CIGI 2017
État de l'art sur les systèmes de recommandation

CAMELIA DADOUCHI[1,2], BRUNO AGARD[1,2]

[1] Département de mathématiques et génie industriel, École Polytechnique de Montréal
CP 6079, succursale Centre-Ville, Montréal, Québec, Canada

[2] Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport (CIRRELT)

camelia.dadouchi@polymtl.ca, bruno.agard@polymtl.ca

***Résumé* – Les Systèmes de recommandations (SRs) sont une conséquence de l'utilisation d'internet comme canal de distribution mondial. L'abondance de la variété des produits a causé une surcharge d'information et les SRs visent à proposer les produits adaptés à chaque client. Les données utilisées dans les SRs sont diverses et de provenances multiples. Nous tenterons de donner un aperçu des catégories de SRs, de leurs fonctionnements, des métriques d'évaluations et des chalenges auxquels ces outils font face.**

***Abstract* - Recommendation Systems (RS) are a consequence of using the Internet as a global distribution channel. The abundance of the variety of products has caused an overload of information and SRs aim to propose content adapted to each customer. The data used in the SRs are diverse and of multiple origins. We try to give an overview of the categories of SRs, how they operate, the challenges they face and their evaluations metrics.**

Mots clés – Systèmes de recommandation, fouilles de données, recommandations basées sur le contenu, filtrage collaboratif.

Keywords – recommender systems, datamining, content-based recommender systems, collaborative filtering.

1 INTRODUCTION

Au cours des dernières décennies, nous avons assisté à une révolution de l'information. Cette révolution a changé le monde tel que nous le connaissions. Nous avons assisté à la démocratisation de l'ordinateur et à la création de l'intelligence artificielle, en passant par la démocratisation de l'internet. Mais l'un des effets de cette révolution qui n'étaient pas prévus est l'émergence explosive d'internet en tant qu'important canal de distribution mondial des marchandises (Drucker, 1999).

L'adoption massive du web comme plateforme de commerce a mené à des changements fondamentaux dans l'interaction avec les clients. Cela a permis de toucher un public plus large et plus diversifié, mais cela a également ouvert les portes à une compétition plus féroce avec la disparition virtuelle des frontières. Cette évolution spectaculaire a créé des enjeux de taille, ce qui a poussé les entreprises à s'adapter pour survivre, en développant des outils stratégiques sophistiqués (Bossenbroek et Gringhuis, 2015). Ces outils ne sont plus utilisés uniquement pour des campagnes ponctuelles, mais également pour créer des relations à long terme avec des clients. Parmi les formes prises par les outils de renforcement des relations clients, on trouve les systèmes de recommandations (SR) (Demiriz, 2004). Les systèmes de recommandation sont présents un peu partout, et nous y sommes confrontés constamment, que ce soit pour les recommandations de contenu informatif, ou pour la recommandation de produits sur les sites d'e-commerce. (Chen et al., 2016).

Dans cet article, nous proposons un état de l'art sur cette technologie : (2.1) détaille les définitions couramment admises, (2.2) précise les objectifs visés par ces outils, (2.3) liste leurs principales fonctionnalités, (2.4) catégorise les systèmes de recommandation, (2.5) se concentre sur le fonctionnement des principales méthodes utilisées dans ces systèmes et sur les métriques d'évaluation. La section (2,6) fait ressortir les principaux chalenges techniques à relever dans les prochaines années, puis nous concluons en section (4)

2 SYSTEMES DE RECOMMANDATION

2.1 Définitions des systèmes de recommandation

On trouve différentes définitions des systèmes de recommandation dans la littérature. Certaines, expliquent que les systèmes de recommandation sont des outils qui assistent les utilisateurs avec les problèmes de surcharge croissante d'informations et améliore la gestion de la relation client en fournissant aux utilisateurs des recommandations personnalisées de produits ou de services (Jie et al., 2015).

D'autres, considèrent les SRs comme étant une sous-classe des systèmes de filtrage d'informations qui tente de prédire les « évaluations » ou les « préférences » que les utilisateurs attribueraient à un produit en collectionnant des informations sur les préférences des utilisateurs du SR pour un ensemble d'items (Ricci et al, .2011).

On trouve également des définitions, comme celle fournie par (Adomavicius et al., 2005), qui explique que les systèmes de recommandation tentent de prédire les évaluations pour les produits inconnus, pour chaque utilisateur, souvent en utilisant les évaluations des autres utilisateurs, et recommande les N meilleurs items ayant la plus haute valeur d'évaluation prédite. Adomavicius a aussi donné une définition plus formelle, qu'il a exprimé comme suit :

"Let C be the set of all users and let S be the set of all possible items that can be recommended, such as books, movies, or restaurants. The space S of possible items can be very large, ranging in hundreds of thousands or even millions of items in some applications, such as recommending books or CDs. Similarly, the user space can also be very large—millions in some cases. Let u be a utility function that measures the usefulness of item s to user c, i.e., u:C×S→R, where R is a totally ordered set (e.g., nonnegative integers or real numbers

within a certain range). Then, for each user c∈C, we want to choose such item s'∈S that maximizes the user's utility. More formally:

∀c∈C,s'c=argmaxs∈Su(c,s)" (Adomavicius et al., 2005)

Dans une édition de 2011 de l'Association de l'avancement de l'intelligence artificielle, les auteurs considèrent que les différentes définitions s'accordent sur deux traits distinctifs des systèmes de recommandation. On peut lire dans l'article « Recommender Systems : An Overview" les deux affirmations suivantes :

- "*A recommender system is personalized. The recommendations it produces are meant to optimize the experience of one user, not to represent group consensus for all.*
- *'A recommender system is intended to help the user select among discrete options. Generally, the items are already known in advance and not generated in a bespoke fashion." (Burke et al., 2011)*

Les définitions pour les systèmes de recommandation diffèrent d'une communauté à une autre, mais restent tout de même toutes axées sur l'établissement d'une liste d'items à recommander aux utilisateurs, en se basant sur des algorithmes issus de différentes disciplines qui permettent de prédire l'appréciation d'un utilisateur pour une liste de produits inconnus et de lui suggérer, parmi un large éventail d'items, ceux qui sauront répondre à des besoins spécifiques.

Les systèmes de recommandations sont utilisés dans plusieurs disciplines, tel : e-government, e-business, e-commerce, e-learning, e-tourisme, e-resource services and e-group activities. Une revue de littérature dans ce sens a été fournie par (Jie et al., 2015), ils illustrent les différents rôles que jouent les SRs dans les différentes disciplines tout en expliquant les techniques de recommandations utilisées par discipline. Parmi les nombreux champs d'application de ce domaine, la suite de l'article se concentre uniquement sur les systèmes de recommandation appliqués au e-commerce et au e-business.

Pour la suite du document nous définirons :

-Item comme étant tout objet/service/bien/relation pouvant être recommandé. Exemples d'items : films, musiques, livres, articles scientifiques, restaurants, produits, services financiers, amis, emplois…

-Utilisateur l'individu cible du système de recommandation

2.2 Objectifs des systèmes de recommandations

En plus des deux points sur lesquels s'accordent les définitions des systèmes de recommandation. On peut relever plusieurs objectifs des systèmes de recommandation. Pour n'en citer que trois : les systèmes de recommandation aident (1) à décider quels produits offrir à un client déterminé dans un domaine où ils disposent de peu d'informations pour trier et évaluer les alternatives possibles, (2) à augmenter les ventes croisées en proposant des produits supplémentaires aux clients, et (3) à améliorer la fidélité des consommateurs, car ceux-ci ont tendance à revenir vers les sites qui répondent le mieux à leurs besoins (Lu et al., 2012 ; Shardanand et al., 1995 ; Resnick et al., 1997 ; Konstant, 1997).

2.3 Fonctionnalités des systèmes de recommandations

Dans sa thèse de doctorat, (Meyer, 2012) fait un découpage des systèmes de recommandation selon les fonctions qu'ils accomplissent. Il en relève 4, énoncées ci-dessous :

"**Help to Decide**: predicting a rating for a user for an item

Help to Compare: rank a list of items in a personalized way for a user

Help to Discover: provide a user with unknown items that will be appreciated

Help to Explore: give items similar to a given target item"

2.3.1 Aide à la décision

Il existe une variété de produits disponibles à l'achat par l'utilisateur d'un système de recommandations. Ces produits sont de différentes catégories et il est plus au moins difficile de quantifier leurs qualités et leurs fiabilités.

Les systèmes de recommandations sont des outils qui aident l'utilisateur à prendre une décision quand il n'est pas sûr de son choix ou qu'il n'a pas assez de connaissances ou de moyens pour évaluer lui-même ces items. En faisant la prédiction de l'appréciation d'un utilisateur pour une liste de produits, on aide l'utilisateur à choisir l'item qui est le plus prometteur pour lui. Cette fonctionnalité est intéressante, car l'utilisateur lui-même pourrait ignorer des attributs essentiels quand il est novice dans un domaine, chose que le système de recommandation aide à pallier en accumulant des informations basées sur le contenu créé par les utilisateurs.

