverture que le SBR de référence tout en limitant le nombre d'itérations. Nous avons étudié les performances obtenues, et avons observé que le SBR peut non seulement être utilisé dans un temps limité, mais peut également fournir des recommandations ayant une précision similaire à celle fournie dans sa version standard en utilisant la notion de sous-historiques informatifs.

Enfin, pour améliorer la couverture, nous avons proposé de nous inspirer du principe des all-k^{th}-order Markov model au modèle SBR. Une étude expérimentale a montré que le modèle obtenu, le AKO SBR, permet de fournir des recommandations dans 100% des cas.

Chapitre 5

Alignements de séquences pour prendre en compte les navigations parallèles

La navigation Web est en constante évolution, et bien qu'ayant été largement étudiée, sa modélisation est un défi perpétuel. La technologie utilisée pour surfer sur le Web devient plus ergonomique et plus sophistiquée chaque jour, ce qui implique des changements dans le comportement habituel des utilisateurs. En particulier, les navigateurs actuels fournissent des fonctionnalités appelées *onglets* qui permettent d'inclure plusieurs pages dans la même fenêtre et de passer de l'une à l'autre, ce qui est communément appelé *tabbing*. Bien que ce type de fonctionnalités soit aujourd'hui communément utilisé par les utilisateurs du Web, très peu de travaux s'y sont intéressés.

Dans plusieurs travaux, il est supposé que les sessions contiennent des *tâches* qui peuvent être caractérisées (Jin *et al.*, 2005b ; Gündüz et Özsu, 2003). Une tâche peut être considérée comme une suite typique de ressources, qui peut avoir plusieurs variations. Par exemple, il est possible qu'un utilisateur voulant trouver une page en particulier suive une suite typique de pages menant à la page qu'il recherche, et qu'en cours de route il utilise le bouton « précédent » du navigateur, puis clique à nouveau sur le lien sur lequel il avait cliqué précédemment. Au sein d'une même session, un utilisateur peut effectuer plusieurs tâches en parallèle via les onglets, ce qui est appelé navigation parallèle. Même s'il est possible de le faire, les changements d'onglet ne sont généralement pas stockés dans les logs de navigation, et seules les requêtes HTTP et leur datation sont stockées, et quand des navigations parallèles sont effectuées, la session qui résulte consiste en une imbrication de plusieurs suites de ressources. Il est alors difficile de retrouver quelles ressources correspondent à quelles tâches.

Côté client, le tabbing peut être classé en deux catégories : le tabbing inter-sites et le tabbing intra-site. Côté serveur, seul le tabbing intra-site peut se produire. Dans le cas du tabbing inter-sites, il est facile de discerner les tâches, puisqu'il suffit de considérer les URL associées aux ressources pour en déduire à quels sites elles correspondent. Dans le cas du tabbing intra-site,

il est plus difficile de faire ce discernement. En effet, la façon habituelle de réaliser cette action est de cliquer sur un lien en utilisant le bouton du milieu de la souris, ou de maintenir la touche CTRL enfoncée en cliquant sur un lien. Étant donné que ni ces actions, ni le changement d'onglet ne sont généralement indiqués dans les logs de navigation, déterminer quelle ressource correspond à quelle tâche est un problème difficile.

Les approches de l'état de l'art ne tiennent pas compte des navigations parallèles, et se contentent habituellement de supposer qu'une session n'est composée que d'une seule tâche. Nous faisons donc l'hypothèse que si les tâches imbriquées dans une session sont extraites et que ces tâches sont prises en compte séparément, la précision des recommandations obtenue sera améliorée.

Le but de ce chapitre est d'étudier comment discriminer les différentes tâches contenues dans une session. Nous commencerons par nous intéresser aux travaux en rapport avec les navigations parallèles. Puis nous proposons un algorithme d'extraction de tâches basé sur un algorithme d'alignement de séquences. Nous proposons ensuite un modèle qui exploite les tâches extraites pour calculer les recommandations de ressources. Enfin, nous présentons une étude expérimentale de ces approches.

1 ÉTAT DE L'ART DE LA NAVIGATION PARALLÈLE

À notre connaissance, la navigation parallèle n'a jamais été étudiée du point de vue de la modélisation prédictive. Cependant quelques travaux sont en rapport avec cette problématique.

Dans (Weinreich *et al.*, 2006), une étude du comportement d'utilisateurs du Web effectuée sur une période de 195 jours est présentée. Les auteurs comparent leur étude à deux études antérieures remontant à 1995 et 1997 (Catledge et Pitkow, 1995 ; Tauscher et Greenberg, 1997). En effet, aucune autre étude similaire n'a été effectuée pendant les 9 années qui ont précédé. Les résultats montrent des changements considérables dans le comportement des utilisateurs sur ces 9 années, tels qu'une baisse de la fréquence d'utilisation du bouton « précédent », qui représentait 30% des activités de l'utilisateur au milieu des années 90, et qui ne représente plus que 15% actuellement. L'utilisation de plusieurs fenêtres [1] est passée de 1% à plus de 10%. Il est important de remarquer que dans cette étude, il est considéré qu'une nouvelle fenêtre a été ouverte uniquement quand cela est effectué via le menu contextuel. Par conséquent, le chiffre réel est plus élevé. De plus, il est mentionné que les participants de l'étude ont utilisé les onglets, mais aucun chiffre n'est fourni.

Dans (Viermetz *et al.*, 2006), un modèle appelé *clicktree* est proposé pour représenter sous forme arborescente toutes les possibilités de navigation utilisant des onglets correspondant aux sessions extraites de logs. En utilisant ce modèle, les auteurs estiment que les onglets sont utilisés entre 4% et 85% du temps par les utilisateurs, ce qui représente un intervalle important.

1. Du point de vue de la navigation parallèle, l'utilisation de plusieurs fenêtres peut être considérée comme similaire à l'utilisation de plusieurs onglets.

De plus, la manipulation de tels clicktrees implique une complexité en temps et en mémoire importantes et ne semble pas appropriée pour la recommandation.

Dans (Huang et White, 2010) une étude de grande envergure portant sur des millions d'utilisateurs fournit des chiffres précis sur la navigation Web. En particulier, on y apprend que 42,6% des navigations à l'intérieur d'un même onglet se font sans changement d'onglet. Par conséquent, pour 57,4% des navigations à l'intérieur d'un même onglet, au moins un changement d'onglet est effectué. Par conséquent, les navigations parallèles sont plus fréquentes que les navigations linéaires.

Quelques travaux sont en rapport avec la modélisation de la navigation parallèle, en ce sens qu'ils recherchent une plus grande granularité dans la modélisation des sessions. En particulier, (Jin *et al.*, 2005b) proposent un système de recommandation web capable de découvrir des motifs qui correspondent à des tâches. Les auteurs utilisent une approche appelée *probabilistic latent semantic analysis* pour caractériser les tâches de navigation. Ils utilisent ensuite un réseau bayésien pour estimer les probabilités des tâches étant donné la session active de l'utilisateur. Les recommandations sont alors calculées en utilisant un modèle d'entropie maximale dans lequel une des caractéristiques utilise un modèle de Markov d'ordre 1. Le modèle ainsi obtenu est capable de détecter les changements de tâche au sein d'une session et peut être considéré comme un modèle satisfaisant pour la navigation parallèle, car il discerne implicitement les différentes tâches.

Ce dernier modèle est donc dans le même esprit que le SBR du point de vue de la navigation parallèle, en ce sens qu'il la prend en compte de façon implicite. Cependant, dans ce chapitre, nous proposons d'aller plus loin en déterminant explicitement quelle ressource correspond à quelle tâche, afin de modéliser plus précisément les navigations, et ainsi améliorer la précision des recommandations.

2 EXTRACTION DE SESSIONS LINÉAIRES

Dans les sessions extraites des logs de navigation, aucune indication ne permet généralement de distinguer quelle ressource a été consultée sur quel onglet/fenêtre. Nous proposons dans cette section un algorithme simple capable d'extraire automatiquement cette information. Plusieurs termes doivent être définis dans un premier temps.

- Nous appelons *tâche* une suite typique de ressources pouvant avoir plusieurs variations, et correspondant implicitement à un objectif particulier. Nous faisons donc l'hypothèse que deux sessions correspondent à une même tâche si les suites de ressources qui les composent sont identiques ou fortement similaires.
- Nous appelons *session linéaire* une session dans laquelle une seule tâche est effectuée. Plusieurs sessions linéaires différentes peuvent correspondre à la même tâche, et une session linéaire ne peut être associée qu'à une seule tâche ;
- Nous appelons *session non linéaire* une session dans laquelle plusieurs tâches sont imbriquées ;
- Nous appelons *session interrompue* une session linéaire qui a été arrêtée brutalement.

– Étant données deux sessions X et Y, X est une *sous-session* de Y si X est incluse dans Y. Plus précisément, pour que X soit considérée comme une sous-session de Y, il faut soit que X soit une session linéaire et Y une session non linéaire, soit que X soit une session interrompue et Y une session quelconque.

La figure 1 montre les différents types de sessions ainsi que leurs relations. Les ronds A, B, C, D, E et F correspondent à des sessions, et chaque flèche indique qu'une session est une sous-session de la session pointée. La figure contient trois sortes de sessions :

1. Les sessions linéaires : sur la figure, cette catégorie correspond aux ronds C, D et E ;
2. Les sessions non linéaires : sur la figure, cette catégorie correspond aux ronds A et B. C, D et E sont trois sous-sessions de A, D et E sont deux sous-sessions de B ;
3. Sessions interrompue : sur la figure, cela correspond au rond F, qui est identifié comme une sous-session de la session linéaire D.

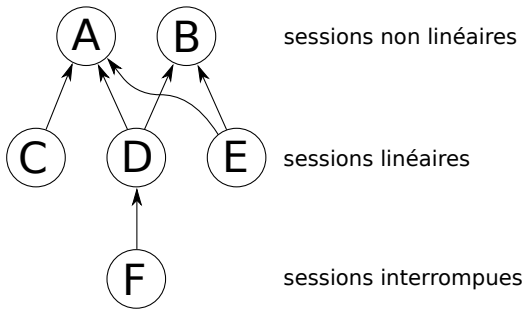

FIGURE 1 – Les différentes sortes de sessions.

