



HAL
open science

Modélisation et contrôle de la diversité dans les systèmes de recommandation au cours du temps

Amaury L'Huillier

► **To cite this version:**

Amaury L'Huillier. Modélisation et contrôle de la diversité dans les systèmes de recommandation au cours du temps. Intelligence artificielle [cs.AI]. 2014. hal-01078276

HAL Id: hal-01078276

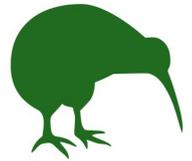
<https://inria.hal.science/hal-01078276v1>

Submitted on 30 Oct 2014

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

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

Copyright



Université de Lorraine

Mémoire de Stage

RECHERCHE

pour obtenir le diplôme de Master 2 délivré par

UFR Mathématiques et Informatique de Nancy

Sciences Cognitives et Applications
Spécialité “Traitement des médias numériques”

présenté et soutenu publiquement par

Amaury L’Huillier

le 3 septembre 2014

Modélisation et contrôle de la diversité dans les systèmes de recommandation au cours du temps

Directeur du stage : **Sylvain Castagnos**

Co-encadrante du stage : **Anne Boyer**

Jury

M. Jean Lieber,	MCF, LORIA (Orpailleur)	Parrain
Mme Christine Bourjot,	MCF, LORIA (MAIA)	Examineur
Mme Laure Buhry,	MCF, LORIA (Neurosys)	Examineur
M. Manuel Rebuschi,	MCF, Archives Henri-Poincaré	Examineur

Laboratoire Lorrain de Recherche en Informatique et ses Applications

Equipe KIWI

Campus Scientifique, B.P.239 - 54506 Vandœuvre-les-Nancy Cedex

Résumé

La quantité d'information disponible sur le web et les services informatiques associés ne cessent d'augmenter. De ce fait, il devient difficile pour les utilisateurs de trouver ce qu'ils cherchent. Certains assistants intelligents, tels que les systèmes de recommandation, ont donc été mis au point dans le but de filtrer l'information afin de proposer aux utilisateurs les items les plus pertinents par rapport à leurs besoins.

Les systèmes de recommandation peuvent utiliser des approches différentes en fonction du domaine dans lequel ils sont utilisés (e-commerce, recherche documentaire, service musical ou vidéo...). Cependant, un dénominateur commun existe pour évaluer toutes ces approches : l'évaluation de la précision des recommandations. Ainsi, un système de recommandation sera jugé performant s'il est capable de proposer à un utilisateur un item que ce dernier est susceptible d'apprécier. Pourtant, un item peut être apprécié dans l'absolu mais peut ne pas convenir si ce dernier n'est pas proposé au bon moment, dans le bon contexte ou de la bonne manière. A titre d'exemple, de récentes études ont démontré qu'il est indispensable de prendre en compte les facteurs humains dans les recommandations et qu'une différence de précision entre deux algorithmes de recommandation n'est pas toujours perçue par les utilisateurs Jones (2010). Parmi les facteurs humains existants, le besoin en diversité est l'un des facteurs dont l'importance a récemment été démontrée. Dans le cadre d'un site de e-commerce, Castagnos et al. (2010) proposent un modèle permettant de maximiser la satisfaction de l'utilisateur courant en injectant de la diversité en début et en fin de session. Cependant, les systèmes actuels ne sont pas capables d'anticiper les fins de sessions et ils ne savent donc pas quand proposer de la diversité. Par ailleurs, proposer des recommandations trop diverses au mauvais moment peut provoquer une méfiance de la part des utilisateurs envers le système Castagnos et al. (2013).