2.3.2 Aide à la comparaison

En plus de faire des prédictions d'appréciation d'items, le système de recommandation peut aussi faire le classement de ceux-ci par pertinence pour l'utilisateur. L'item au premier rang devrait être plus pertinent pour l'utilisateur que celui au dernier rang. Ce qui permet à l'utilisateur de comparer différents produits qui pourraient tous être adaptés à ses intérêts. Cette fonctionnalité est particulièrement intéressante lorsque les données dont on dispose sont des données implicites qui ne fournissent pas le degré d'appréciation pour un item (nous savons si un utilisateur a apprécié un item, mais pas à quel point celui-ci l'a apprécié).

2.3.3 Aide à la découverte et à l'exploration

Cette fonction des systèmes de recommandation permet aux utilisateurs de découvrir des produits dont ils ne connaissaient peut-être pas l'existence. Ces produits peuvent être nouveaux ou juste spécialisés. L'aide à la découverte est un des défis auxquels font face les systèmes de recommandation. Cette fonction est très pertinente quand il s'agit de recommandations de produits niches, car elle permet d'explorer différents types d'items. Dans ce cas, il n'est pas pertinent de recommander les produits populaires, car l'utilité est d'avoir une large couverture.

Le magazine « Wired » a proposé, d'utiliser la longue traîne « Long Tail » pour décrire le modèle de e-commerce d'entreprises comme Amazon (Anderson, 2004).

Proposer un large éventail de produit, malgré la faible fréquence d'achats pour certains items, peut générer beaucoup de profits. Le profit généré par les meilleurs vendeurs peut être comparable à celui des produits niches quand le catalogue de produits est large et varié.

La liste d'items proposés à l'utilisateur pour l'aide à la découverte doit remplir certains critères comme : être pertinente, être utile, être crédible.

Cependant, pour faire de la recommandation de découverte pertinente. Il faut avoir cumulé assez d'informations sur les intérêts des utilisateurs (historique d'achats/navigations/click Stream / ratings…). Dans le cas contraire, il serait risqué de proposer des produits niches, surtout si le système est en « boîte noire » et que l'utilisateur n'a pas de visibilité sur le mécanisme de fonctionnement de la recommandation, cela pourrait nuire à la crédibilité du système (He et al., 2016). On peut donc distinguer deux types de recommandations : la recommandation pour l'exploration et la recommandation pour la découverte de nouveaux produits. L'exploration serait utilisée quand l'utilisateur est nouveau sur la plateforme ou qu'il n'y a pas assez d'information sur ses préférences, qu'elles soient implicites ou explicites. Dans ce cas, il faudra se contenter d'une recommandation item à item. Dans le cas le plus répandu, la recommandation se fait en deux étapes : on commence par fournir des produits de la catégorie des meilleurs vendeurs, ensuite en fonction des produits dont l'utilisateur a visité la page, une recommandation pourra être faite en proposant des items similaires à celui préféré par l'utilisateur.

2.4 Catégories de systèmes de recommandations

Afin de répondre aux fonctionnalités attendues des systèmes de recommandations, plusieurs méthodes existent. La typologie traditionnelle a été établie par (Adomavicius et al., 2005). Ils considèrent trois types de systèmes de recommandation, basés sur les fonctions internes du système :

1. Recommandations basées sur le contenu (CB) : L'utilisateur se fera recommander des items similaires à ceux qu'il a préférés dans le passé.
2. Recommandations collaboratives (CF) : L'utilisateur se fera recommander des items que des personnes avec des goûts et des préférences semblables ont appréciés dans le passé
3. Approches hybrides: Ces méthodes combinent les deux méthodes précédentes.

Une autre classification bien répandue est celle proposée par (Xiaoyuan et al., 2009) qui fait une classification se basant sur les méthodes de recommandation.

1. Memory-based CF techniques : Les algorithmes de filtrage collaboratif basés sur la mémoire utilisent la totalité ou un échantillon de la base de données item/utilisateur pour générer des prédictions. Chaque utilisateur fait partie d'un groupe de personnes qui a des intérêts similaires. En identifiant les « voisins » de l'utilisateur actif, une prédiction des préférences par rapport à de nouveaux items peut être produite.
2. Model-based CF techniques : Le design et le développement de modèles peuvent permettre au système d'apprendre à reconnaître des modèles complexes dans les données d'apprentissage. Une fois que le modèle est appris, le SR pourra faire des prédictions intelligentes sur de nouvelles données. Les algorithmes collaboratifs basés sur les modèles aident à contourner plusieurs des lacunes auxquels ont fait face les systèmes collaboratifs basés sur la mémoire.
3. Hybrid CF techniques : Les CFs hybrides combinent les recommandations CF avec d'autres techniques de recommandations comme les recommandations basées sur le contenu.

Ces typologies ont été la base de plusieurs autres classifications de systèmes de recommandation. Une classification enrichie et plus récente, incluant les deux classifications précédentes, est présentée par (Candillier et al., 2007).

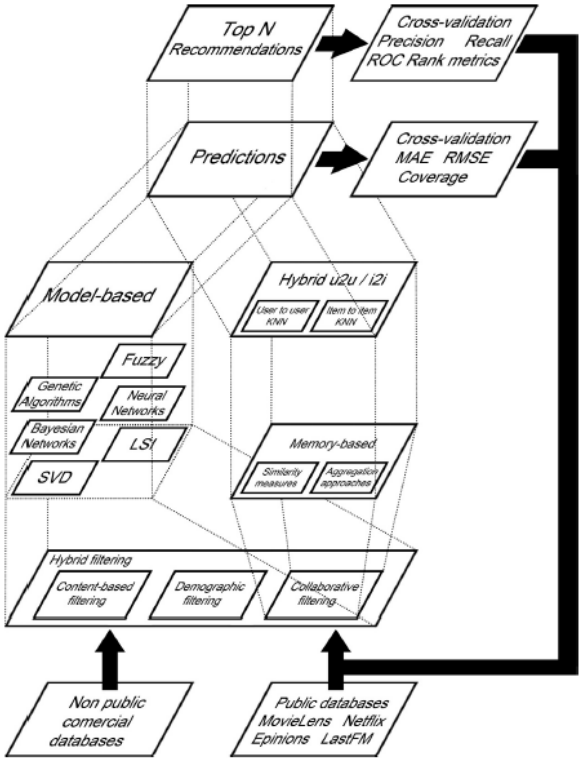

Figure 1 : Modèles traditionnels de SR et la relation entre eux. Reproduit de Knowledge-Based Systems, 46 (2013), J. Bobadilla et al, Recommender systems survey, 109–132, Copyright (2016) avec permission de Elsevier.

La figure 2, de (Bobadilla et al., 2013), montre les relations entre les différents algorithmes de filtrages traditionnels (basé sur le contenu, le filtrage démographique et le filtrage collaboratif) et fait le lien avec les méthodes de recommandations utilisées, soit la méthode basée sur la mémoire et la méthode basée sur le modèle. En plus de cela, la figure met en reliefs les aspects les plus importants d'un système de recommandation soit : les données, les algorithmes et les métriques utilisées.

Nous traiterons dans les sections suivantes tous les aspects de la figure 2 vu la pertinence de ceux-ci. Nous commencerons par présenter les algorithmes les plus utilisés par méthode de recommandation, ensuite nous donnerons un aperçu du type de données que peuvent traiter les systèmes de recommandation et de leurs influences sur le choix de la méthode de recommandation. Ensuite, quelques métriques d'évaluation traditionnelles des SRs seront présentées dans leurs contextes.

2.5 Fonctionnement des méthodes de recommandations

Dans cette section, nous commençons par faire état du type de données utilisées dans la recommandation et leurs spécificités. Ensuite, nous présenterons les différents algorithmes utilisés dans la recommandation. Finalement, nous présenterons les différents types de recommandation.

2.5.1 Types de données

Commençons par le type de données que l'on trouve dans les SRs. Dans un premier temps, il est nécessaire de différencier entre les données **explicites** et les données **implicites.**

Les données explicites sont des données qui n'ont pas été interprétées, l'utilisateur nous les fournit directement. Généralement ce sont des données comme les données sur l'appréciation (ratings) de l'utilisateur pour un produit. Des systèmes de recommandation interactifs plus récents recueillent des informations explicites à travers une interface ou l'utilisateur est sollicité à fournir des renseignements sur ses préférences.

Les données implicites sont des données qui ne requièrent pas d'intervention spécifique de l'utilisateur. Généralement ces données sont recueillies à travers la surveillance des actions des utilisateurs sur la plateforme qu'ils utilisent. Typiquement des données sur le click Stream (données représentants le parcours de l'utilisateur sur un site web), les téléchargements, sites web visités, l'historique des achats, les coordonnées GPS…

Les données implicites englobent également les données sociodémographiques, les données recueillies sur les réseaux sociaux et plus récemment les données recueillies à partir des objets connectés (internet of things). Les données sur **les usagers** et les données sur **les items**.