Plusieurs considérations peuvent alors être mises en avant. La première est qu'un utilisateur peut avoir un comportement *a priori* en contradiction avec ces notions : il peut par exemple effectuer plusieurs tâches successives dans un même onglet ou dans une même fenêtre. Il peut également avoir ouvert plusieurs onglets dans la même fenêtre pour un même site, et pour une même tâche. Dans le premier cas, nous choisissons d'assimiler la session obtenue à une session non linéaire, dans laquelle plusieurs sessions linéaires se succèdent. Dans le second cas, notre but étant la recommandation, nous considérons qu'il n'est pas utile de savoir que plusieurs onglets ont été utilisés pour effectuer une même tâche, et choisissons d'assimiler la session à une session linéaire.

Étant données ces notions, la problématique est la détermination du type de chaque session issue des logs de navigation. Une première solution serait de supposer que si une session X est une sous-session d'une autre session Y, alors Y n'est pas une session linéaire. Cependant, il est possible que X corresponde à une session interrompue, mais que X et Y soient toutes les deux des sessions linéaires assimilables à une même tâche. Par conséquent, une contrainte plus forte doit être utilisée. Nous proposons de considérer l'hypothèse : si deux sessions X et Y ne correspondent pas à une même tâche et sont toutes les deux des sous-sessions d'une troisième

session Z, alors Z n'est pas une session linéaire (mais X et Y ne sont pas des sessions linéaires pour autant).

En utilisant cette hypothèse, on est assuré que les sessions ayant été reconnues comme des sessions non linéaires sont bien des sessions non linéaires. Cependant, les autres sessions ne sont pas toutes des sessions linéaires pour autant, mais constituent dans leur ensemble un corpus plus cohérent que l'ensemble des sessions d'origine. Par commodité d'écriture, nous parlerons tout de même d'extraction de sessions linéaires dans le reste de ce chapitre, en gardant en vue que les sessions extraites ne sont en réalité pas toutes linéaires.

En utilisant ce dernier principe, une comparaison de toutes les sessions doit être effectuée afin de supprimer toutes les sessions non linéaires chaque fois que la situation décrite ci-dessus est rencontrée. Ce processus est détaillé dans l'algorithme 2.

Données : Une liste S de sessions
Résultat : Une sous-liste de S contenant uniquement des sessions linéaires
pour chaque session X dans S **faire**
  pour chaque session Y après X dans la liste S **faire**
    si NON memeTache(X, Y) **alors**
      pour chaque session Z dans S, $Z \neq X$ et $Z \neq Y$ **faire**
        si estSousSession(X, Z) **et** estSousSession(Y, Z) **alors**
          supprimer Z de S;
        fin
      fin
    fin
  fin
fin

Algorithme 2 : Extraction des sessions linéaires.

Cet algorithme d'extraction utilise deux autres algorithmes :

- memeTache(X, Y) indique si oui ou non deux sessions X et Y sont suffisamment similaires pour être considérées comme correspondant à la même tâche. Cet algorithme est présenté dans la section 2.1 ;
- estSousSession(X, Y) indique si oui ou non X est une sous-session de Y. Cet algorithme est présenté dans la section 2.2.

2.1 Détermination des sessions correspondant à la même tâche

Nous rappelons que selon notre définition, deux sessions qui correspondent à la même tâche peuvent avoir plusieurs variations. Par exemple, étant données les sessions suivantes :

$$X = \langle \texttt{A B C D E F G H} \rangle$$
$$Y = \langle \texttt{A B C D C D E F G H} \rangle$$
$$Z = \langle \texttt{A B I J K L} \rangle$$

X et Y doivent être considérées comme correspondant à la même tâche car Y est une légère variation de X, et Z doit être considéré comme correspondant à une autre tâche que celle à laquelle correspondent X et Y, car les ressources qu'elle contient sont très différentes de celles de X et Y. Afin de mesurer la distance entre les sessions, nous proposons d'utiliser un algorithme d'alignement global, comme cela est fait dans (Gündüz et Özsu, 2003). L'algorithme utilisé dans (Gündüz et Özsu, 2003), l'algorithme FastLSA, est une version améliorée en terme de complexité en mémoire de l'algorithme de Needleman-Wunsch (Needleman et Wunsch, 1970) qui est un algorithme standard d'alignement global. Cependant, nous nous sommes rendu compte que la complexité en temps de l'algorithme FastLSA est supérieure à celui de l'algorithme de Needleman-Wunsch. Or, la complexité en mémoire n'est pas problématique pour le calcul d'alignements de sessions, puisque la taille des sessions est en général réduite. Nous avons donc utilisé la version standard de l'algorithme de Needleman-Wunsch, qui est présenté dans l'algorithme 3. Cet algorithme calcule un score d'alignement entre deux séquences, dont les bornes sont dépendantes de la taille des séquences, des pénalités des substitutions, des pénalités d des insertions, et du score des correspondances. Plus le score est élevé, plus les séquences sont similaires. Des scores négatifs sont possibles, et indiquent que les deux séquences ne sont pas similaires.

Nous considérons donc que X correspond à la même tâche que Y si le score de leur alignment α est proche de la taille de X (ou de manière équivalente Y). Un seuil t_1 doit donc être fixé, de sorte que deux sessions correspondent à la même tâche si le score de leur alignement global est supérieur à t_1% du score maximum possible. Tout comme les autres algorithmes de ce chapitre, cet algorithme est volontairement simple car il constitue une première proposition.

Données : Deux séquences X et Y de tailles m et n
Résultat : Score α de l'alignement local optimal
$F(i,0) = -i \cdot d$ pour tout $i = 0, 1, ..., m$;
$F(0,j) = -j \cdot d$ pour tout $j = 1, 2, ..., n$;
pour $i = 1$ **à** m **faire**
  pour $i = j$ **à** n **faire**

$$F(i,j) \leftarrow max \begin{cases} F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{cases} ;$$

  fin
fin
$\alpha \leftarrow F(m,n)$

Algorithme 3 : L'algorithme de Needleman-Wunsch.

Dans la figure 2 nous présentons les résultats des meilleurs alignements globaux des trois sessions X, Y et Z, avec $d = -1$ et $s(x_i, y_j) = 1$ si $x_i = y_j$, et -2 sinon. Les scores indiquent que X et Y correspondent à la même tâche, alors que Z correspond à une tâche différente.

$$\begin{array}{ll} X & \texttt{A B C D - - E F G H} \\ Y & \texttt{A B C D C D E F G H} \end{array}$$
$$\alpha = 6$$

$$\begin{array}{ll} X & \texttt{A B C D E F G H} \\ Z & \texttt{A B I J K L - -} \end{array}$$
$$\alpha = -8$$

$$\begin{array}{ll} Y & \texttt{A B C D C D E F G H} \\ Z & \texttt{A B I J K L - - - -} \end{array}$$
$$\alpha = -10$$

FIGURE 2 – Exemple d'alignements globaux.

2.2 Identification des sous-sessions

L'autre étape de notre algorithme d'extraction de sessions linéaires permet de déterminer si une session X est une sous-session d'une autre session Y. Une première manière de déterminer cela serait de simplement vérifier que chaque élément de X peut être trouvé dans Y dans le même ordre. Cependant, étant donné que deux sessions qui correspondent à la même tâche peuvent avoir plusieurs variations, une telle stratégie est trop restrictive. Nous proposons donc d'utiliser une version adaptée d'un algorithme d'alignement local, basée sur l'algorithme de Smith-Waterman (Smith et Waterman, 1981). Nous avons choisi cet algorithme car c'est un des algorithmes standard d'alignement local.

Le problème classique de l'alignement local est de trouver le meilleur alignement entre deux sous-séquences arbitraires des séquences X et Y. Notre but est un peu différent. En effet, la problématique est de trouver des imbrications de sous-sessions. Il faut déterminer si X peut être aligné avec Y de telle manière que n'importe quel nombre d'insertions puisse être appliqué entre les éléments de X sans pénaliser le score final. Nous proposons donc d'appliquer une modification simple de l'algorithme de Smith-Waterman qui consiste à ne pénaliser les insertions qu'entre les éléments de Y, et non pas entre les éléments de X. L'algorithme obtenu est donc un algorithme asymétrique, et est présenté dans l'algorithme 4.

Il y a donc deux différences entre l'algorithme de Needleman-Wunsch et l'algorithme de Smith-Waterman.

1. la valeur minimum de chaque élément de la matrice est fixée à 0. En effet, dans le cadre d'un alignement local, il est préférable de commencer un nouvel alignment si le meilleur alignement à partir d'une position donnée mène à un score négatif ;
2. le score de l'alignement ne correspond pas au dernier élément de la matrice, mais à la cellule contenant la plus grande valeur de score.

Données : Deux séquences X et Y de tailles m et n
Résultat : Score β de l'alignement local optimal
$F(i,0) = 0$ pour tout $i = 0, 1, ..., m$;
$F(0,j) = 0$ pour tout $j = 1, 1, ..., n$;
$\beta \leftarrow 0$;
pour $i = 1$ **à** m **faire**
  pour $i = j$ **à** n **faire**
    $$F(i,j) \leftarrow max \begin{cases} 0 \\ F(i-1,j-1) + s(x_i, y_j) \\ F(i-1,j) - d \\ F(i,j-1) \end{cases} ;$$
    si $F(i,j) > \beta$ **alors**
      $\beta \leftarrow F(i,j)$
    fin
  fin
fin

Algorithme 4 : Version modifiée de l'algorithme de Smith-Waterman.

Nous considérons donc que X est une sous-session de Y lorsque le score β de l'alignement local est proche de la taille de X. Un seuil t_2 doit donc être fixé, de sorte que si le score de l'alignement local est supérieur à t_2% du score maximum possible, alors X est une sous-session de Y.