Ainsi, afin de mieux satisfaire le besoin en diversité des utilisateurs, nous avons conçu le premier modèle permettant de modéliser l'évolution de la diversité au cours du temps. Ce modèle peut être utilisé afin de détecter des événements représentatifs du comportement des utilisateurs. L'événement principalement détecté est le changement de contexte, tant que la diversité des items consultés par un utilisateur n'évolue pas ou peu, nous considérons que l'utilisateur se trouve dans un seul et même contexte. A l'inverse, un fort apport en diversité au cours d'une séquence de consultation sera considéré comme un changement de contexte. En nous appuyant sur le fait que la diversité est plus importante en début et en fin de session, nous avons posé l'hypothèse selon laquelle les changements de contexte détectés peuvent en partie être expliqués par le passage d'une session à une autre. Pour autant, le contexte n'est pas invariant au sein d'une même session : il peut exister plusieurs contextes dans une même session qui sont également détectés par notre modèle. Au final, notre modèle peut être utilisé afin de modéliser la diversité, prédire, anticiper et expliquer le comportement des utilisateurs. Nous avons validé la stabilité de notre modèle sur la base d'un corpus musical comportant plus de 200.000 écoutes et nous avons démontré qu'il peut être paramétré afin de privilégier la détection de certains événements en particulier. Nous envisageons par la suite d'utiliser nos travaux dans le but d'adapter la diversité des recommandations en fonction du besoin en diversité qui peut à présent être déterminé grâce à notre modèle.

Mots-clés : diversité, analyse des usages, contexte, recommandation, facteurs humains, modélisation utilisateur.

Table des matières

Introduction	1
1 État de l'art	3
1.1 Les systèmes de recommandation	3
1.1.1 Le filtrage collaboratif	4
1.1.2 Le filtrage par contenu	6
1.1.3 Limites des systèmes de recommandation et facteurs humains	7
1.2 La diversité	9
1.2.1 Définition et rôle	9
1.2.2 Intégration de la diversité dans les systèmes de recommandation	10
1.3 Le contexte	12
1.3.1 Définition	12
1.3.2 Contexte et systèmes de recommandation	13
1.4 Discussion	14
2 Modèle DANCE : Diversity And Natural Context Elicitation	16
2.1 Modélisation temporelle de la diversité	16
2.1.1 Notation	16
2.1.2 Description du modèle	16
2.1.3 Corpus utilisé	17
2.1.4 Résultats	19
2.2 Détection du contexte : approche naïve	22
2.2.1 Paramètres utilisés	23
2.2.2 Résultats	23
2.2.3 Discussion	24
2.3 Étude de la stabilité et optimisation des paramètres	24
2.3.1 Utilisation d'un algorithme génétique	25
2.3.2 Paramètres utilisés	26
2.3.3 Résultats	27
2.3.4 Discussion	29
3 Conclusion et perspectives	30
3.1 Conclusion	30
3.2 Perspectives	30
Bilan personnel	31
Bibliographie	32
Annexes	36
A Définition des attributs Echonest	36

Introduction

L'équipe KIWI (Knowledge Information and Web Intelligence) est l'une des 28 équipes de recherche du LORIA (Laboratoire Lorrain de Recherche en Informatique et ses Applications). Les travaux de recherche de cette équipe ont pour objectif d'améliorer la qualité du service rendu par un système de recherche et d'accès à l'information tels qu'un moteur de recherche, un intranet d'entreprise, une bibliothèque de ressources numériques, un espace numérique de travail. Les recherches menées par cette équipe portent sur la modélisation de l'utilisateur à travers ses usages, la personnalisation des services proposés et l'évaluation de la qualité de ces services. Afin d'y parvenir, différentes techniques principalement issues de l'Intelligence Artificielle (I.A.) numérique sont utilisées tels que les modèles stochastiques, statistiques, graphiques ou encore les modèles à base de grammaires.

Sur Internet, la quantité d'information à laquelle les utilisateurs ont accès ne cesse d'augmenter et de ce fait, il devient nécessaire de les assister afin de leur permettre de trouver rapidement l'information qu'ils recherchent Castagnos (2008). Pour ce faire, des systèmes de recommandation ont été développés afin de proposer (généralement sous forme de liste) aux utilisateurs des items susceptibles de les intéresser. Ces systèmes se basent sur les usages des utilisateurs proches de l'utilisateur courant (celui que l'on cherche à assister) ou sur les items similaires à ceux appréciés par ce dernier. Grâce à des techniques d'Apprentissage Automatique empruntées au domaine de l'I.A, les systèmes de recommandation ont permis de fournir des recommandations de plus en plus précises. Cependant, alors que la précision des recommandations s'approche de ses limites, certaines études mettent en avant la nécessité de prendre en compte les facteurs humains afin de mettre au point des systèmes holistiques Jones (2010). En effet, une recommandation peut être précise sans pour autant plaire à l'utilisateur si elle n'est pas faite au bon moment ou si ce dernier la juge inadaptée au contexte de sa recherche. Ainsi, un bon système de recommandation se doit de prendre en compte les facteurs humains afin de satisfaire au mieux les utilisateurs et ainsi gagner leur confiance, ce qui les amènera à utiliser de nouveau ce système.