Il est important de différencier les données sur les items de celles sur les usagers pour le calcul des recommandations. Selon les méthodes, les calculs peuvent être effectués sur différentes combinaisons de données items/utilisateurs soit : items/items, utilisateurs/items, utilisateurs/utilisateurs ou items/utilisateurs (Meyer, 2012)

Suite à la différenciation entre données implicites/explicites et données utilisateurs/items, nous pouvons voir comment les caractéristiques des données s'imbriquent dans le découpage présenté à la figure 3 (Bobadilla et al., 2013).

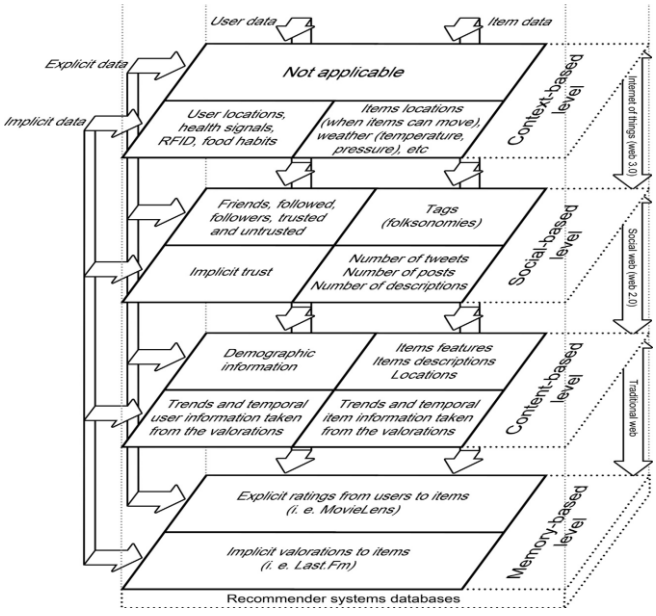

Figure 2: Taxonomie des données usuellement utilisées dans les SRs, adapté de (Bobadilla et al., 2013)

On peut voir à la figure 2, une classification des données selon leur nature. Cette classification se base sur trois axes, la **cible des données** (utilisateur ou item), le **mode d'acquisition (**implicite ou explicite), et **le niveau d'information** (basée sur la mémoire, basée sur le contenu, basé sur le social ou basé sur le contexte). Les données disponibles ont évolué en parallèle avec le Web. La figure découpe le web en trois niveaux : le web traditionnel, le web social et l'internet des objets. Le type de données qu'il est possible de recueillir est relatif au niveau d'évolution du web. À titre d'exemple, la première génération du web (traditionnel) contrairement à la troisième (internet des objets) ne permet pas de faire le suivi en temps réel de la situation géographique d'un produit.

2.5.2 Techniques de data mining utilisées dans les SR

Les systèmes de recommandation ont vu le jour en combinant plusieurs techniques et méthodes qui appartiennent à différentes disciplines telles la recherche d'information, le filtrage d'information ou l'interaction homme-machine. Cependant, il est intéressant de noter que la majeure partie de ses disciplines utilisent des algorithmes de data mining. Dans cette section nous tenterons de faire un portrait des algorithmes les plus utilisés dans les systèmes de recommandation dans une optique de data mining.

(Amatriain et al., 2011) représente les outils de data mining utilisés dans les systèmes de recommandation dans le cadre type du processus de data mining qui se divise en 3 étapes, soit : prétraitement des données, analyse des données, et interprétation des résultats.

Dans cette section, nous expliquerons les méthodes les plus importantes et les plus utilisées, sans pour autant rentrer dans les détails de chacune d'entre elles.

(A) Prétraitement des données :

Les données réelles sont généralement des données brutes qu'il faut prétraiter avant de pouvoir les utiliser, en particulier les données des SRs sont souvent non structurées et clairsemées (sparse). Parmi les méthodes utilisées, on trouve :

(i) Le calcul de similarité

Généralement, pour les systèmes de recommandation les calculs de similarité se font entre les items disponibles, non notés préalablement par l'utilisateur, et le profil de l'utilisateur ou entre un utilisateur cible et tous les autres utilisateurs.

La méthode la plus simple est le calcul des **distances**. Parmi les distances les plus utilisées, on trouve la distance euclidienne, généralisée par (Minkowski, 1897) comme suit :

$$d(x,y) = \sum_{k=1}^{n}(|x_k - y_k|^r)^{\frac{1}{r}} \qquad (1)$$

Avec n le nombre de dimension (attributs) et x_k et y_k sont les k^{th}composants des objets x et y respectivement. Avec r étant le degré de distance. Ex : Distance euclidienne ↔ r = 2

Une autre approche très commune est de considérer les items/utilisateurs comme des vecteurs à n dimensions et calculer **la similarité** entre les deux vecteurs. La « cosine similarity » qui évalue le cosinus de l'angle que forment les deux vecteurs est la mesure la plus utilisée. Elle est définie comme suit :

$$\text{Cos (x,y)} = \frac{(x \bullet y)}{||x||\ ||y||} \qquad (2)$$

La similarité peut aussi être donnée par la **corrélation**. La corrélation entre deux objets est la mesure de la relation de linéarité entre ses deux objets. La mesure la plus utilisée est la Pearson corrélation donnée par l'équation 3.

$$\text{Pearson (x, y)} = \frac{\text{Cov (X,Y)}}{\sigma X\ .\sigma Y} \qquad (3)$$

Il existe différentes variantes des deux mesures énoncées précédemment, ainsi que d'autres mesures de similarité pour les données binaires comme « The Simple Matching

coefficient SMC, the Jaccard coefficient. The Extended Jaccard (Tanimoto) coefficient. »
Les mesures de similarités ont évolué avec le temps pour tenir compte de la variance dans les évaluations entre les usagers. Par exemple, la cosine similarité ajustée soustrait de l'évaluation de l'utilisateur la moyenne de ses évaluations passées avant de calculer les similarités sur lesquelles se basent la recommandation.
Dans le cas général, il a été démontré dans la littérature que la précision des prédictions pour les recommandations est indépendante du choix de la mesure de similarité à utiliser (Lathia et al., 2008).

(ii) La réduction des dimensions de la matrice SR :

Les données qui alimentent les SRs sont souvent des données lourdes, à cause du nombre élevé de clients et d'items offerts par les entreprises de e-commerce. En plus de cela, elles sont très clairsemées, à cause de la quantité limitée d'appréciations (ratings) disponibles pour chaque utilisateur et/ou item. Dans de telles conditions, les algorithmes de data mining voient leurs performances diminuées à cause de la faible densité d'information dans les données. Pour pallier au problème, on utilise des méthodes de réduction de dimensions, qui améliorent grandement la qualité des prédictions et font baisser le coût de calcul. Cette étape, est considérée comme une approche de recommandation et non comme une étape de prétraitement, comme elle l'aurait été en data mining. Parmi les méthodes les plus utilisées dans le contexte des RSs on trouve principalement : l'analyse des composantes principales (PCA- Principal component analysis) et la décomposition de la valeur singulière (SVD - Singular Value decomposition).
PCA est une méthode de statistique classique utilisée pour trouver des modèles dans des jeux de données à dimensionnalité élevée (Jolliffe, 2002). La méthode consiste à transformer des variables liées en des variables décorrélées les unes des autres. Ces variables sont les « composantes principales ». Cette méthode permet de trouver une liste ordonnée, les composantes responsables de la plus grande partie de la variation dans les données. On réduit la dimensionnalité des données en ignorant les composantes ayant très peu d'impact sur la variance.
SVD est une méthode de réduction de la dimensionnalité, c'est un cas spécial de la factorisation de matrice (Golub et Reinsch, 1970). L'objectif principal est de trouver une dimensionnalité plus faible dans laquelle les nouvelles caractéristiques sont des « concepts » dont on peut calculer l'importance relative. Un avantage notable de la SVD pour les SRs est que c'est une méthode qui peut être incrémentale et permettre d'accepter de nouveaux utilisateurs/items/appréciations sans avoir à refaire en totalité les calculs pour les nouvelles recommandations.

(B) Traitement des données

Cette partie fera état des méthodes de traitement de données les plus répandues dans ce domaine.

(i) La classification

La classification est une méthode qui permet de prédire la catégorie d'un objet. Un modèle est construit à partir d'un jeu d'apprentissage, il est ensuite utilisé pour classer les données nouvelles.
Il existe plusieurs types de classificateurs, qui sont généralement divisés en deux catégories : classification avec apprentissage supervisée et classification avec apprentissage non supervisée. Pour l'apprentissage supervisé, les données de base ont des variables dépendantes, préalablement classifiées en catégories connues. L'apprentissage non supervisé, quant à lui, dispose de données qui ne sont pas préalablement catégorisées. L'objectif de l'apprentissage non supervisé est de trouver une classification adaptée à la variable dépendante dans les données (Amatriain et al., 2011).