Des exemples d'alignements locaux des trois précédentes sessions X, Y et Z, avec deux autres sessions T_1 et T_2 (présentées ci-dessous), avec $d = -1$ et $s(x_i, y_j) = 1$ si $x_i = y_j$, et -2 sinon, sont présentés dans la figure 3.

$$T_1 = \langle \texttt{ A B C D A B E F I J K G H L } \rangle$$
$$T_2 = \langle \texttt{ I J A B C K M N } \rangle$$

Étant donnés ces scores d'alignement, nous pouvons déduire que X et Z sont des sous-sessions de T_1. Étant donné que X et Z correspondent à des tâches différentes (information obtenue à l'étape présentée dans la section 2.1), nous pouvons déduire que T_1 n'est pas une session linéaire et est une imbrication de X et Z. Le même raisonnement peut être appliqué sur Y, Z pour T_1. Cependant, aucune conclusion ne peut être faite à propos de T_2.

2.3 Expérimentation

Nous nous intéressons à présent à l'étude expérimentale de notre algorithme d'extraction de sessions linéaires. Étant donné que nos algorithmes sont une première proposition, nous nous sommes contenté de ne les expérimenter que sur un des corpus, à savoir le corpus de l'université DePaul. Ce corpus date de 2002. À cette époque, les navigateurs offraient déjà la

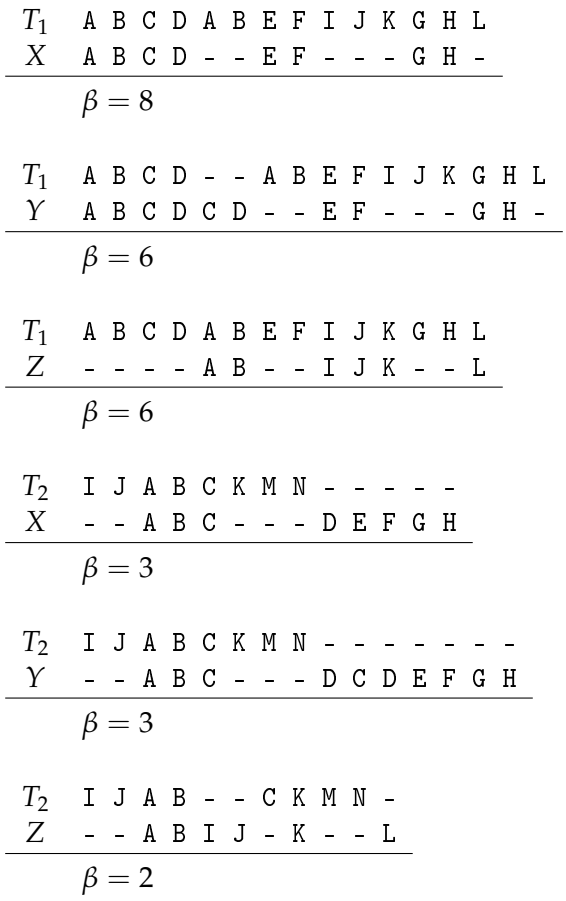

FIGURE 3 – Exemple d'alignements locaux.

possibilité d'utiliser des onglets, mais il n'est pas certain que ce type de fonctionnalité fût beaucoup utilisé. Comme (Viermetz *et al.*, 2006) l'ont montré, tenter de déterminer si des onglets ont effectivement été utilisés à partir de logs standards aboutit à une fourchette large minorée par une valeur très faible (4%). De plus, il est important de rappeler que notre algorithme est une étape préliminaire pour calculer des listes de recommandation, et n'est pas précisément destiné à estimer la part de navigations parallèles des sessions. En effet, les sessions linéaires extraites ne sont pas toutes linéaires et les sessions considérées comme non linéaires peuvent consister en une succession de tâches effectuées dans un même onglet (*cf.* introduction de cette section). Nous pouvons toutefois supposer que des navigations parallèles sont contenues dans ces sessions, et observer leur proportion en utilisant notre algorithme.

Dans le tableau 1, nous présentons le pourcentage de sessions linéaires (SL) obtenues en fonction des seuils utilisés pour la détermination des sessions correspondant à la même tâche et l'identification des sous-sessions. En utilisant des seuils variant entre 75 et 95%, le nombre de sessions linéaires extraites ne varie pas beaucoup. En effet, la plus petite quantité de ses-

sions linéaires est 46, 8% quand la plus grande est 50, 5%. Cela signifie que les deux types de sessions, linéaire et non linéaire, sont suffisamment discriminables pour qu'un intervalle important de seuils fournisse des résultats similaires. Par conséquent n'importe quel seuil dans cette fourchette peut être choisi indifféremment.

TABLE 1 – Sessions linéaires extraites

t_1	t_2	SL
75	75	46, 8%
75	80	48, 5%
75	85	48, 7%
75	90	48, 8%
80	75	49, 2%
80	80	50, 1%
80	85	50, 4%
80	90	50, 5%
85	75	49, 2%
85	80	50, 1%
85	85	50, 4%
85	90	50, 5%
90	75	49, 2%
90	80	50, 1%
90	85	50, 4%
90	90	50, 5%

Nous présentons dans la figure 4 un exemple réel d'imbrication détecté pendant le processus. Étant donnés les alignements calculés, nous pouvons déduire le scénario suivant : la première page consultée par l'utilisateur est la page d'accueil du site web (nous avons supposé cela, car la plupart des sessions commencent par cette page). Il consulte ensuite dans un nouvel onglet une autre zone du site (`/cti/advising/display.asp`...) et parcourt quelques pages relatives à cette tâche tout en restant dans ce même onglet. Puis il revient vers l'onglet où est ouverte la page d'accueil, et semble consulter dans le premier onglet une zone du site consacrée aux membres de l'université. Enfin, il retourne au second onglet, et consulte une dernière page de cette zone du site, avant de se déconnecter.

3 LE MODÈLE TABAKO

Dans cette section, nous présentons le modèle de recommandation exploitant les navigations parallèles que nous proposons, et que nous appelons TABAKO pour TAbbing-Based All-k^{th}-Order Markov model.

Id sessions	Id ressources
5068	388 317 287 285 310 416 559 476 385 284
1311	- 317 287 - 310 - - - - 284
5069	388 - - - - 416 559 476 - -

388	/news/default.asp
317	/cti/advising/login.asp
287	/cti/advising/display.asp?page=intranetnews
285	/cti/advising/display.asp?page=fctquicklist
310	/cti/advising/display.asp?tab=faculty
416	/people/
559	/people/search.asp?sort=pt
476	/people/facultyinfo.asp?id=251
385	/hyperlink/hyperspring2002/lobby.asp
284	/cti/advising/display.asp?page=facultyevl

FIGURE 4 – Exemple de sessions imbriquées

Comme mentionné dans le chapitre 2, le all-k^{th}-order Markov model est un des modèles les plus performants de l'état de l'art de la recommandation basée sur la détection de motifs. Cependant, un de ses principaux inconvénients est qu'il ne prend en compte que des séquences contiguës et ne permet pas de prendre en compte les erreurs de navigation et les navigations parallèles. Cependant, appliqué à des sessions linéaires, il représente la modélisation la plus appropriée. En effet, il fournit une bonne précision, une couverture totale, et une faible complexité en temps, en particulier en comparaison avec les motifs séquentiels discontigus. Par conséquent, nous supposons que si nous sommes capables de détecter les sessions linéaires qui ont été mélangées dans une même session et si nous ignorons les erreurs de navigation, alors le all-k^{th}-order Markov model fournira une bonne précision.

Nous proposons donc d'exploiter les algorithmes présentés dans la section précédente afin d'extraire les sessions linéaires qui ont été imbriquées, et d'appliquer ensuite le all-k^{th}-order Markov model traditionnel sur ces sessions. C'est pourquoi nous appelons le modèle obtenu TAbbing-Based All-K^{th}-Order Markov model (TABAKO). En comparaison avec le all-k^{th}-order Markov model classique, deux améliorations majeures sont fournies :

1. **Pendant la phase d'apprentissage :** Les sessions linéaires du corpus d'apprentissage sont tout d'abord extraites et stockées afin d'être utilisées à la phase de recommandation (section 2). Le modèle est ensuite construit exactement de la même manière qu'un all-k^{th}-order Markov model, mais uniquement sur les sessions linéaires du corpus. De cette manière, seules les séquences utiles sont considérées pour l'apprentissage du modèle ;
2. **Pendant la phase de recommandation :** Avant de rechercher des correspondances entre la session active de l'utilisateur et les états du modèle, les éventuelles imbrications de sous-sessions sont détectées. De cette manière, les ressources de la session correspondant

à la même tâche sont mises en commun et peuvent être utilisées pour la recherche de correspondance avec le all-k^{th}-order Markov model.

Le processus de recommandation est effectué en deux étapes : (1) la détermination de la meilleure imbrication de sous-sessions, et (2) la construction de la liste de recommandations en appliquant le all-k^{th}-order Markov model sur les sessions linéaires extraites. Ce processus est détaillé dans l'algorithme 5.

Une fois que la meilleure imbrication a été déterminée, il est possible que certaines ressources subsistent dans la session active de l'utilisateur qui ne soient liées à aucune session linéaire. Nous considérons alors ces ressources comme des erreurs de navigation et proposons de les ignorer. De cette manière, il est possible de tenir compte à la fois des navigations parallèles et du bruit. Nous présentons maintenant ces deux étapes.

Données : L'historique h de l'utilisateur actif
Un all-k^{th}-order Markov model M
Une liste de sessions linéaires L
Résultat : Une liste de recommandation R
subhistories $\leftarrow \emptyset$;
tant que $|h| \neq 0$ **et** $c \neq \emptyset$ **faire**
  $c \leftarrow$bestSubsession(h, L);
  ajouter c à *subhistories*;
  $h \leftarrow h - c$;
fin
pour chaque sous-historique c dans *subhistories* **faire**
  $R_2 \leftarrow$buildList(c, M);
  $R \leftarrow$ merge(R, R_2);
fin

Algorithme 5 : Processus de recommandation

3.1 Détermination de la meilleure imbrication

Nous proposons de déterminer la meilleure imbrication par un processus itératif :

1. Rechercher la meilleure sous-session c de l'historique h ;
2. Supprimer les éléments correspondant dans h ;
3. Réitérer sur l'historique restant, jusqu'à ce qu'aucune ressource ne reste ou qu'aucun sous-historique ne soit trouvé.