La problématique de ce stage concerne le besoin en diversité qui a récemment été mis en avant comme étant une dimension importante dans les systèmes de recommandation. Il a été prouvé que proposer des recommandations diverses améliore la satisfaction des utilisateurs car cela permet de proposer des alternatives intéressantes par rapport aux items suggérés Smyth and McClave (2001). Récemment, Castagnos et al. (2010) ont démontré que le besoin en diversité varie chez les utilisateurs au cours du temps lors d'une démarche d'achat sur un site de e-commerce. Ils ont observé que le besoin en diversité est plus important en début et en fin de session d'achat. Cependant, à l'heure actuelle, la fin d'une session ne peut pas être anticipée ou détectée suffisamment tôt et de ce fait les recommandations ne peuvent pas être adaptées en fonction de ce besoin en diversité. De plus, Castagnos et al. (2013) ont établi que la diversité pouvait provoquer la méfiance des utilisateurs si cette dernière n'est pas proposée au bon moment. Mon travail au cours de ce stage a donc consisté à concevoir un modèle permettant de modéliser l'évolution de la diversité au cours du temps en analysant les usages des utilisateurs, et à offrir une approche permettant de détecter et de caractériser le contexte dans lequel se trouve un utilisateur.

Le mémoire est organisé en trois parties. Dans la première partie, un état de l'art est établi afin de présenter au lecteur les systèmes de recommandation, les travaux sur la diversité et le

contexte. Le vocabulaire et la syntaxe que nous utiliserons dans ce mémoire sont également explicités dans cette section. La deuxième partie présente le modèle que nous avons conçu afin de modéliser l'évolution de la diversité au cours du temps. Nous présentons également les résultats que nous avons pu obtenir avec ce modèle afin de détecter des événements caractéristiques des utilisateurs à travers leurs usages à partir d'un corpus de consultations musicales issues de l'application de *scrobbling*¹ du site Lastfm.fr². Les résultats sont proposés selon deux approches : une première dite "naïve" dans laquelle notre modèle n'est pas paramétré et une seconde utilisant un algorithme génétique afin de tester les différents paramètres de notre modèle et de vérifier sa stabilité. Au travers des ces résultats, nous testerons la validité de notre modèle de plusieurs manières. Tout d'abord en terme de complexité et de preuve de concept sur un corpus de grande taille. Par la suite, nous chercherons à trouver une correspondance entre les changements de contextes détectés et des événements en particuliers tels que les fins de sessions. L'objectif étant de voir si il est possible d'expliquer les changements de contexte, et si l'hypothèse de départ Castagnos et al. (2010) tirée du e-commerce se vérifiait dans un autre domaine. Enfin, la dernière partie est consacrée à la conclusion sur les travaux que nous avons effectués et sur les perspectives de recherche envisagées, notamment par rapport à l'intégration de notre modèle dans les phases de recommandations en situation réelle.

Les Hypothèses étudiées au cours de ce mémoire sont les suivantes :

- **H1** : l'évolution de la diversité peut être modélisée au cours du temps.
- **H2** : ce modèle peut être utilisé avec un corpus musical afin de détecter certains événements tels que les nouvelles sessions d'écoute, les musiques qui n'ont pas été écoutées en entier et les changements de contexte.
- **H3** : ce modèle est stable et peut être paramétré pour privilégier la détection de certains événements en particulier.

1. Le scrobbling consiste à installer sur un ordinateur personnel une application permettant de récupérer des informations sur les musiques écoutées par l'utilisateur en vue de lui proposer des recommandations et de partager ses listes de lectures avec d'autres utilisateurs.