Plus proche voisin (Nearest Neighbors - KNN)

KNN est une méthode d'apprentissage supervisée. Considérons un point à classifier, KNN trouve les K points les proches de celui-ci à partir du jeu de données d'apprentissage. Ensuite, le point est catégorisé selon la classe à laquelle appartiennent la majorité des points les plus proches de lui.
L'un des avantages de ce classificateur est que son concept est fortement relié à celui du CF : trouver des utilisateurs ayant des intérêts semblables (ou des items similaires) est essentiellement équivalent à trouver les voisins les plus proches d'un utilisateur ou un item donné. Cette méthode est une **méthode heuristique basée sur la mémoire** ce qui implique qu'il faut refaire le calcul pour chaque itération de recommandation impliquant un changement dans les données. (Amatriain et al., 2011)

Arbres de décision

L'apprentissage par les arbres de décision est une méthode de prédiction dont le but est de prédire la valeur d'une variable cible à partir de la valeur de plusieurs variables d'entrées. Un arbre permet de classer des enregistrements par division hiérarchiques successives en sous classes. L'objectif est d'obtenir des classes homogènes, en couvrant au mieux les données.
Un arbre de décision se compose de **feuilles**, qui représentent les valeurs de la variable cible, et **d'embranchements**, qui correspondent à des tests sur des combinaisons de variables d'entrée qui mènent aux valeurs des variables cibles. Une variable d'entrée est sélectionnée à chaque **nœud interne** de l'arbre. Chaque arête vers un nœud-fils correspond à un ensemble de valeurs que peut prendre une variable d'entrée, de manière à ce que l'ensemble des arêtes vers les nœuds-fils couvre toutes les valeurs possibles de la variable d'entrée.
Usuellement, les arbres de décision sont construits en divisant l'arbre du sommet vers les feuilles en choisissant à chaque étape la variable d'entrée qui réalise le meilleur partage de l'ensemble d'objets, soit en minimisant le taux d'erreur de classification ou en maximisant l'homogénéité des sous-ensembles, c'est ce qu'on appelle le critère de coupe. Selon la technique, les arbres obtenus sont plus ou moins larges, plus ou moins profond, balancés ou non, élagués ou non, etc... Pour que la méthode soit efficace, il faut éviter de fractionner exagérément les données afin de ne pas produire des groupes d'effectifs trop faibles, ne correspondant à aucune réalité statistique.
Il existe différents types d'arbres de décision qui considèrent plusieurs facteurs, dont le nombre de branches à chaque nœud (binaires, n-aires), le type d'attributs (discrets, continus), type d'élagages (ex : bottom up), le critère de coupe…
L'avantage principal des arbres de décisions est qu'ils sont faciles à interpréter et à générer. Ceci permet de comprendre le processus de prédictions et donc de mieux expliquer les phénomènes. Les arbres de décisions peuvent être utilisés pour les approches basées sur le modèle en SR. Cependant, ils sont extrêmement difficiles à utiliser pour la construction d'un modèle quand le nombre de variables est trop élevé. On peut

donc utiliser des arbres de décisions pour créer un modèle à partir d'une ou plusieurs sous-partie(s) des données.

Classifieur de Bayes (Bayesian classifiers)

La classification bayésienne se base sur une approche probabiliste pour résoudre des problèmes de classification. Elle se base sur la définition de la probabilité conditionnelle et le théorème de Bayes.
Le principe est que l'on veut prévoir le futur (probabilité d'avoir une valeur de sortie donnée) à partir du passé (les valeurs de sortie réelles historiques).
À chaque hypothèse, on associe une probabilité (probabilité d'être la solution), l'observation d'une (ou de plusieurs) instance(s) peut modifier cette probabilité, on peut parler de l'hypothèse la plus probable, au vu des instances (Liefooghe, 2012).
La formule de Bayes s'écrit comme suit :

$$P(k/d) = P(d/k)\ P(k) / P(d) \qquad (4)$$

Avec :
P (d) a probabilité qu'un élément ait d pour description,
P(k) la probabilité qu'un élément de P soit de classe k,
P(d/k) la probabilité qu'un élément de classe k ait d pour description
P(k/d) la probabilité qu'un élément ayant d pour description soit de classe k.

Le classificateur bayésien naïf :
Chaque enregistrement est un tuple que l'on peut définir comme suit **X = (x1, x2, ...xn) sur R (A1, A2, ...An).** L'objectif est de le classer parmi m classes C1, ..., Cm et cela en l'assignant à la classe la plus probable, qui est la maximisation de

$$P(Ci/X) = P(X/Ci) * P(Ci)/P(X) \qquad (5)$$

P(X) est supposé constant (équiprobabilité des tuples), on cherche donc la classe qui maximise :

$$P(X/Ci) * P(Ci) \text{ pour i= 1 à m} \qquad (6)$$

P(Ci) est déduit de l'échantillon défini comme tel : Taille(Ci)/ Taille(E_{ch})
P(X/Ci) est approché comme suit $\prod_k P(x_k/Ci)$ (On suppose l'indépendance des attributs)
P(x_k/Ci) est estimé par variable nominale = Taille (t=x_k de Ci) /Taille(Ci) sachant que l'on considère une distribution gaussienne si les variables sont continues (Amatriain et al., 2011).

Les réseaux bayésiens :
Les réseaux bayésiens considèrent les nœuds comme des variables aléatoires. La structure quant à elle se compose de parents et de descendants dans un graphe direct acyclique de dépendance. Les variables non liées dans le graphe sont indépendantes. L'instanciation des variables non classées permet de calculer la probabilité des classes en appliquant les calculs classiques de probabilités ainsi que le théorème de Bayes. Les réseaux bayésiens sont pratiques lorsqu'on connaît la structure du réseau (relation entre les variables).

Réseau de neurones (Artificial Neural Network - ANN)

ANN est un ensemble de nœuds interconnectés et de liens pondérés, qui a été inspiré de l'architecture biologique du cerveau. Les nœuds en ANN sont appelés des neurones, en analogie aux neurones biologiques. Ces unités simples composent un réseau qui est capable d'apprendre à résoudre des problèmes de classification une fois qu'ils ont appris un modèle à partir d'une base de données d'apprentissage. Un ANN peut avoir plusieurs couches. Ces couches sont soit des couches d'entrées, des couches cachées, ou des couches de sorties. Les unités dans la couche d'entrée sont les données qui alimentent le système. Les couches cachées reçoivent les sorties pondérées des unités d'entrées et les unités de sorties correspondent aux unités de sortie des couches cachées et génèrent le résultat final du réseau. Il y a plusieurs architectures de réseau, mais nous ne rentrerons pas dans les détails. L'avantage principal des ANN est qu'ils fonctionnent pour des classifications non linéaires. Les ANN comme les Réseaux bayésiens peuvent être utilisés de façon similaire dans les SRs (Amatriain et al., 2011).

(ii) La segmentation

Les algorithmes de segmentation sont utilisés pour les mêmes raisons que la réduction de dimensionnalité. L'intérêt étant de réduire le coût de calcul et de minimiser le nombre d'objets pour lesquels il faut calculer les distances en les regroupant en segment.
La segmentation est un apprentissage non supervisé, qui consiste à assigner des items à des groupes de façon à avoir des groupes homogènes. L'homogénéité est calculée de la même façon que la similarité (section 2.5.2). Il existe deux catégories majeures d'algorithmes de segmentations : la segmentation hiérarchique et le partitionnement.
La segmentation hiérarchique regroupe successivement des éléments dans des segments (clusters), produisant un ensemble de grappes imbriquées, organisées en arborescence hiérarchique. Le partitionnement, quant à lui, divise les données en segments distincts de façon à ce que chaque objet se retrouve dans un unique segment.
L'algorithme de segmentation le plus utilisé est **k-means**. Cet algorithme de partitionnement a besoin de connaitre le nombre de classes, de façon à ce que chaque objet soit assigné à un segment homogène.
Il existe plusieurs variantes à l'algorithme K-means qui, malgré sa popularité, présente plusieurs limitations. Parmi les limitations de k-means : l'algorithme suppose une connaissance des données et donc la capacité de choisir le nombre de segments adapté (k). Les résultats que l'algorithme génèrent à chaque fois qu'il est relancé sur le même jeu de données ne sont pas nécessairement stables. Cela est dû au choix des points de départ de l'algorithme qui peut être déterminé de différente façon. De plus, il peut y avoir des résultats avec des segments vides. D'autres limitations relatives aux données diminuent la performance de l'algorithme, par exemple : la taille des segments, la densité des données, les couches dans les données.
Dans le contexte des systèmes de recommandations, (Xue et al., 2005) présente k-means comme une méthode de prétraitement des données qui permet d'aider à la formation du voisinage. Ils ne restreignent pas le voisinage de l'utilisateur au groupement (cluster) auquel il appartient, mais utilisent plutôt la distance de l'utilisateur aux différents centroïdes des clusters comme étape de présélection pour les voisins. Ils mettent également en œuvre une technique de lissage à base de grappes dans laquelle les valeurs manquantes pour les utilisateurs d'un cluster sont remplacées par des représentants de grappes.
La littérature ne fait pas état de l'utilisation d'alternatives au K-means comme les méthodes hiérarchiques (agglomérative ou par division), algorithmes basés sur la densité, les méthodes

par grilles, pour la recommandation (Amatriain, 2011). Cela s'explique par la facilité d'utilisation de k-means.