La recherche de la meilleure sous-session de l'historique h est effectuée en utilisant l'algorithme de Smith-Waterman. Dans un premier temps, pour chaque session linéaire l extraite pendant la phase d'apprentissage, chaque préfixe de l de taille variant entre $\max(|\ell| - 1, |h|)$ et

1 est extrait. En effet, comme la session de l'utilisateur est active, elle ne peut être comparée aux sessions linéaires entières. Par exemple, un utilisateur peut n'avoir parcouru que 4 ressources correspondant à une tâche qui implique habituellement 10 ressources. Enfin, la sous-session ayant le meilleur score est retournée et ajoutée à l'ensemble des sous-historiques. Ce processus est détaillé dans l'algorithme 6.

Données : L'historique h de l'utilisateur actif
Une liste de sessions linéaires L
Résultat : Les sous-sessions s qui ont les meilleures correspondances avec h
$C \leftarrow \emptyset$;
pour chaque session linéaire l de L **faire**
  pour $i \leftarrow \max(|\ell| - 1, |h|)$ **à** 1 **faire**
    $\ell_i \leftarrow$substring(ℓ, i);
    $(c, \beta) \leftarrow$Smith-Waterman(ℓ_i, h);
    ajouter (c, β) à L;
  fin
fin
$i \leftarrow \arg\max L$;
retourner $L(i)$

Algorithme 6 : Détermination de la meilleure sous-session

3.2 Construction des listes de recommandation

Après que la meilleure imbrication ait été déterminée, les sous-sessions correspondantes sont extraites de la session active. Pour chaque sous-session, une liste de recommandation est alors calculée en utilisant le all-k^{th}-order Markov model. Les sous-listes obtenues sont enfin combinées selon la politique suivante : les recommandations issues de la sous-session qui a obtenu le meilleur score selon l'algorithme de Smith-Waterman sont placées en haut de la liste. Les autres recommandations sont concaténées à la liste par ordre décroissant du score de la sous-session correspondante, sauf quand elles figurent déjà dans la liste.

3.3 Expérimentation

Nous nous intéressons à présent aux résultats fournis par le modèle TABAKO en comparaison de l'état de l'art.

Étant donné la faible variation en fonction des seuils utilisés pour l'extraction des sessions similaires, nous avons choisi un couple de seuils arbitraire dans notre expérimentation. Ces seuils sont $t_1 = 75\%$ et $t_2 = 75\%$. Les sessions linéaires obtenues avec ces seuils représentent $46,8\%$ du total de sessions et ont une taille moyenne de $3,91$ alors que les sessions non linéaires ont une taille moyenne de $8,45$. Dans la figure 5, les résultats obtenus par le modèle TABAKO et le all-k^{th}-order Markov model (AKO) sont présentés. Des listes de recommandation de taille

variant de 1 à 10 sont utilisées. Comme nous pouvons le voir, le modèle TABAKO a une précision très supérieure à celle du all-k^{th}-order Markov model, quelle que soit la taille de la liste de recommandation. En particulier, pour des listes de taille 10, le all-k^{th}-order Markov model fournit un hit ratio de 42,9 alors que le modèle TABAKO fournit un hit ratio de 54,7, ce qui représente une amélioration de 27,5%.

Par conséquent, l'extraction de sessions linéaires et leur exploitation avec le modèle TABAKO mène à une amélioration de la précision des recommandations en comparaison de all-k^{th}-order Markov model. Cependant, nous rappelons que sur ce même corpus, le modèle SBR, en tenant compte de manière implicite de la navigation parallèle, fournissait un hit ratio de 60,1, ce qui est encore supérieur au résultat obtenu par le modèle TABAKO. Cependant, les algorithmes utilisés pour l'extraction des sessions linéaires et la recommandation de ressources ont été choisis dans le cadre d'une première proposition. Le modèle TABAKO représente donc un bon premier pas dans la prise en compte explicite des navigations parallèles, mais nous envisageons de l'améliorer.

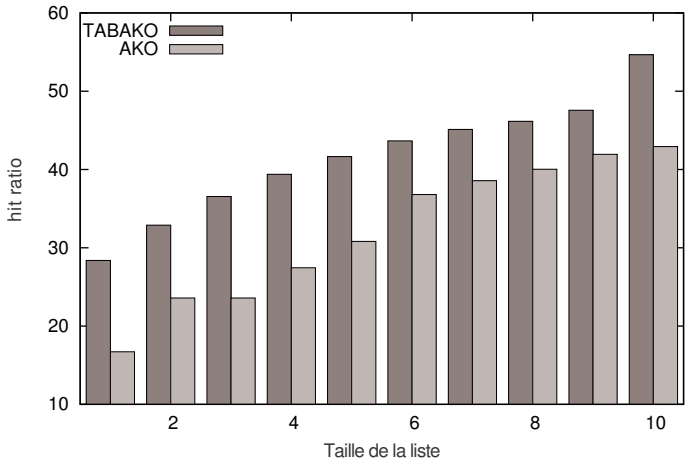

FIGURE 5 – hit ratio du modèle TABAKO et du all-k^{th}-order Markov model.

4 CONCLUSION

Dans ce chapitre, nous nous sommes intéressés à la prise en compte explicite de la navigation parallèle afin d'améliorer la précision des recommandations. Nous avons proposé un algorithme permettant de discriminer les sessions linéaires des sessions non linéaires. Cet algorithme exploite les alignements globaux et locaux de séquence. Nous avons ensuite proposé un nouveau modèle de recommandation, le modèle TABAKO. Ce modèle est basé sur le all-k^{th}-order Markov model et exploite l'algorithme d'extraction de sessions linéaires.

Les résultats expérimentaux montrent que notre algorithme a une grande stabilité dans l'extraction de sessions linéaires. De plus, le modèle TABAKO fournit une précision supérieure à

celle du all-k^{th}-order Markov model standard, mais inférieure à celle du modèle SBR. Cependant, les algorithmes utilisés par le modèle TABAKO sont simples et constituent une première proposition, largement améliorable.

Conclusion et perspectives

1 CONCLUSION

Dans cette thèse, nous nous sommes intéressé à la problématique générale des systèmes de recommandation. Nous avons dans un premier temps présenté le domaine selon une séparation en deux approches principales : la recommandation basée sur le contenu et la recommandation basée sur les usages. Cette présentation nous a permis, entre autres, de constater que la prise en compte de la séquentialité des données peut améliorer la qualité des recommandations.

Partant de ce constat, nous avons décidé de nous focaliser sur la prise en compte de la séquentialité dans les recommandations. L'intérêt de la prise en compte de cette notion étant dépendante du type de données, nous avons décidé de nous intéresser à la navigation Web, car ce type de données est régi par des contraintes séquentielles qui sont déterminantes des intentions des utilisateurs. Nous avons alors fait un parallèle avec la modélisation statistique du langage naturel, et avons proposé d'adapter les modèles statistiques de langage aux particularités de la navigation Web. En particulier, les sessions de navigation ont la particularité de contenir du bruit, et peuvent consister en des imbrications de navigations parallèles.

Nous avons donc dans un premier temps proposé un nouveau modèle de recommandation pour la navigation Web qui est inspiré de la modélisation statistique du langage. Ce modèle, le modèle Skipping Based Recommender (SBR), est basé sur le concept de skipping, qui consiste à considérer des n-grammes non contigus dans une fenêtre de taille fixe. Le skipping permet de fournir une bonne résistance au bruit et une prise en compte implicite des navigations parallèles. Dans ce cadre, nous avons proposé une nouvelle variante de skipping induisant une complexité en temps linéaire et permettant la prise en compte des navigations parallèles, que nous avons appelée variante du skipping multinavigationnel. Nous avons également proposé un nouveau schéma de pondération selon une décroissance exponentielle prenant en compte la distance entre tous les éléments des n-grammes, que nous avons appelé décroissance exponentielle multiple. Nous avons montré que le modèle SBR a une plus faible complexité en temps et en mémoire que les modèles de l'état de l'art et avons confirmé expérimentalement cette propriété. Nos expérimentations montrent plus généralement que le modèle SBR constitue une meilleure alternative en ce qui concerne le compromis entre précision, complexité et couverture, ainsi que la résistance au bruit. Ces expérimentations confirment également que la vari-

ante de skipping et le schéma de pondération que nous proposons constituent les meilleures alternatives.

Nous avons alors poussé plus loin notre recherche d'amélioration de précision, de complexité et de couverture. Pour améliorer la précision, nous avons proposé différentes alternatives au modèle SBR permettant un apprentissage de jeux de pondérations plus élaborés ou optimaux en utilisant l'algorithme EM. Dans ce cadre, nos expérimentations ont mis en avant deux points : (1) les n-grammes ne semblent pas avoir de configuration de prédilection à exploiter pour améliorer la précision des recommandations, (2) la précision des alternatives plus élaborées que nous avons proposées est similaire à celle du SBR. De par sa plus grande simplicité conceptuelle, le modèle SBR reste donc la meilleure alternative.

La rapidité des recommandations est un enjeu important, notamment dans le cadre du Web mobile. Afin de réduire le temps de calcul des recommandations, nous avons proposé de tirer profit d'une caractéristique de notre politique de combinaison des recommandations. Cette politique permet en effet de calculer les recommandations selon un algorithme incrémental : les recommandations sont calculées en incorporant un à un des sous-historiques de plus en plus distants tant que le temps imparti n'est pas dépassé. Limiter le nombre de sous-historiques réduit le temps de calcul, mais peut également diminuer la précision et la couverture. Afin de conserver la couverture du modèle SBR tout en utilisant ce principe, nous avons introduit le concept de sous-historiques informatifs. Dans nos expérimentations, nous avons constaté que le SBR peut non seulement être utilisé dans un temps limité, mais peut également fournir des recommandations ayant une précision similaire à celle fournie dans sa version standard en presque deux fois moins de temps en utilisant la notion de sous-historiques informatifs.

La couverture du modèle SBR, bien que quasiment totale, n'atteint pas les 100%. Par conséquent, nous nous sommes employés à maximiser le nombre de cas pour lesquels une recommandation est fournie. Dans ce cadre, nous avons proposé de modifier le modèle SBR en s'inspirant du principe des all-k^{th}-order Markov model. Nous avons alors montré expérimentalement que ce modèle fournit une couverture totale tout en induisant une précision globale comparable à celle du modèle SBR de référence.