2. <http://www.lastfm.fr/>

1 État de l'art

De plus en plus présents sur les sites internet, les systèmes de recommandation sont devenus des outils indispensables pour les utilisateurs. Leur but est simple : filtrer l'information disponible afin de proposer aux utilisateurs des items qu'ils sont susceptibles d'apprécier. Dans cette partie, nous présentons plus en détail ces méthodes de recommandation ainsi que leurs limites. Nous expliquons ensuite pourquoi le besoin en diversité est un facteur humain qui joue un rôle majeur dans de tels systèmes. Enfin, étant donné que notre sujet s'intéresse également à la notion de contexte, nous nous attachons à donner une définition de ce terme et à comprendre son rôle et son intégration dans les systèmes de recommandation.

1.1 Les systèmes de recommandation

Actuellement utilisés dans de nombreux domaines tels que le e-commerce, la recherche documentaire, les sites d'écoutes de musiques en ligne ou encore les services de VOD³, les systèmes de recommandation assistent les utilisateurs dans leurs démarches de recherche d'information. La recommandation peut, soit être désirée par l'utilisateur (par exemple un utilisateur qui a aimé le film "Matrix" cherche des films qui sont similaires), soit simplement être proposée dans une démarche proactive, sans que l'utilisateur ne fasse de recherche. Dans le premier cas, il s'agit de *recommendation-seeking* c'est-à-dire que l'utilisateur cherche à obtenir des recommandations alors que dans le deuxième cas, il s'agit de *recommendation-giving* car la recommandation sert d'assistance à l'utilisateur Jones and Pu (2007).

Quelle que soit la méthode de filtrage de l'information utilisée, il existe trois étapes primordiales au bon fonctionnement d'un système de recommandation Castagnos (2008) :

- collecter les traces d'usages des utilisateurs ;
- transformer et synthétiser ces traces afin qu'elles soient exploitables et utilisables pour la création du profil utilisateur ;
- utiliser les profils utilisateurs afin de trouver les recommandations adaptées aux différents profils utilisateurs à l'aide de méthodes d'apprentissage automatique.

Les traces récupérées suite aux interactions entre l'utilisateur et le système sont des données dites brutes et leurs formes dépendent de la nature des items. Autant il est relativement facile pour un utilisateur d'attribuer une note pour un film ou un livre, autant il sera très difficile de faire de même pour un parfum car ce dernier possède des dimensions beaucoup plus complexes. Il existe donc plusieurs catégories de données collectables qui sont les usages, la listes d'actions effectuées par l'utilisateur lors de la navigation, les annotations, le goût, le contexte et les préférences. Tous les items ne peuvent pas être notés explicitement, et de ce fait un système doit être capable de quantifier l'intérêt apporté par un item à un utilisateur afin de pouvoir établir son profil. Les données brutes peuvent donc être divisées en deux catégories : les données explicites et implicites. Les premières se réfèrent aux données étant explicitement fournies par l'utilisateur telles que les notes ou les critiques. Les secondes correspondent aux données calculées par le système comme le temps de consultation d'une page, d'un item ou le nombre de clics effectués par un utilisateur sur la fiche d'un produit...

Une fois les données récupérées, elles sont synthétisées afin de créer un modèle pour chaque utilisateur appelé modèle (ou profil) utilisateur ou encore modèle de préférence (dans le cas du filtrage collaboratif). Ce modèle permet d'estimer sous une forme exploitable par le système les

3. *Video On Demand*, site sur lequel l'utilisateur regarde ce qu'il désire contrairement à un programme télévisuel. Le service de VOD le plus connu est Netflix.

préférence d'un utilisateur. Grâce à ce modèle utilisateur, le système distingue les items potentiellement appréciés par l'utilisateur de ceux ne l'étant pas. Un profil utilisateur est représenté la plupart du temps par un vecteur de notes et l'ensemble des profils utilisateurs est concaténé dans une matrice utilisateur-item comme le montre la figure 1 ci-dessous.

	i_1	i_2	i_3	i_4	i_5
u_1			7	6	
u_2	7		5	6	1
u_3			6	6	
u_4	7	5			7
u_5	7	6			

FIG. 1 – Matrice Utilisateur-Item

Chaque ligne de la matrice représente le vecteur de note de chacun des utilisateurs. Grâce à la matrice de la figure 1, nous savons que l'utilisateur 2 (u_2) a attribué les notes 7, 5, 6 et 1 pour les items 1(i_1), 3(i_3), 4(i_4) et 5(i_5). Un modèle utilisateur est rarement complet : sur l'ensemble des items proposés dans un système, seul un faible pourcentage est noté par les utilisateurs.