(iii) Les règles d'associations

Le but des règles d'associations est de découvrir des relations qui prédisent l'occurrence d'un item en se basant sur les occurrences d'autres items.

"an itemset is a collection of one or more items (e.g. (Milk, Beer,Diaper)). A k-itemset is an itemset that contains k items. The frequency of a given itemset is known as support count (e.g. (Milk, Beer, Diaper) = 131). And the support of the itemset is the fraction of transactions that contain it (e.g. (Milk, Beer, Diaper) = 0.12). A frequent itemset is an itemset with a support that is greater or equal to a minsup threshold. An association rule is an expression of the form X ⇒Y, where X and Y are itemsets. (e.g. Milk,Diaper ⇒ Beer). In this case the support of the association rule is the fraction of transactions that have both X and Y. On the other hand, the confidence of the rule is how often items in Y appear in transactions that contain X.

Given a set of transactions T, the goal of association rule mining is to find all rules having support ≥ minsupthreshold and confidence ≥ mincon f threshold." (Amatriain, 2011)

Calculer le support et la confiance de chaque règle d'association, afin d'éliminer celles qui ne répondent pas à la condition énoncée. Afin de limiter le coût de traitement, on commence par calculer la première condition soit : support ≥ minsupthreshold qui représente les items fréquents, ensuite on calcule les règles de confiance pour chaque itemset.

Il existe différentes techniques pour trouver les items fréquents, la plus répandue est A-priori. Elle soutient que si un itemset est fréquent, alors tous ses sous-groupes doivent être fréquents. Ensuite, il faut trouver tous les sous-groupes qui satisfont la confiance minimum requise.

Dans les systèmes de recommandation, cette approche est semblable au filtrage collaboratif basé sur les items qui lui est plus flexible, car il n'implique pas de définir des seuils explicites. Cependant, on peut tout de même trouver des SRs basé sur les règles d'associations dans la littérature, comme ceux développés par (Mobasher et al., 2001), (Smith et al., 2005) ou (Lin et al., 2004).

Cette section, présente un aperçu non exhaustif des techniques de data mining utilisées dans les SRs classiques. En effet, grâce aux plateformes open source et aux forums spécialisés, différentes communautés d'utilisateurs du web (programmeurs, physiciens, statisticiens, économistes…), développent de nouvelles méthodes et les partagent sans restriction sur le web sans pour autant les publier.

2.5.3 Techniques de filtrage

A) Content-based filtering (CBF)

De façon très simple, la recommandation basée sur le contenu (content-based filtering - CBF) tente de définir les préférences des utilisateurs et de les comparer avec les caractéristiques des items disponibles.

Comme on peut le lire dans (Bobadilla et al., 2013) "Content-*based filtering makes recommendations based on user choices made in the past (e.g. in a web-based e-commerce RS, if the user purchased some fiction films in the past, the RS will probably recommend a recent fiction film that he has not yet purchased on this website). Content-based filtering also generates recommendations using the content from objects intended for recommendation, therefore, certain content can be analyzed, like text, images and sound. From this analysis, a similarity can be established between objects as the basis for recommending items similar to items that a user has bought, visited, heard, viewed and ranked positively."*

Cette approche se fait donc sur trois étapes, dans un premier temps : il faut former les profils des items et les profils des utilisateurs. Les profiles des items sont composés de leurs caractéristiques. Les profiles des utilisateurs sont composé par les caractéristiques des items précédemment appréciés par les utilisateurs.

Dans un second temps, on effectue la prédiction de l'appréciation de l'utilisateur pour tous les items non évalués par celui-ci.

Finalement, on construit la liste de recommandation avec les N items ayant les meilleures prédictions.

Les caractéristiques des items sont généralement tirées de contenu textuel (Description de produits, articles de journaux, site internet…) ce qui explique que le CBF découle principalement de deux disciplines : la recherche d'information et le filtrage d'information. Ces disciplines permettent d'établir les profils des items à partir de contenu textuel. La « *Term Frequency /Inverse document frequency (TF-IDF)* » présentée à la section suivante est la méthode la plus utilisée pour la découverte de caractéristiques à partir de textes.

Comme expliqué précédemment la deuxième étape du CBF est d'utiliser les profils des utilisateurs pour estimer l'appréciation de ceux-ci pour les items non évalués dans le passé. Cette étape est traditionnellement effectuée de deux façons avec :

- Des heuristiques se basant sur la similarité (section 2.5.2) entre les caractéristiques des items déjà consultés/acquis/évalués et tous les autres items disponibles.

  Ou

- Des méthodes de prédictions sophistiquées comme les **classificateurs bayésiens**, **les analyses de grappes**, **les arbres de décision** et **les réseaux de neurones artificiels** qui calculent la probabilité que l'utilisateur cible apprécie chacun des items.

La différence entre les deux méthodes est que l'une, calcule les prédictions grâce à des heuristiques comme les mesures de similarités. Tandis que les autres techniques, se basent sur l'apprentissage de model à partir des données disponibles en utilisant des modèles statistiques et des algorithmes d'apprentissage machine (Adomavicius et al., 2005).

La méthode de recommandation basée sur le contenu est rarement utilisée seule dans les systèmes de recommandation commerciaux. Généralement, elle est utilisée en combinaison avec d'autres méthodes de recommandations, car elle nécessite des données précises sur les caractéristiques des items, ce qui n'est pas nécessairement disponible pour tous les produits proposés en e-commerce.

Term Frequency /Inverse document frequency (TF-IDF)

Term frequency/inverse document frequency est l'une des méthodes les plus utilisées pour bâtir les profils des items. La TF-IDF est définie traditionnellement dans le contexte des SR comme suit :

« *N est le nombre total de documents qui peuvent être recommandé à l'utilisateur*

K_j *les mots clés qui apparaissent dans* n_i *d'entre eux.*

$f_{i,j}$ *est le nombre de fois que le mot clé* k_i *apparait dans le document* d_j

$TF_{i,j}$*, Le terme de fréquence (ou de fréquence normalisé) du mot clé* k_i *dans le document dj est défini par :*

$$TF_{i,j} = \frac{f_{i,j}}{Max_z f_{z,j}} \qquad (7)$$

Où le maximum est calculé sur les fréquences $f_{z,j}$ de tous les mots clés k_z qui apparaissent dans le document d_j. Cependant, les mots qui apparaissent fréquemment dans de nombreux documents (ex : le, la, un, beaucoup…) ne sont pas utiles pour distinguer un document pertinent d'un document non pertinent. Par conséquent, la mesure de la fréquence de document inverse (IDF_i) est souvent utilisée en combinaison avec la fréquence de terme simple TF_i, la fréquence du document inverse pour le mot clé ki est généralement définie comme suit :

$IDF_i = Log\frac{N}{Ni}$ (8)

Le poids TF-IDF pour le mot clé k_i dans le document d_j est défini comme suit :

$$W_{i,j} = TF_{i,j} \times IDF_i \qquad (9)$$

Le contenu du document peut être défini comme (Adomavicius, 2005) :

$$contenu(d_j) = (w1, \ldots, wk_j)$$

Le contenu(d_j) sert à bâtir le profil des items sur un site d'e-commerce. Le profil de l'utilisateur est bâti dans les mêmes termes que ceux des caractéristiques des items et la TF-IDF permet de calculer les poids relatifs des caractéristiques préférées par l'utilisateur.

B) Filtrage collaboratif

Le filtrage collaboratif (CF) est la méthode la plus répandue dans le monde de la recommandation. Comme son nom l'indique, cette approche est à caractère collaboratif. Elle se base sur l'hypothèse du « stéréotype » qui implique qu'un utilisateur faisant partie d'un groupe de personnes ayant eu des préférences semblables dans le passé, auront des goûts semblables aux leurs. Le CF tente donc de prédire l'appréciation d'un utilisateur type pour des items en se basant sur les préférences passées d'autres utilisateurs qui sont considérés semblables à lui. Cette méthode permet de faire des recommandations sans connaître les caractéristiques des items ou des utilisateurs, ce qui rend son implantation bien plus simple que celle du filtrage basé sur le contenu.
(Breese, 1998) propose une taxonomie largement acceptée aujourd'hui, qui divise les méthodes de recommandation du filtrage collaboratif en deux catégories soit : le contenu collaboratif basé sur la mémoire (memory based CF – MBCF) et le contenu collaboratif basé sur les modèles (Model based collaborative filtering) :

i) Memory-based methods / heuristic-based (MBCF)

Les méthodes basées sur la mémoire sont des méthodes qui n'agissent que sur la matrice des notes des utilisateurs pour les items (user x item) et utilisent toute évaluation (rating/preference) générée avant le processus de recommandation (la matrice doit régulièrement être mise à jour avant la compilation).
On peut voir un utilisateur comme un vecteur dont les caractéristiques sont des items et dont les appréciations (implicites ou explicites) sont les valeurs données aux caractéristiques.