Notre dernière contribution a porté sur les navigations parallèles, un phénomène récent du domaine de la navigation Web. Nous avons dans un premier temps proposé un algorithme d'extraction de sessions linéaires destiné à être exploité pour le calcul de recommandations. Cet algorithme est basé sur des algorithmes d'alignement global et local de séquences. Nous avons ensuite proposé un nouveau modèle de recommandation appelé TABAKO pour Tabbing-Based All-k^{th}-Order Markov model qui exploite l'algorithme précédent pour calculer les recommandations. Une première expérimentation a montré que notre algorithme permet d'améliorer la précision des recommandations de façon significative en comparaison du all-k^{th}order Markov model, mais que cette précision reste inférieure à celle fournie par le modèle SBR.

2 PERSPECTIVES

■ SBR : Prédiction de ressources utiles

Les perspectives d'amélioration du modèle SBR du point de vue des critères habituels que sont la précision, la complexité et la couverture, sont assez limitées :

- Les alternatives permettant une modélisation plus fine que nous avons proposées dans cette thèse ont fourni des résultats similaires à ceux du modèle SBR ;
- Nous avons proposé une adaptation de notre modèle permettant de fournir des recommandations ayant une précision similaire dans un temps limité ;
- Nous avons proposé de s'inspirer du principe des all-k^{th}-order Markov models pour fournir une couverture totale tout en induisant une précision globale comparable.

Par conséquent, poursuivre ce travail semble impliquer de sortir du cadre habituel d'évaluation de ce type de modèle. Dans cette optique, une possibilité réside dans le constat mis en avant dans la section 3.4 du chapitre 2 : se baser uniquement sur une prédiction de ressources pour calculer des recommandations, comme le font la plupart des systèmes de recommandation Web, peut paraître inapproprié. En effet, selon cette stratégie, de simples probabilités conditionnelles sont utilisées pour ordonner les éléments de la liste de recommandation. Or, dans une telle configuration, les ressources les plus fréquentes sont plus susceptibles d'être recommandées, même si elles peuvent ne pas être d'une grande pertinence pour un utilisateur. Dans cette même section, nous mettions en avant l'information mutuelle, qui permet de limiter l'importance des ressources fréquentes. De façon plus générale, nous projetons de rechercher d'autres critères permettant de valoriser les ressources pertinentes.

Quand bien même un critère optimal serait utilisé, évaluer combien il est adapté en comparaison de simples probabilités conditionnelles n'est pas un problème aisé. Un premier travail utilisant l'information mutuelle a été effectué dans cette optique, dans lequel nous avons conduit un certain nombre d'expérimentations basées sur des notations implicites qui n'ont pas fourni de résultats probants. Cependant la solution des notations implicites est fortement biaisée par le choix des critères utilisés. Afin de pouvoir mesurer effectivement la satisfaction des utilisateurs, et ainsi observer l'utilité effective de la recommandation basée sur des probabilités conditionnelles et sur l'information mutuelle, nous envisageons donc de nous consacrer à la récupération d'appréciations explicites comme cela est fait par exemple dans (Fox *et al.*, 2005) pour la recherche d'information.

■ TABAKO : Approfondissement de la prise en compte des navigations parallèles

Les algorithmes que nous avons proposés dans le chapitre 5 dans le but de prendre en compte les navigations parallèles sont une première proposition. Nous projetons donc d'améliorer les deux algorithmes d'extraction des sessions linéaires et de calcul des recommandations du modèle TABAKO.

En particulier, nous pensons que le concept de tâches sur lequel nous nous sommes basés pour l'extraction de sessions linéaires peut être caractérisé de façon plus élaborée. En effet,

dans notre algorithme, nous considérons que deux sessions correspondent à la même tâche si ces deux sessions sont similaires du point de vue d'un alignement de séquences. Or, cette définition peut paraître réductrice, et l'on peut envisager l'utilisation d'une définition plus générale dans laquelle deux sessions non similaires du point de vue des alignements de séquences correspondent tout de même à une même tâche.

Un exemple de concept pouvant être utilisé dans cette optique est le concept de *maximal forward reference* (MFR) de (Chen *et al.*, 1996), ce qui pourrait se traduire par « référence maximale vers l'avant », et qui a pour but de filtrer les retours en arrière lors des navigations. En effet, quand un utilisateur navigue sur un site Web, il peut, à l'aide du bouton « précédent » du navigateur, retourner vers une page consultée antérieurement, et effectuer un autre parcours à partir de cette page. Ainsi, le véritable point de départ de ce second parcours est la première page consultée de la session, puis la seconde et ainsi de suite, jusqu'à tomber sur le moment où la page sur laquelle l'utilisateur est retourné pour effectuer ce parcours a été consultée pour la première fois. Nous pensons que ce type de concepts représente une définition plus précise du concept de tâches, et que leur utilisation permettrait de caractériser plus précisément les sessions linéaires, et donc de détecter plus précisément les navigations parallèles. Le problématique porterait donc essentiellement sur la manière dont ces deux approches peuvent être efficacement combinées.

Nous projetons également d'étendre le modèle TABAKO en utilisant un algorithme de clustering de sessions. Plusieurs algorithmes existent dans l'état de l'art et peuvent être utilisés afin de regrouper les sessions similaires tout en utilisant nos algorithmes (Gündüz et Özsu, 2003 ; Wang et Zaïane, 2002). Par exemple, des clusters de sessions peuvent être hiérarchisés en une structure arborescente dans laquelle un lien signifie que les sessions d'un cluster correspondent à des sous-navigations des sessions du cluster pointé. Une telle structure pourrait diminuer le temps de calcul et améliorer la qualité des recommandations. Ce projet soulève à lui seul de nombreuses questions, telles que la définition précise du concept de hiérarchie entre les clusters et la détermination de la meilleure stratégie d'utilisation de la structure arborescente résultante.

Une autre perspective réside dans l'amélioration du calcul des recommandations. En effet, l'algorithme du modèle TABAKO exploite prioritairement la sous-session linéaire qui a obtenu le meilleur score d'alignement. Cet algorithme peut donc être amélioré selon deux aspects : la notion de meilleure sous-séquence et la manière dont les recommandations des différentes sessions linéaires contenues dans l'historique de l'utilisateur sont combinées. Une première piste est de déterminer quelle prochaine tâche sera le plus probablement effectuée par l'utilisateur, dans l'esprit du travail de (Yeong *et al.*, 2005), présenté dans la section 4.3.2 du chapitre 1.

Une dernière perspective concernant ce travail est d'évaluer plus précisément la capacité de notre d'algorithme d'extraction de sessions linéaires à détecter les imbrications de navigations sur un corpus où les évènements de changement de fenêtres et d'onglets sont spécifiés.

■ Temps écoulé entre les consultations

L'ordre chronologique ne représente qu'une forme particulière de contexte temporel. D'autres formes de contexte temporel sont la périodicité et le temps écoulé entre les consultations. Nous

projetons de nous intéresser à cette dernière forme de contexte temporel. En effet, plutôt que de se contenter d'observer qu'un utilisateur a consulté la page A puis la page B, il est possible de considérer qu'il a passé τ_1 secondes sur la page A et τ_2 secondes sur la page B. Un intérêt particulier de ce critère est qu'il permet une évaluation implicite du niveau d'expertise de l'utilisateur : si la page A consiste en une explication générale de concepts de base, un utilisateur novice passera plus de temps à la lire qu'un utilisateurs expert. Or, l'intention d'un utilisateur novice peut être différente de celle d'un utilisateur expert ayant consulté successivement la même suite de pages. Ainsi, une même suite de ressources $\langle A, B \rangle$ peut correspondre à une intention différente selon le temps passé sur chacune de ces ressources.

Plusieurs travaux peuvent être rapprochés de cette considération. En particulier, les travaux liés à l'attribution de notes implicites (Chan, 1999 ; Fox *et al.*, 2005 ; Castagnos, 2008 ; Esslimani *et al.*, 2009). En particulier, (Fox *et al.*, 2005) montrent que le critère du temps passé sur une page est un des critères qui a la plus forte corrélation avec l'appréciation effective de l'utilisateur, ce qui plaide en faveur de l'importance de la prise en compte du temps écoulé entre deux consultations. Notre perspective est différente, car elle ne consiste pas forcément à estimer une appréciation implicite (même si l'on peut considérer que cela est inclus dans notre perspective), mais plus généralement à allier l'information du temps écoulé entre les consultations à l'ordre chronologique.

Un autre travail se rapprochant de cette idée est celui de (Gündüz et Özsu, 2003), où le temps passé sur les pages est utilisé pour calculer le score d'alignement de séquences dans l'optique d'un clustering de sessions. Ce travail est donc un peu plus en rapport avec la perspective que nous proposons, mais sa problématique reste très différente. À notre connaissance, la problématique de la prise en compte à la fois de l'ordre chronologique et du temps écoulé entre les consultations n'a jamais été précisément explorée.

Dans le cadre de modèles basés sur la détection de motifs séquentiels, il suffit de rechercher des correspondances entre les antécédents du modèle et l'historique de l'utilisateur. En rajoutant le critère de temps écoulé, deux motifs identiques peuvent correspondre à des contextes différents. Par conséquent, cette perspective soulève de nombreuses questions, dont la principale est la recherche d'une technique efficace permettant la combinaison de ces deux critères. En particulier, est-il préférable d'utiliser une mesure de similarité basée uniquement sur la durée entre les consultations des motifs identiques, ou bien de répartir ces durées en classes (*e.g.* longue durée et courte durée) et construire des motifs de paires (ressource, durée) ?

Publications

G. Bonnin, A. Brun et A. Boyer. Collaborative Filtering inpired from Language Modeling. *First International Workshop on Recommender Systems and Personalized Retrieval (RSPR)*, pages 192–197, 2008.

G. Bonnin, A. Brun et A. Boyer. Using Skipping for Sequence-Based Collaborative Filtering. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI)*, pages 775–779, 2008.

G. Bonnin, A. Brun et A. Boyer. A Low-Order Markov Model integrating Long-Distance Histories for Collaborative Recommender Systems. *Proceedings of the 13th international conference on Intelligent user interfaces (IUI)*, pages 57–66, 2009.

G. Bonnin, A. Brun et A. Boyer. Renforcement des modèles probabilistes en utilisant l'Information Mutuelle pour des Recommandations contextualisées. *7ème colloque du chapitre français de l'ISKO*, pages 147–153, 2009.