Il existe de nombreux types de systèmes de recommandation. Dans cette section, nous présentons les deux principales familles d'algorithmes permettant d'une part, de présenter le vocabulaire nécessaire à la bonne compréhension de ce mémoire et d'autre part, de bien comprendre les enjeux de la diversité. Nous présentons donc le filtrage collaboratif dont le fonctionnement repose sur l'utilisation de la communauté d'utilisateurs du système, et le filtrage par contenu dont le principe repose sur l'exploitation des caractéristiques des items contenus dans le système. Il existe d'autres types de systèmes de recommandation tels que les systèmes basés sur les ontologies.

1.1.1 Le filtrage collaboratif

Le principe du filtrage collaboratif repose sur le postulat que des utilisateurs qui sont similaires vont apprécier les mêmes choses. Goldberg et al. (1992) ont mis au point le premier le premier système de recommandation basé sur ce principe. Afin de calculer les recommandations, le système se base donc sur les notes des utilisateurs proches de l'utilisateur que l'on cherche à recommander.

Bien que reposant sur le même principe, les algorithmes utilisés pour le filtrage collaboratif se divisent en deux catégories : les algorithmes basés sur la mémoire et les algorithmes basés sur le contenu Breese et al. (1998).

LES ALGORITHMES BASÉS SUR LA MÉMOIRE (memory-based) Resnick et al. (1994), utilisent l'ensemble des données disponibles par le système afin de calculer les recommandations et présentent l'avantage d'être très précis. Ces systèmes utilisent les matrices représentant les notes des utilisateurs et recherchent les utilisateurs les plus proches de l'utilisateur courant. On peut ainsi proposer des items appréciés par les utilisateurs qui lui sont le plus similaires. Les utilisateurs les plus proches de l'utilisateur courant sont appelés des voisins et le nombre de voisins utilisés pour calculer les recommandations dépend du système. De tels systèmes cherchent donc à prédire les notes qu'un utilisateur attribuerait à un item qu'il n'a pas encore noté en exploitant la connaissance d'une communauté. Si la note prédite pour un item est élevée, alors l'item sera recommandé. Par exemple, si l'on utilise la matrice de la figure 2 et que

l'on cherche à fournir une recommandation à l'utilisateur 3, nous devons dans un premier temps trouver les voisins de cet utilisateur. Nous constatons que l'utilisateur 2 est son plus proche voisin (avec l'utilisateur 1, mais comme l'utilisateur 1 n'a noté que des items que l'utilisateur 3 a déjà noté et il n'apporte pas d'information nécessaire à la prédiction). Nous pouvons donc prédire que l'utilisateur 3 appréciera l'item 1 et mais sûrement pas l'item 5. L'item 1 constitue donc un item à recommander à l'utilisateur 3. En situation réelle, l'utilisateur courant possède de nombreux voisins et les notes prédites sont calculées à partir de la moyenne des notes des voisins considérés.

	i_1	i_2	i_3	i_4	i_5
u_1			7	6	
u_2	7		5	6	1
u_3			6	6	
u_4	7	5			7
u_5	7	6			

FIG. 2 – Prédiction des préférences de l'utilisateur 3

Afin de connaître les voisins d'un utilisateur, le système calcule les similarités entre les utilisateurs. La similarité peut être calculée de différentes manières, à l'aide du coefficient de corrélation de Pearson (forcé ou non), la méthode des moindres carrés, le vecteur cosinus... Il est cependant admis que le coefficient de corrélation de Pearson offre les meilleures performances Shardanand and Maes (1995), Breese et al. (1998). Malgré leurs bonnes performances, ces algorithmes basés sur la mémoire sont très coûteux en temps et en puissance de calcul et, de ce fait, ils ne sont pas adaptés pour travailler sur des bases de données trop importantes engendrant des problèmes d'explosion combinatoire.