Pour ce type de recommandation, les méthodes utilisées sont des heuristiques se basant sur le calcul des similarités, des corrélations ou des distances pour recommander à l'utilisateur cible les items qui sont les plus appréciés par les utilisateurs les plus proches de celui-ci.
Afin que les résultats ne soient pas biaisés par les différences de perceptions dans les jugements. (Certains utilisateurs ont tendance à dévaluer ou à surévaluer) on corrige nos prédictions pour tenir compte du biais. Une façon de faire est de calculer les prédictions comme suit :

$$Prédiction\ d'évaluation = \bar{u}_a + k\sum_i^n w(a,i)(v_{i,j} - \bar{v}_i) \qquad (10)$$

$\bar{u}_a$ *étant la moyenne des évaluations pour l'utilisateur cible.*
k *est un facteur de normalisation dont la somme est égale à 1.*
n *est le nombre d'utilisateurs considéré pour la recommandation*
w(a,i) *sont les poids des évaluations qui peuvent être la similarité, la corrélation ou la distance.*
$v_{i,j}$ *sont les appréciations de l'utilisateur i pour l'item j*
$\bar{v}_i$ *est la moyenne des appréciations de chacun des utilisateurs*

De cette façon, peu importe la tendance de l'utilisateur, le fait de considérer les biais, permet d'améliorer les prédictions.
La distinction entre le CBRS et le MBCF est que le premier créé un profil pour l'utilisateur et calcule les similarités entre le profil de l'utilisateur cible et tous les items disponibles en se basant sur les caractéristiques des items. Tandis que le deuxième, utilise une matrice Items x utilisateurs pour le calcul des similarités entre les utilisateurs en se basant sur leurs appréciations des items. Suite à cela, une prédiction des appréciations pour les items est estimée par une moyenne pondérée des appréciations des utilisateurs avoisinant l'utilisateur cible (Bobadilla, 2013).
Les approches basées sur la mémoire peuvent être classées en deux types principaux (Sharma, M. et Mann,S., 2013) :

Pour les systèmes basés sur l'utilisateur (**user based systems**), la similarité entre les utilisateurs est calculée en comparant leurs évaluations du même item. L'appréciation pour l'élément j par l'utilisateur i est donc calculée en tant que moyenne pondérée des appréciations de j par des utilisateurs similaires à l'utilisateur i.
Pour les systèmes basés sur les items (**item-based systems**), la similarité entre deux éléments est déterminée en comparant la note du même utilisateur i pour les items. Ensuite, l'appréciation prédite de l'item j par l'utilisateur i est obtenue en tant que moyenne pondérée des appréciations de i pour les items, pondérée par la similarité entre ces items.

ii) Model based methods (MCF)

Les techniques basées sur les modèles fournissent des recommandations en estimant les paramètres de modèles statistiques. Dans un premier temps, les appréciations passées des utilisateurs pour des items sont collectées et utilisées pour apprendre un modèle. Ce modèle est ensuite utilisé pour faire des prévisions d'appréciation qui sont généralement rapides et précises. Parmi les modèles les plus utilisés, il y'a les classificateurs bayésiens, les réseaux de neurones, les systèmes de logiques floues (fuzzy systems), les algorithmes génétiques, les factorisations de matrice (matrix factorization), facteurs latents (latent features) (Adomavicius et al., 2005).

iii) Memory-Based vs. Model-Based Algorithms

La distinction principale entre les deux méthodes est que celle basé sur la mémoire utilise toujours toute les données pour faire les prédictions (ce qui est coûteux), tandis que celle basé sur le modèle utilise les paramètres définis grâce à l'apprentissage pour des prédictions. Ce qui se traduit, pour les modèles basés sur la mémoire, par une utilisation importante de l'espace de stockage en tout temps. Tandis que pour les méthodes basées sur le modèle, les prédictions se font grâce au modèle préalablement appris ce qui se traduit par une utilisation des données bien moins conséquente et donc des résultats plus rapides.

iv) Autres méthodes communément utilisées dans la littérature

Il existe d'autres méthodes de recommandation qui ont été utilisées dans les différentes études. Ces recommandations sont généralement des sous types des recommandations présentées précédemment. Des exemples seraient :

Filtrage démographique : Identifie les utilisateurs qui sont démographiquement similaire à l'utilisateur cible afin de prédire les évaluations des items.

Filtrage basé sur l'utilité : Identifie à partir des items précédemment évalué par l'utilisateur, les caractéristiques qui décrivent les préférences de ce dernier et trie les items selon leurs utilités pour l'utilisateur cible.

Filtrage basé sur la connaissance : Identifie à partir des descriptions des items et de leurs caractéristiques les besoins de l'utilisateur. Les items dont les caractéristiques correspondent sont recommandés à l'utilisateur cible (Burke, 2002).

C) Méthodes hybrides

Souvent, pour avoir des systèmes plus performants en recommandation, il est plus intéressant de combiner différentes méthodes. Les systèmes de recommandation récents sont souvent une combinaison des deux méthodes ou plus avec des choix de paramètres différents selon les besoins. La combinaison de système de recommandation permet également de pallier à plusieurs des problèmes types (section 2.6) auxquels font face les systèmes de recommandions.

La littérature fait état de sept types de recommandation hybride classique présentée ci-dessous :

Weighted Cette technique utilise plusieurs méthodes de recommandation séparément. La recommandation des items est faite en faisant la moyenne des scores attribués à chaque item existant dans chacune des listes de recommandation produites. Les items ayant les meilleurs scores sont choisis pour la recommandation finale.

Switching Cette technique ne combine pas différents résultats de systèmes de recommandations, mais choisit un système à la fois de façon dynamique selon un critère relatif au type de recommandation que l'on veut produire (exemple recommandation à court terme VS recommandation à long terme). Ce critère peut être difficile à déterminer. On peut également choisir l'approche selon un critère de confiance de la recommandation s'il y'a lieu.

Mixed Cette technique utilise plusieurs méthodes de recommandation en parallèle et présente un mélange des items recommandés par les différentes méthodes. La sélection des items candidats se fait en demandant à chaque système de délivrer à ses candidats une note associée, une note prédite et/ou un indice de confiance. Ensuite, un module spécifique effectue un mélange de ces recommandations avec un tri et une sélection basée sur les scores associés aux éléments candidats. Cette méthode d'hybridation est seulement citée par (Burke, 2007) et n'est pas évaluée.

Feature combination Cette technique utilise des données normalement utilisées pour un type de système de recommandation dans un autre contexte. Par exemple, on peut utiliser les données des évaluations des utilisateurs sur les éléments normalement traités par un système basé sur le contenu.

Feature augmentation Cette technique rajoute des données caractéristiques des utilisateurs et des items avant de les utiliser comme données d'entrée d'un système de recommandation.

Cascade Le procédé d'hybridation en cascade est une méthode hiérarchique dans laquelle chaque méthode de SR raffine une recommandation obtenue par une méthode de recommandation utilisée précédemment. Par exemple, le système EntreeC (Burke, 2002), qui fournissait trop d'items avec des scores identiques, a été amélioré en ajoutant un post-classement basé sur une recommandation collaborative.

Meta Level Cette méthode utilise comme entrée un modèle fait par un autre SR. Comparé au Feature augmentation, cette méthode nécessite un remplacement total du modèle d'entré par la sortie de la recommandation précédente.

2.5.4 Métriques d'évaluations

A) La précision des évaluations

Les systèmes de recommandations sont généralement mis en place pour prédire les intérêts futurs des utilisateurs. Il existe plusieurs méthodes pour évaluer les performances du système de recommandation, les deux métriques les plus utilisées sont la Mean absolute Error (MAE) et la RMSE (Root mean squared error). Elles permettent de comparer les prédictions aux résultats réels :

Si $r_{i\alpha}$ est la vraie valeur de l'appréciation de l'utilisateur I pour l'objet α. $\tilde{u}_{i\alpha}$ est la valeur prédite de l'évaluation et E^p est l'ensemble des évaluations non utilisées pour l'apprentissage, MAE et RMSE sont défini comme suit :

$$\text{MAE} = \frac{1}{|E^p|}\sum_{(i,\alpha)\epsilon E^p}|r_{i\alpha} - \tilde{r}_{i\alpha}| \quad (7)$$

$$RMSE = \sqrt{\left(\frac{1}{|E^p|}\sum_{(i,\alpha)\epsilon E^p}(r_{i\alpha} - \tilde{r}_{i\alpha})^2\right)} \quad (8)$$

Plus les MAE et RMSE sont faibles, meilleure est la précision de l'évaluation. Ces métriques ne sont pas optimales quand il s'agit d'aide à la découverte, car toutes les évaluations sont traitées de la même façon peu importe leurs positionnements (ranking) sur la liste de recommandations. Elles sont tout de même très utilisées vu leurs simplicités.

B) Corrélation pour l'évaluation et le classement

Une autre façon d'évaluer la précision d'une prédiction est de calculer la corrélation entre les valeurs prédites et les évaluations réelles. Les trois méthodes les plus connues sont Pearson corrélation (Pearson product-moment correlation coefficient), Spearman corrélation (Spearman, 1904) et Kendall's Tay (kendall, 1938).