G. Bonnin, A. Brun et A. Boyer. Skipping-Based Collaborative Recommendations inspired from Statistical Language Modeling. *Web Intelligence and Intelligent Agents*, INTECH, pages 263–288, 2010.

G. Bonnin, A. Brun et A. Boyer. Detecting Parallel Browsing to Improve Web Predictive Modeling. *International Conference on Knowledge Discovery and Information Retrieval (KDIR)*, 2010.

G. Bonnin, A. Brun et A. Boyer. Towards Tabbing Aware Recommendations. *International Conference on Intelligent Interactive Technologies and Multimedia (IITM)*, 2010.

A. Brun, G. Bonnin et A. Boyer. History dependent Recommender Systems based on Partial Matching. *First and Seventeenth International Conference on User Modeling, Adaptation, and Personalization (UMAP)*, pages 343–348, 2009.

Bibliographie

J. ABERNETHY, F. BACH, T. EVGENIOU et J.-P. VERT : A New Approach to Collaborative Filtering : Operator Estimation with Spectral Regularization. *The Journal of Machine Learning Research*, 10:803–826, 2009. ISSN 1532-4435.

G. ADOMAVICIUS, R. SANKARANARAYANAN, S. SEN et A. TUZHILIN : Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23(1):103–145, 2005.

G. ADOMAVICIUS et A. TUZHILIN : Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

R. AGRAWAL, T. IMIELIŃSKI et A. SWAMI : Mining Association Rules between Sets of Items in Large Databases. *In SIGMOD '93 : Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, 1993.

R. AGRAWAL et R. SRIKANT : Mining Sequential Patterns. *In ICDE'95 : Proceedings of the International Conference on Data Engineering*, pages 3–14, 1995.

J. ALPERT et N. HAJAJ : We knew the web was big... `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`, 2008.

K. ALTHOFF : Case-Based Reasoning. volume 1, pages 549–588, 2001.

R. BAEZA-YATES et B. RIBEIRO-NETO : *Modern information retrieval*. Addison-Wesley Reading, MA, 1999.

M. BALABANOVIĆ et Y. SHOHAM : Fab : Content-Based, Collaborative Recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

Michele BANKO et Eric BRILL : Mitigating the paucity-of-data problem : Exploring the effect of training corpus size on classifier performance for natural language processing. *In Human Language Technology Conference (HLT)*, 2001.

R. BAYARDO JR : Efficiently Mining Long Patterns from Databases. *ACM Sigmod Record*, 27 (2):85–93, 1998.

R. BELL, Y. KOREN et C. VOLINSKY : Modeling Relationships at Multiple Scales to Improve Accuracy of Large Recommender Systems. *In KDD '07 : Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7.

J. BENNETT et S. LANNING : The Netflix Prize. *In KDD Cup and Workshop in conjunction with KDD*, pages 3–6, 2007.

Jose BORGES et Mark LEVENE : Generating dynamic higher-order Markov models in web usage mining. 2005.

Anne BOYER et Armelle BRUN : Natural Language Processing for Usage Based Indexing of Web Resources. *In 29th European Conference on Information Retrieval (ECIR)*, volume 4425 de *Lecture Notes in Computer Science*, pages 517–524, Rome, Italy, 2007. Fondazione Ugo Bordoni, Springer Berlin / Heidelberg.

J. BREESE, D. HECKERMAN et C. KADIE : Empirical Analysis of Predictive Algorithms for Collaborative Filtering. pages 43–52, 1998.

S. BRIN et L. PAGE : The Anatomy of a Large-Scale Hhypertextual Web Search Engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

A. BRUN et A. BOYER : Du e-commerce au m-commerce : vers une Recommandation Incrémentale. *In Proceedings of the 7th COnférence en Recherche d'Information et Applications (CORIA)*, 2010.

D. BURDICK, M. CALIMLIM et J. GEHRKE : Mafia : A Maximal Frequent Itemset Algorithm for Transactional Databases. *In Proceedings of the International Conference on Data Engineering*, pages 443–452, 2001.

R. BURKE : Hybrid Recommender Systems : Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002. ISSN 0924-1868.

R. BURKE : Hybrid Web Recommender Systems. pages 377–408, 2007.

S. CASTAGNOS : *Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information*. Thèse de doctorat, Université Nancy 2, 2008.

S. CASTAGNOS, N. JONES et P. PU : Eye-Tracking Product Recommenders' Usage. *In In proceedings of the 4th ACM Conference on Recommender Systems*, 2010.

L. CATLEDGE et J. PITKOW : Characterizing Browsing Strategies in the World-Wide Web. *Comput. Netw. ISDN Syst.*, 27(6):1065–1073, 1995.

P. CHAN : A Non-Invasive Learning Approach to Building Web User Profiles. *In KDD-99 Workshop on Web Usage Analysis and User Profiling*, 1999.

P. CHEESEMAN et J. STUTZ : Bayesian Classification (AutoClass) : Theory and Results. pages 153–180, 1996.

M. CHEN, J. PARK et P.S. YU : Data Mining for Path Traversal Patterns in a Web Environment. *In Proceedings of the 16th International Conference on Distributed Computing Systems*, pages 385–392. IEEE Computer Society, 1996.

S. CHEN et J. GOODMAN : An Empirical Study of Smoothing Techniques for Language Modeling. *Computer Speech and Language*, 1999.

Y. CHEN et K. CHAN : Extended multi-word trigger pair language model using data mining technique. *In IEEE International Conference on Systems, Man and Cybernetics*, pages 262–267, 2003.

E. CHI, J. PITKOW, J. MACKINLAY, P. PIROLLI, R. GOSSWEILER et S. CARD : Visualizing the Evolution of Web Ecologies. *In CHI '98 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 400–407, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-30987-4.

D. CHICKERING, D. HECKERMAN et C. MEEK : A Bayesian Approach to Learning Bayesian Networks with Local Structure. *In In Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1997.

M. CLAYPOOL, P. LE, M. WASED et D. BROWN : Implicit Interest Indicators. *In Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40. ACM, 2001.

R. COOLEY, B. MOBASHER et J. SRIVASTAVA : Web Mining : Information and Pattern Discovery on the World Wide Web. *In ICTAI*. Published by the IEEE Computer Society, 1997.

A. DAS, M. DATAR, A. GARG et S. RAJARAM : Google News Personalization : Scalable Online Collaborative Filtering. *WWW'07 : Proceedings of the 16th International Conference on World Wide Web*, pages 271–280, 2007.

A. DEMPSTER, N. LAIRD et D. RUBIN : Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

M. DESHPANDE et G. KARYPIS : Selective Markov Models for Predicting Web Page Accesses. *Transactions on Internet Technology*, 4(2):163–184, 2004. ISSN 1533-5399.

Magdalini EIRINAKI et Michalis VAZIRGIANNIS : Web Site Personalization Based on Link Analysis and Navigational Patterns. *Transactions on Internet Technology*, 7(4):21, 2007.

J. ELLENBERG : This Psychologist Might Outsmart the Math Brains Competing for the Netflix Prize. *Wired Magazine*, 2008.

I. ESSLIMANI, A. BRUN et A. BOYER : Enhancing Collaborative Filtering by Frequent Usage Patterns. *In First International Workshop on Recommender Systems and Personalized Retrieval*, pages 180–185, 2008.

I. ESSLIMANI, A. BRUN et A. BOYER : A Collaborative Filtering approach combining Clustering and Navigational based correlations. *In Web Information Systems and Technologies*, pages 364–369, 2009.

M. ESTER, H. KRIEGEL, S. JÖRG et X. XU : A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

M. FLEISCHMAN, E. HOVY et A. ECHIHABI : Offline Strategies for Online Question Answering : Answering Questions Before They Are Asked. *In* Erhard HINRICHS et Dan ROTH, éditeurs : *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 1–7, 2003.

S. FOX, K. KARNAWAT, M. MYDLAND, S. DUMAIS et T. WHITE : Evaluating Implicit Measures to Improve Web Search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.

N. FRIEDMAN, D. GEIGER et M. GOLDSZMIDT : Bayesian Network Classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

D. GOLDBERG, D. NICHOLS, B. OKI et D. TERRY : Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, pages 61–70, 1992.

J. GOODMAN : A bit of progress in Language Modeling (Extended Version). Rapport technique, 2001.

M. GRČAR, D. MLADENIČ, B. FORTUNA et M. GROBELNIK : Data Sparsity Issues in the Collaborative Filtering Framework. *Advances in Web Mining and Web Usage Analysis*, pages 58–76, 2006.

S. GÜNDÜZ et M. ÖZSU : A Web Page Prediction Model Based on Click-Stream Tree Representation of User Behavior. *In KDD '03 : Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–540, New York, NY, USA, 2003. ACM.

C. HALL : *L'A.B.C de la psychologie freudienne*. Paris, Éditions Montaigne, 1957.

J. HAN et M. KAMBER : *Data Mining : Concepts and Techniques (Second Edition)*. Morgan Kaufmann, second édition, 2006.

J. HAN, J. PEI et Y. YIN : Mining Frequent Patterns without Candidate Generation. *In SIGMOD '00 : Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 1–12, New York, NY, USA, 2000. ACM.

D. HECKERMAN, D. CHICKERING, C. MEEK, R. ROUNTHWAITE et C. KADIE : Dependency Networks for Inference, Collaborative Filtering, and Data Visualization. *J. Mach. Learn. Res.*, 1:49–75, 2001.

D. HECKERMAN, D. MAXWELL, C. MEEK, R. ROUNTHWAITE et C. KADIE : Dependency Networks for Inference, Collaborative Filtering, and Data Visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

K. HORNIK : Some New Results on Neural Network Approximation. *Neural Networks*, 6(8): 1069–1072, 1993.

B. HUANG et T. JEBARA : Collaborative Filtering via Rating Concentration. *In Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 334–341, 2010.

C. HUANG et W. HUANG : Handling Sequential Pattern Decay : Developing a Two-Stage Collaborative Recommender System. *Electronic Commerce Research and Applications*, 8(3):117–129, 2009.

J. HUANG et R. WHITE : Parallel Browsing Behavior on the Web. *In 21st ACM Conference on Hypertext and Hypermedia (Hypertext 2010)*, pages 13–17, 2010.

S. HUANG : Comparison of Utility-Based Recommendation Methods. *In Pacific Asia Conference on Information Systems*, pages 1–12, 2008.

X. HUANG, F. ALLEVA, M. HWANG et R. ROSENFELD : An Overview of the SPHINX-II Speech Recognition System. *In HLT '93 : Proceedings of the workshop on Human Language Technology*, pages 81–86, Morristown, NJ, USA, 1993. Association for Computational Linguistics. ISBN 1-55860-324-7.

U. HÖLZLE : Energy and the Internet. `http://googleblog.blogspot.com/2009/05/energy-and-internet.html`, 2009.

C. JACQUEMIN, J. KLAVANS et E. TZOUKERMANN : Expansion of Multi-Word Terms for Indexing and Retrieval Using Morphology and Syntax. *In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 24–31, 1997.

X. JIN, B. MOBASHER et Y. ZHOU : A Web Recommendation System Based on Maximum Entropy. *In Proceedings of the International Conference on Information Theory : Coding and Computing*, pages 213–218, 2005a.

X. JIN, Y. ZHOU et B. MOBASHER : Task-oriented Web User Modeling for Recommendation. *In Proceedings of the 10th International Conference on User Modeling (UM)*, pages 109–118, 2005b.

A. JINHA : Article 50 Million : An Estimate of the Number of Scholarly Articles in Existence. *Learned Publishing*, 23:258–263, 2010.

R. JOULE et J. BEAUVOIS : *Petit traité de manipulation à l'usage des honnêtes gens*. Presses Universitaires de Grenoble, 1987.

B KALMAN et S. KWASNY : Why tanh : Choosing a Sigmoidal Function. *In Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 4, pages 578–581, 1992.

C. KIM et J. KIM : A recommendation algorithm using multi-level association rules. *In WI '03 : Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, pages 524–527, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1932-6.

D. KIM, V. ATLURI, M. BIEBER, N. ADAM et Y. YESHA : A Clickstream-Based Collaborative Filtering Personalization Model : Towards a Better Performance. *In WIDM '04 : Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 88–95, New York, NY, USA, 2004. ACM.

J. KLEINBERG : Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46 (5):604–632, 1999. ISSN 0004-5411.

T. KOHONEN : The Self-Organizing Map. volume 78, pages 1464–1480, 1990.

Y. KOREN : Collaborative Filtering with Temporal Dynamics. *In KDD '09 : Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9.

T. KROEGER, D. LONG et J. MOGUL : Exploring the Bounds of Web Latency Reduction from Caching and Prefetching. *In USITS'97 : Proceedings of the USENIX Symposium on Internet Technologies and Systems on USENIX Symposium on Internet Technologies and Systems*, pages 2–11, Berkeley, CA, USA, 1997. USENIX Association.

C. KWOK, O. ETZIONI et D. WELD : Scaling Question Answering to the Web. *ACM Transactions on Information and System Security*, 19(3):242–262, 2001. ISSN 1046-8188.

S. LAMPRIER : *Vers la conception de documents composites : Extraction et organisation de l'information pertinente*. Thèse de doctorat, Université d'Angers, Laboratoire d'étude et de Recherche en Informatique d'Angers, 2008.

K. LANG : Newsweeder : Learning to Filter Netnews. *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

J. LI et O. ZAIANE : Combining Usage, Content, and Structure Data to Improve Web Site Recommendation. *In Proceedings of the 5th International Conference on Electronic Commerce and Web Technologies (EC-Web)*, 2004.

G. LINDEN, B. SMITH et J. YORK : Amazon.com Recommendations : Item-to-Item Collaborative Filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

F. LOUSAME et E. SÁNCHEZ : A Taxonomy of Collaborative-Based Recommender Systems. *Web Personalization in Intelligent Environments*, pages 81–117, 2009.

L. LU, M. DUNHAM et Y. MENG : Mining Significant Usage Patterns from Clickstream Data. *In 7th International Workshop on Knowledge Discovery on the Web*, pages 1–17, 2005.

J. MACQUEEN : Some Methods for Classification and Analysis of MultiVariate Observations. *In* L. M. Le CAM et J. NEYMAN, éditeurs : *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

A. MARKOV : Extension of the Limit Theorems of Probability Theory to a Sum of Variables Connected in a Chain. *In* R. HOWARD, éditeur : *Dynamic Probabilistic Systems (Volume I : Markov Models)*, chapitre Appendix B, pages 552–577. John Wiley & Sons, Inc., New York City, 1971.

F. MASSEGLIA, F. CATHALA et P. PONCELET : The PSP Approach for Mining Sequential Patterns. *Principles of Data Mining and Knowledge Discovery*, pages 176–184, 1998.

B. MILLER, I. ALBERT, S. LAM, J. KONSTAN et J. RIEDL : MovieLens Unplugged : Experiences with an Occasionally Connected Recommender System. *In Proceedings of the 8th international conference on Intelligent user interfaces*, pages 263–266, 2003.

K. MIYAHARA et M. PAZZANI : Collaborative Filtering with the Simple Bayesian Classifier. *PRICAI 2000 Topics in Artificial Intelligence*, pages 679–689, 2000.

K. MIYAHARA et M. PAZZANI : Improvement of Collaborative Filtering with the Simple Bayesian Classifier. *Transactions of Information Processing Society of Japan*, 43(11):3429–3437, 2002.

B. MOBASHER : *Data Mining for Web Personalization*, chapitre 3, pages 90–135. LNCS 4321 - Brusilovsky, P. and Kobsa, A. and Nejdl, W., 2007.

R. MOONEY et L. ROY : Content-based Book Recommending using Learning for Text Categorization. *In Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, 2000.

M. NAKAGAWA et B. MOBASHER : A Hybrid Web Personalization Model Based on Site Connectivity. *In Proceedings of WebKDD*, pages 59–70, 2003a.

M. NAKAGAWA et B. MOBASHER : Impact of Site Characteristics on Recommendation Models Based On Association Rules and Sequential Patterns. *In Intelligent Techniques for Web Personalization*, 2003b.

S. NEEDLEMAN et C. WUNSCH : A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March 1970.

L. NERIMA, V. SERETAN et E. WEHRLI : Le problème des collocations en TAL. *Nouveaux cahiers de linguistique française*, 27:95–115, 2006.

H. NEY, U. ESSEN et R. KNESER : On Structuring Probabilistic Dependences in Stochastic Language Modelling. *Computer Speech and Language*, 8(1):1–38, 1994.

A. NGUYEN, N. DENOS et C. BERRUT : Exploitation des données disponibles à froid pour améliorer le démarrage à froid dans les systèmes de filtrage d'information. *In INFormatique des ORganisations et Systèmes d'Information et de Décision*, pages 81–95, 2006.

F. OCH et H. NEY : Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *In ACL '02 : Proceedings of the 40th Annual Meeting on Association for*

Computational Linguistics, pages 295–302, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

C. PALMISANO, A. TUZHILIN et M. GORGOGLIONE : Using Context to Improve Predictive Modeling of Customers in Personalization Applications. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1535–1549, 2008.

N. PASQUIER, Y. BASTIDE, R. TAOUIL et L. LAKHAL : Discovering Frequent Closed Itemsets for Association Rules. *Database Theory—ICDT'99*, pages 398–416, 1999.

D. PAVLOV, E. MANAVOGLU, D. PENNOCK et C. GILES : Collaborative Filtering with Maximum Entropy. *IEEE Intelligent Systems*, 19(6):40–48, 2004. ISSN 1541-1672.

M. PAZZANI et D. BILLSUS : Learning and Revising User Profiles : The Identification of Interesting Web Sites. *In Machine Learning*, pages 313–331, 1997.

M PAZZANI et D. BILLSUS : Content-Based Recommendation Systems. *The Adaptive Web*, pages 325–341, 2007.

J. PEI, J. HAN et R. MAO : CLOSET : An Efficient Algorithm for Mining Frequent Closed Itemsets. *In ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 21–30, 2000.

J. PEI, J. HAN, B. MORTAZAVI-ASL, H. PINTO, Q. CHEN, U. DAYAL et M. HSU : PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. *In Proceedings of the 17th International Conference on Data Engineering*, pages 215–224, Washington, DC, USA, 2001. IEEE Computer Society.

J. PITKOW et P. PIROLLI : Mining Longest Repeating Subsequences to Predict World Wide Web Surfing. *In USITS'99 : Proceedings of the 2nd conference on USENIX Symposium on Internet Technologies and Systems*, pages 139–150, 1999.

A. POLGUERE : *Lexicologie et sémantique lexicale : notions fondamentales*. Pum, 2003.

M. PORTER : An Algorithm for Suffix Stripping. pages 313–316, 1997.

P. RESNICK, N. IACOVOU, M. SUCHAK, P. BERGSTROM et J. RIEDL : GroupLens : An Open Architecture for Collaborative Filtering of Netnews. *In CSCW '94 : Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186, New York, NY, USA, 1994. ACM.

P. RESNICK et H. VARIAN : Recommender Systems. *Communications of the ACM*, 40(3):56–58, 1997. ISSN 0001-0782.

P. RESNIK : *Selection and Information : a Class-Based Approach to Lexical Relationships*. Philadelphia, 1993.

R. ROCHLITZ : Dans le flou artistique. Élements d'une théorie de la « rationalité esthétique ». *Bouchindhomme et Rochlitz*, pages 203–238, 1992.

F ROSENBLATT : The Perception : A probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, pages 386–408, 1958.

R. ROSENFELD : *Adaptive Statistical Language Modeling : A Maximum Entropy Approach*. Thèse de doctorat, Carnegie Mellon University, Computer Science Department, 1994.

R. ROSENFELD : Two Decades of Statistical Language Modeling : Where do we go from here?. *Proceedings of the IEEE*, pages 1270–1278, 2000.

R. ROSENFELD et X. HUANG : Improvement in Stochastic Language Modeling. *In Speech and Natural Language*, pages 107–111, San Mateo,CA, 1992.

G. SALTON et C. BUCKLEY : Term Weighting Approaches in Automatic Text Retrieval. Rapport technique, Ithaca, NY, USA, 1987.