La Pearson corrélation mesures l'étendu de la relation linéaire présente entre deux ensembles d'évaluations.

$$PCC = \frac{\sum_{\alpha}(\tilde{r}_\alpha - \bar{\tilde{r}})(r_\alpha - \bar{r})}{\sqrt{\sum_{\alpha}(\tilde{r}_\alpha - \bar{\tilde{r}})^2}\sqrt{\sum_{\alpha}(r_\alpha - \bar{r})^2}}, \quad (9)$$

Avec $r_\alpha$et $\tilde{r}_\alpha$ étant la vraie valeur et la valeur prédite de l'évaluation

Le coefficient de la **Spearman corrélation** ρ est défini de la même manière que la Pearson corrélation, excepté que $r_\alpha$et $\tilde{r}_\alpha$ sont remplacé par le classement des items respectifs.

Spearman et Kendall, mesurent l'étendue de l'accord des deux classements sur l'exactitude des valeurs des évaluations. **Kendall** est défini comme suit :

$$\tau = (C - D) / (C + D) \qquad (10)$$

***C** est le nombre de paires concordantes d'objets que le système prédit dans l'ordre classé correct*

***D** le nombre de paires-paires discordantes que le système prédit dans le mauvais ordre.*

τ = 1 lorsque le classement vrai et prédit sont identiques et τ = -1 quand ils sont exactement opposés.

Dans le cas où il existe des objets ayant des valeurs vraies ou prédites égales, une variation du Tau de Kendall existe et est proposé dans (J.L. Herlocker, et al. 2004).

C) Précisions des métriques de comparaison

Cette métrique est principalement utilisée quand il n'y a pas d'évaluation explicite « ratings », c'est-à-dire que les données d'entrées sont implicites. En d'autres termes, cette métrique est utilisée quand nous sommes capables de savoir quels objets ont été préférés par les utilisateurs, mais pas à quel degré d'appréciation. Quand on a une liste à recommander, le seuil utilisé pour la recommandation est ambigu ou variable. Il existe différentes métriques pour évaluer ce genre de recommandation. Les plus connues seraient Area Under ROC Curve, Precision and Recall. Nous nous intéresserons aux deux dernières, car ce sont les mesures qui évaluent le nombre d'objets pertinents recommandés dans le haut de la liste de recommandations (Top L objets).

Pour un utilisateur i la précision(precision) et le rappel (recall) de la recommandation, sont donnés respectivement par :

$$\mathrm{Pi(L)} = \frac{\mathrm{di(L)}}{\mathrm{L}} \qquad \mathrm{Ri(L)} = \frac{\mathrm{di(L)}}{D_i} \qquad (11)$$

di(L) *est le nombre d'objets pertinents qui ont été acquis/appréciés par i et qui sont dans le haut de la liste de recommandations* **L.**

Di *est le nombre total de I pertinents.*

En faisant la moyenne de la précision et du rappel sur tous les utilisateurs qui ont au moins un objet significatif dans la liste de recommandations, on obtient, la précision P(L) et le rappel R(L) moyens.

Ces valeurs peuvent être comparées à la précision et au rappel résultant d'une recommandation aléatoire, amenant à l'amélioration de la précision et du rappel défini comme suit :

$$e_p(L) = P(L)\frac{MN}{D} \qquad e_r(L) = R(L)\frac{N}{L} \qquad (12)$$

Avec M et N étant le nombre d'utilisateurs et d'objets, respectivement, et D est le nombre total d'objets pertinents. Tandis que la précision a tendance à baisser quand L augmente, le rappel augmente. C'est ce qui explique l'utilisation d'une métrique combinatoire qui est moins dépendante du L appelé F_1-score.

$$F_1(L) = \frac{2PR}{P+R} \qquad (13)$$

Cette métrique combinatoire est généralement favorisée pour éviter la confusion liée à la taille de la liste de recommandation et à son effet sur la précision et le rappel.

D) Métriques du Rang dans la liste de recommandations

Tous les utilisateurs ne prennent pas le temps d'inspecter toute la liste de recommandations, d'où l'importance de mesurer la satisfaction en tenant compte de la position de chaque objet pertinent dans la liste de recommandations et d'attribuer des pondérations selon le classement. Il existe plusieurs métriques pour évaluer cet aspect dont : *half life utility, Discounted cumulative gain, rank biased precision*.

Couverture (Coverage)

Cette métrique mesure le pourcentage d'objets proposés sur le catalogue d'items qui sont recommandé par le système. Elle peut également être considérée comme une mesure de la diversité, car plus la couverture de la longue traîne est élevée plus les recommandations sont diversifiées.

$$COV(L) = {}^{N_d}/_{N} \qquad (14)$$

N_d est le nombre total d'items qui est recommandé par le système à l'ensemble des utilisateurs.

N est le nombre d'item total du catalogue.

E) Autres métriques

Les SRs peuvent être évalué d'autres façons : en considérant des métriques de performances relatives à l'impact des SRs sur les ventes, sur les relations avec la clientèle ou encore sur l'impact d'un SR sur la prise de décisions d'achats par les utilisateurs (Senecal et al., 2003).

2.6 Chalenges

Cette section est adaptée de (Linyuan et al., 2012), qui font un découpage des chalenges les plus importants auxquels font face les SR. Elle est augmentée par les conclusions de (Adomavicius et al., 2005) et (Bobadilla et al., 2013) dans leurs revues de littérature respectives.

2.6.1 Data sparsity

L'une des caractéristiques du e-commerce est que la diversité des produits existants est très importante. Les produits offerts sur le catalogue d'un site de e-commerce peuvent se compter par millions. La problématique liée à cela est que dans tous les systèmes de recommandations, les items que l'on tente d'évaluer sont bien plus nombreux que les items déjà évalués par les clients. Ce qui implique que les données sont clairsemées. Ceci rend la tâche de calcul de similarité très difficile et influence grandement la qualité des recommandations. Pour pallier à ce problème, une des solutions serait d'utiliser les profils des utilisateurs pour calculer les similarités. Les profils peuvent être incrémentés par différentes informations disponibles sur l'utilisateur. Des exemples de ces données seraient les données sociodémographiques. La prise en compte de ces données permet de faire des recommandations basées non seulement sur les appréciations communes, mais également sur le contexte sociodémographique. Les recommandations utilisant ces informations sont celles que nous avons nommées précédemment « le filtrage démographique ». Une seconde manière serait d'utiliser une méthode de réduction de dimension comme la Singular Value décomposition (**section 2.5.2**). D'autres méthodes existent comme celle présentée par (Huang, 2004) qui propose de rajouter d'autres données à la matrice utilisateur/item, en explorant « les relations d'interaction transitive », pour l'éviter le problème de clairsemance (sparsness).

2.6.2 Cold start

Le problème de démarrage à froid (cold start) est le phénomène qui survient quand un SR peine à fournir des recommandations dues à un manque d'informations initiales sur les appréciations des utilisateurs. Il touche principalement le filtrage collaboratif. Il existe différents types de problèmes de démarrage à froid. Les plus communs sont les problèmes de démarrages à froid de nouvelle communauté (**New community**), de nouvel utilisateur (**new user**), et de nouvel item (**new item**).

Généralement, la solution à ces problèmes est de combiner plusieurs méthodes de recommandations (CF-content based RS, CF-demographic based RS, CF-social based RS). En guise d'exemple, la combinaison du CBF et du CF peut être utilisée pour inclure les caractéristiques des items et les évaluations simultanément dans le calcul des recommandations. Des questions explicites peuvent aussi être posées aux utilisateurs pour avoir des informations générales sur ces derniers. Un suivi des comportements des utilisateurs sur d'autres sites est également possible si l'on veut acquérir des données sur les utilisateurs chez des tiers partis tel « syndicated data suppliers » ou « internet service providers » (User-centric clickstream) (Bucklin et Sismeiro, 2009). L'exploration des interactions transitives présentées comme solution au data sparness peut aussi être une solution au cold start (Leung et al., 2008).

2.6.3 Scalability

Le coût de traitement des données (scalabity) est un facteur important pour la réussite d'un système de recommandation. En effet, la majorité des sites de e-commerce proposent des millions d'items à des millions de clients différents. Le volume de données à traiter est donc extrêmement élevé. La solution à ce problème est d'utiliser des algorithmes peu demandant et/ou de paralléliser le traitement des données. Une autre approche serait d'utiliser des algorithmes incrémentaux qui n'utilisent pas la totalité des données à chaque itération des recommandations (Bobadilla, 2009).

2.6.4 Diversity vs. Accuracy.

L'un des buts principaux des systèmes de recommandation commerciaux, si ce n'est le but ultime, est l'augmentation des ventes. La diversité des produits est un facteur important pour l'augmentation des ventes. Si les produits niches ne sont pas recommandés, le but du système de recommandation n'est pas atteint (Schafer, Konstan, et Riedl, 1999). Pour créer une diversité dans le catalogue de produits à recommander, il faut éviter de recommander uniquement les produits populaires. Des recommandations intéressantes doivent contenir des produits moins évidents qui sont difficilement accessibles aux utilisateurs et qui n'aurait probablement pas été considérés par ses derniers. La diversité des SRs peut être forcée par des études plus poussées sur les préférences des utilisateurs. Une méthode est présentée par (Ziegler et al., 2005), qui balance et diversifie les recommandations personnalisées dans le but de refléter le spectre complet des intérêts de l'utilisateur. L'utilisation de SR hybride apporte encore une fois une solution à ce problème en proposant des recommandations découlant de plusieurs techniques différentes.