J. SANDVIG, B. MOBASHER et R. BURKE : Robustness of Collaborative Recommendation Based on Association Rule Mining. *In RecSys '07 : Proceedings of the 2007 ACM conference on Recommender systems*, pages 105–112, New York, NY, USA, 2007. ACM.

R. SARUKKAI : Link Prediction and Path Analysis Using Markov Chains. *In Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 377–386, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

B. SARWAR, G. KARYPIS, J. KONSTAN et J. REIDL : Item-based Collaborative Filtering Recommendation Algorithms. *In WWW '01 : Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

B. SARWAR, G. KARYPIS, J. KONSTAN et J. RIEDL : Analysis of Recommendation Algorithms for e-commerce. *In EC '00 : Proceedings of the 2nd ACM conference on Electronic commerce*, pages 158–167, New York, NY, USA, 2000. ACM.

J. SCHAFER, D. FRANKOWSKI, J. HERLOCKER et S. SEN : *Collaborative Filtering Recommender Systems*, chapitre 9, pages 291–324. Peter Brusilovsky, Alfred Kobsa and W Nejdl. Springer, 2007.

S. SCHMITT et R. BERGMANN : Applying Case-Based Reasoning Technology for Product Selection and Customization in Electronic Commerce Environments. *In 12th Bled Electronic Commerce Conference*, pages 1–15, 1999.

M. SENO et G. KARYPIS : Lpminer : An Algorithm for finding Frequent Itemsets using Length-Decreasing Support Constraint. *In Proceedings of the 2001 IEEE International Conference on Data Mining*, volume 29, pages 505–512, 2001.

C. SHAHABI, F. BANAEI-KASHANI, Y. CHEN et D MCLEOD : Yoda : An Accurate and Scalable Web-Based Recommendation System. *In Sixth International Conference on Cooperative Information Systems*, pages 418–432, 2001.

G. SHANI, D. HECKERMAN et R. BRAFMAN : An MDP-Based Recommender System. *JMLR : The Journal of Machine Learning Research*, pages 453–460, 2005.

T. SMITH et M. WATERMAN : Identification Of Common Molecular Subsequences, 1981.

B. SMYTH : Case-Based Recommendation. *The Adaptive Web*, pages 342–376, 2007.

R. SRIKANT et R. AGRAWAL : Mining Sequential Patterns : Generalizations and Performance Improvements. *In EDBT '96 : Proceedings of the 5th International Conference on Extending Database Technology*, pages 3–17, London, UK, 1996. Springer-Verlag.

M. STOLZE et Walid R. : Towards Scalable Scoring for Preference-based Item Recommendation. *IEEE Data Engineering Bulletin*, 24:42–49, 2001.

J. SU, H. YEH, P. YU et V. TSENG : Music Recommendation Using Content and Context Information Mining. *IEEE Intelligent Systems*, 25(1):16–26, 2010.

X. SU et T. KHOSHGOFTAAR : Collaborative Filtering for Multi-class Data Using Belief Nets Algorithms. *In ICTAI '06 : Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 497–504, Washington, DC, USA, 2006. IEEE Computer Society.

L. TAUSCHER et S. GREENBERG : How People Revisit Web Pages : Empirical Findings and Implications for the Design of History Systems. *International Journal of Human Computer Studies*, 47:97–137, 1997.

C. TILLMANN et H. NEY : Selection Criteria for Word Trigger Pairs in Language Modeling. *Grammatical Interference : Learning Syntax from Sentences*, pages 95–106, 1996.

B. TOWLE et C. QUINN : Knowledge Based Recommender Systems Using Explicit User Models. pages 74–77, 2000.

C. TRONCOSO, T. KAWAHARA, H. YAMAMOTO et G. KIKUI : Trigger-based Language Model Construction by Combining Different Corpora. Rapport technique 542, IEICE, 2004.

B. TROUSSE : Evaluation of the Prediction Capability of a User Behaviour Mining Approach For Adaptive Web Sites. *In Proceedings of the 6th RIAO Conference - Content-Based Multimedia Information Access*, 2000.

B. TROUSSE, M. JACZYNSKI et R. KANAWATI : Une approche fondée sur le raisonnement à partir de cas pour l'aide à la navigation dans un hypermédia. *In Hypertexte & Hypermedia : Products, Tools and Methods*, 1999.

M. VIERMETZ, C. STOLZ, V. GEDOV et M. SKUBACZ : Relevance and Impact of Tabbed Browsing Behavior on Web Usage Mining. *In Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 262–269, Washington, DC, USA, 2006. IEEE Computer Society.

W. WANG et O. ZAÏANE : Clustering Web Sessions by Sequence Alignment. *In Proceedings of the 13th International Workshop on Database and Expert Systems Applications*, pages 389–394. IEEE COmputer Society, 2002.

Y. WANG, Z. LI et Y. ZHANG : Mining Sequential Association-Rule for Improving WEB Document Prediction. *In ICCIMA '05 : Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications*, pages 146–151, Washington, DC, USA, 2005. IEEE Computer Society.

H. WEINREICH, H. OBENDORF, E. HERDER et M. MAYER : Off the Beaten Tracks : Exploring Three Aspects of Web Navigation. *In WWW '06 : Proceedings of the 15th international conference on World Wide Web*, pages 133–142, New York, NY, USA, 2006. ACM.

X. YAN, J. HAN et R. AFSHAR : CloSpan : Mining Closed Sequential Patterns in Large Datasets. *In Proceedings of SIAM International Conference on Data Mining*, 2003.

B. YEONG, H. YOON et H. SAOUNG : Mining changes in customer buying behavior for collaborative recommendations. *Expert Systems with Applications*, 28(2):359–369, 2005.

K. YOSHII, M. GOTO, K. KOMATANI, T. OGATA et H. OKUNO : Hybrid Collaborative and Content-Based Music Recommendation Using Probabilistic Model with Latent User Preferences. *In Proceedings of the International Conference on Music Information Retrieval*, 2006.

M. ZAKI : Fast Mining of Sequential Patterns in Very Large Databases. Rapport technique, University of Rochester, 1997.

T. ZHANG, R. RAMAKRISHNAN et M. LIVNY : BIRCH : An Efficient Data Clustering Method for Very Large Databases. *In ACM SIGMOD International Conference on Management of Data*, pages 103–114, 1996.

Y. ZHANG, J. CALLAN et T. MINKA : Novelty and Redundancy Detection in Adaptive Filtering. *In SIGIR '02 : Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 81–88, New York, NY, USA, 2002. ACM.

G. ZHOU et K. LUA : Word Association and MI-Trigger-based Language Modeling. *In Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 1465–1471. Association for Computational Linguistics, 1998.

Andrew ZIMDARS, David Maxwell CHICKERING et Christopher MEEK : Using Temporal Data for Making Recommendations. *In* Jack S. BREESE et Daphne KOLLER, éditeurs : *UAI*, pages 580–588. Morgan Kaufmann, 2001.

I. ZUKERMAN, D. ALBRECHT et A. NICHOLSON : Predicting Users' Requests on the WWW. *In UM '99 : Proceedings of the seventh international conference on User modeling*, pages 275–284, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

Résumé

Le but de cette thèse est d'améliorer la qualité des systèmes de recommandation pour la navigation Web en utilisant la séquentialité des actions de navigation des utilisateurs. La notion de séquentialité a déjà été étudiée dans ce contexte. De telles études tentent habituellement de trouver un bon compromis entre précision, complexité en temps et en mémoire, et couverture. De plus, le Web a cela de particulier que du bruit peut être contenu au sein des navigations (erreurs de navigation, apparition de pop-ups, etc.), et que les utilisateurs peuvent effectuer des navigations parallèles. La plupart des modèles qui ont été proposés dans la littérature exploitent soit des suites contiguës de ressources et ne sont pas résistants au bruit, soit des suites discontiguës de ressources et induisent une complexité en temps et en mémoire importantes. Cette complexité peut être réduite en effectuant une sélection sur les séquences, mais cela engendre alors des problèmes de couverture. Enfin à notre connaissance, le fait que les utilisateurs puissent effectuer des navigations parallèles n'a jamais été étudié du point de vue de la recommandation.

La problématique de cette thèse est donc de proposer un nouveau modèle séquentiel ayant les cinq caractéristiques suivantes : (1) une bonne précision de recommandation, (2) une bonne résistance au bruit, (3) la prise en compte des navigations parallèles, (4) une bonne couverture (5) et une faible complexité en temps et en mémoire.

Afin de répondre à cette problématique, nous nous inspirons de la Modélisation Statistique du Langage (MSL), qui a des caractéristiques très proches de celles de la navigation Web. La MSL est étudiée depuis beaucoup plus longtemps que les systèmes de recommandation et a largement prouvé sa précision et son efficacité. De plus, la plupart des modèles statistiques de langage qui ont été proposés prennent en compte des séquences. Nous avons donc étudié la possibilité d'exploiter les modèles utilisés en MSL et leur adaptation aux contraintes spécifiques de la navigation Web.

Abstract

The goal of this thesis is to enhance the quality of the recommender systems for Web navigation by using sequentiality. The notion of sequentiality has been widely studied in the frame of Web recommendation. Such studies usually attempt to provide a trade-off between accuracy, space and time complexity, and coverage. Two extra problems are the presence of noise within the sessions of navigations (navigation mistakes, pop-ups, etc.), and parallel browsing. Most of the models that have been proposed in the literature either exploit low size contiguous sequences and are not robust to noise, or discontiguous sequences and induce large time and space complexities. This last problem can be lowered by performing a selection of sequences in order to lower space complexity, but this results in a coverage problem. Last, to the best of our knowledge, parallel browsing has never been studied in the frame of recommendation.

The challenge of this thesis is thus to propose new sequential algorithms that have the five following characteristics : (1) a good precision of recommendations, (2) a good robustness to noise, (3) the ability to take into account parallel browsing, (4) a high coverage and (5) a low time and space complexity.

In order to complete this challenge, we take inspiration from Statistical Language Modeling (SLM). Indeed, the general characteristics of Web navigation are very close to those of natural language. SLM dates back to a longer time than recommender systems and have widely proved their accuracy and efficiency. Moreover, most of the statistical language models that have been proposed take into account sequences. We thus investigated the exploitation of models used in language modeling and adapted them to the specific constraints of Web navigation.