2.6.5 The value of time.

Un des chalenges les plus importants de la recommandation est de faire les bonnes recommandations au bon moment. La plupart des SRs négligent cet aspect critique de la recommandation qui varie grandement selon le type de produit à suggérer. Prenons l'exemple de la recommandation de voyage, elles se basent sur des intérêts à court terme (beaucoup d'intérêt pour la Thaïlande avant le voyage, plus d'intérêt au retour du voyage). Les recommandations d'informations sur un journal électronique se basent principalement sur des intérêts à long terme (une personne qui s'est beaucoup intéressée à la politique dans le passé devrait s'y intéresser longtemps). La prise en compte du temps dans les recommandations est un sujet de recherche d'actualités (Xiang, 2010).

2.6.6 Interface / Interactivité

Une interface de recommandation peut jouer un rôle important dans l'acceptation des recommandations par les utilisateurs. En effet, des études ont démontré que la transparence dans les recommandations était un facteur clé dans l'acceptation de celle-ci. Les utilisateurs sont plus ouverts aux recommandations quand ils sont capables de comprendre pourquoi un item spécifique leur a été recommandé (Sihna, Swearingen, 2002). (He et al., 2016), dans leurs états de l'art sur les interfaces des systèmes de recommandations, donnent un aperçu du rôle que les interfaces et les visualisations des recommandations peuvent jouer dans l'acceptation des recommandations. De plus, ils démontrent que les interfaces ont un impact significatif sur certaines métriques de performances des systèmes de recommandations.

Les défis auxquels font face les systèmes de recommandations sont principalement liés au choix de la technique de recommandation et au type/état/volume de données disponibles. Dans plusieurs cas, l'utilisation d'une méthode hybride peut permettre de contourner certains chalenges quand les données disponibles le permettent.

3 CONCLUSION

Dans cette revue de littérature, nous avons tenté d'établir un portrait clair des SRs. Plusieurs définitions ont été présentées avant de poursuivre sur les objectifs et les fonctionnalités. Nous avons ensuite expliqué le fonctionnement des méthodes de recommandation, qui a inclut une présentation du type de données traitées, des techniques de data mining utilisées, des différentes techniques de filtrage et des métriques d'évaluation. La section finale a présenté les défis que rencontrent les SRs avec certaines des solutions utilisées pour les contourner.

La diversité des techniques et les choix des métriques impliquent une prise de décision influençant la qualité des recommandations ainsi que les coûts associés à la conception, la mise en place et l'utilisation du SR. Dans nos travaux futurs, nous tenterons d'établir un lien entre les diverses variables relatives à la production, à la livraison, à la localisation des produits, aux coups des matières premières afin d'en tenir compte dans la recommandation de produit afin de maximiser le profit de l'entreprise en plus de répondre aux besoins de l'utilisateur.

4 REFERENCES

Adomavicius, G., & Kwon, Y. (2012). Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering, 24(5), 896-911.

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering, 17(6), 734-749.

Amatriain, X., Jaimes, A., Oliver, N., & Pujol, J. M. (2011). Data mining methods for recommender systems. In Recommender Systems Handbook 39-71. Springer US.

American Marketing Association (2008). Definition of marketing. Marketing news.

Bossenbroek, H. & Gringhuis, H. (2015), Recommendation in e-commerce, Luminis Recommendation Services. Tiré de : https://www.luminis.eu/wp-content/uploads/2015/08/White-Paper-Recommendation-in-e-commerce.pdf

Buckley, C., & Voorhees, E. M. (2005). Retrieval system evaluation. TREC: Experiment and evaluation in information retrieval, 53-75. Cambridge: MIT press.

Bucklin, R. E., & Sismeiro, C. (2009). Click here for Internet insight: Advances in clickstream data analysis in marketing. Journal of Interactive Marketing, 23(1), 35-48.

Bughin, J., Doogan, J., & Vetvik, O. J. (2010). A new way to measure word-of-mouth marketing. McKinsey Quarterly, 2, 113-116.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.

Candillier, L., Meyer, F., & Boullé, M. (2007, July). Comparing state-of-the-art collaborative filtering systems. In International Workshop on Machine Learning and Data Mining in Pattern Recognition 548-562. Springer Berlin Heidelberg.

Chunfang, G., & Zhongliang, G. (2015, July). Innovation of enterprise profit patterns based on Big Data. In Logistics, Informatics and Service Sciences (LISS), 2015 International Conference on 1-5. IEEE.

Dann, S. (2010). Redefining social marketing with contemporary commercial marketing definitions. Journal of Business Research, 63(2), 147-153.

Demiriz, A. (2004). Enhancing product recommender systems on sparse binary data. Data Mining and Knowledge Discovery, 9(2), 147-170.

Friedman, N., Geiger, D., and Goldszmidt, M., Bayesian network classifiers. Mach. Learn., 29(2-3):131–163, 1997.

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. Numerische mathematik, 14(5), 403-420.

He, C., Parra, D., & Verbert, K. (2016). Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. Expert Systems with Applications, 56, 9-27.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Transactions on Information Systems (TOIS), 22(1), 116-142.

J.L. Herlocker, J.A. Konstan, K. Terveen, J.T. Riedl(2004), Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 5–53.

Jolliffe, I.T. 2002, Principal Component Analysis. Springer.

Kantor, P. B., Rokach, L., Ricci, F., & Shapira, B. (2011). Recommender systems handbook.

Kendall, M. G. (1938). A new measure of rank correlation. Biometrika, 30(1/2), 81-93.

Kim, J., Suh, E., & Hwang, H. (2003). A model for evaluating the effectiveness of CRM using the balanced scorecard. Journal of Interactive Marketing,17(2), 5–19.

Kotler, P. (2009). Marketing management: A south Asian perspective. Pearson Education India.

Lathia, N., Hailes, S., & Capra, L. (2008, March). The effect of correlation coefficients on communities of recommenders. In Proceedings of the 2008 ACM symposium on Applied computing 2000-2005. ACM.

Lee Rodgers, J., & Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. The American Statistician, 42(1), 59-66.

Léger, P.-M., Pellerin, R. et Babin, G. (2011) Readings on Enterprise Resource Planning,Laboratoire ERPsim : HEC Montréal, 332 pages (ISBN : 978-0-9866653-3-2), chapitres 15.

Lendrevie, J., & Lévy, J. (2014). Mercator 11e édition : Tout le marketing à l'ère numérique. Dunod.

Liu, E. (2001). CRM in e-business era. In CRM conference.

Shin, H. W., & Sohn, S. Y. (2004). Segmentation of stock trading customers according to potential value. Expert Systems with Applications, 27, 27–33.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K., & Zhou, T. (2012). Recommender systems. Physics Reports, 519(1), 1-49.

Marketing. (s. d.). Dans Dictionnaire Larousse en ligne. Tiré de http://www.larousse.fr/dictionnaires/francais/marketing/49526

Meyer, F. (2012). Recommender systems in industrial contexts. arXiv preprint arXiv:1203.4487.

O'Brien, J. M. (2006). You're soooooooo predictable. *The Best of Technology Writing 2007*. University of Michigan Press

Pathak, B., Garfinkel, R., Gopal, R. D., Venkatesan, R., & Yin, F. (2010). Empirical analysis of the impact of recommender systems on sales. Journal of Management Information Systems, 27(2), 159-188.

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. Machine learning, 27(3), 313-331.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In The adaptive web 291-324. Springer Berlin Heidelberg.

Sharma, M., & Mann, S. (2013). A survey of recommender systems: approaches and limitations. International Journal of Innovations in Engineering and Technology, 2(2), 8-14.

Sinha, R., & Swearingen, K. (2002). The role of transparency in recommender systems. In CHI'02 extended abstracts on Human factors in computing systems. 830-831. ACM.

Spearman, C. (1904). The proof and measurement of association between two things. The American journal of psychology, 15(1), 72-101.

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in artificial intelligence, 4.

Vozalis, M. G., & Margaritis, K. G. (2006). Applying SVD on Generalized Item-based Filtering. IJCSA, 3(3), 27-51.

Wei, K., Huang, J., & Fu, S. (2007, June). A survey of e-commerce recommender systems. In *2007 International Conference on Service Systems and Service Management* (pp. 1-5). IEEE.

Xue, G. R., Lin, C., Yang, Q., Xi, W., Zeng, H. J., Yu, Y., & Chen, Z. (2005). Scalable collaborative filtering using cluster-based smoothing. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 114-121). ACM.

Ziegler, C. N., McNee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic

diversification. In Proceedings of the 14th international conference on World Wide Web (pp. 22-32). ACM.