LES ALGORITHMES BASÉS SUR UN MODÈLE (model-based) sont des algorithmes, qui comme leur nom l'indique, créent un modèle de représentation des données. Un modèle permet donc une représentation des utilisateurs, des items et de leur corrélation. Les modèles peuvent être construits soit à partir d'une approche probabiliste Breese et al. (1998) (classifieur naïf de Bayes, réseaux bayésiens et arbres de décisions), à partir de clustering (algorithme de K-mean Herlocker et al. (1999), Repeated Clustering Ungar and Foster (1998), Gibbs Sampling Ungar and Foster (1998), clustering hiérarchisé Chee et al. (2001)...) ou via une matrice item-item (dont le but est de trouver des relations entre les items par rapport aux usages des utilisateurs). Un modèle étant une approximation des données, il en résulte une perte de précision par rapport aux données de base. Les algorithmes basés sur un modèle apportent un gain en temps de calcul et en mémoire utilisée pour le calcul des recommandations puisque la plupart des calculs ont été effectués lors de la création du modèle. Cependant, étant donné qu'un modèle est mis au point à partir de l'ensemble des données, ce modèle devra être recalculé lors de l'ajout de nouvelles données.

Il faut donc retenir que les algorithmes basés sur la mémoire sont très précis et très réactifs mais coûteux en temps et en mémoire. Les algorithmes basés sur un modèle sont quant à eux plus rapides et moins coûteux en mémoire mais perdent un peu en précision et le modèle doit être recalculé afin de prendre en compte de nouvelles informations.

Les algorithmes de filtrage collaboratif rencontrent, encore à l'heure actuelle, un certain nombre de difficultés comme le rappelle Su and Khoshgoftaar (2009) :

- les profils des nouveaux utilisateurs ne sont pas suffisamment bien renseignés pour que le système puisse fournir des recommandations précises. Ce problème est connu sous le nom de démarrage à froid (*cold-start*) ;
- le passage à l'échelle d'un algorithme peut s'avérer difficile dès lors qu'il faut considérer la totalité des utilisateurs et des items disponibles aux sein d'un même système. Les problèmes rencontrés se situent alors au niveau du temps de calcul ou de la mémoire nécessaire aux calculs, d'où la nécessité d'avoir parfois recours à certains modèles moins précis mais limitant les problèmes d'explosion combinatoire ;
- les goûts des utilisateurs évoluant dans le temps, consulter un même item ne doit pas forcément amener les mêmes recommandations si la consultation n'est pas faite dans le même contexte. Le système doit être suffisamment adaptatif pour ne pas proposer des recommandations sclérosées (ne jamais proposer de nouveauté) ;
- effectuer de bonnes recommandations ne doit pas se faire à l'insu de la vie des utilisateurs. Collecter de nombreuses informations sur l'utilisateur peut s'avérer néfaste pour la confiance attribuée au système par les utilisateurs ;
- les systèmes de recommandation ne sont pas à l'abri des manœuvres frauduleuses (*SPAM...*) visant à perturber leur bon fonctionnement ou à utiliser les informations sur les utilisateurs (cryptage des données) ;
- tous les items d'un système n'ont pas nécessairement été notés par un grand nombre d'utilisateurs. Le système peut donc rencontrer des difficultés à fournir des recommandations à un utilisateur voulant obtenir des items similaires à un item peu noté Lee et al. (2004).

1.1.2 Le filtrage par contenu

Le principe des algorithmes de filtrage par contenu est de recommander des items en se basant non plus sur les similarités inter-utilisateur mais sur les similarités inter-item. Ces algorithmes tirent leurs racines du champ de la recherche documentaire et les premiers systèmes s'appliquaient surtout à des domaines dont les ressources étaient de natures textuelles (journaux, pages web...). De tels algorithmes peuvent, soit se baser l'item couramment consulté, soit sur l'historique des items appréciés afin de proposer des recommandations. Certains algorithmes de filtrage par contenu cherchent à proposer des items similaires par rapport à un item cible tout en prenant en compte les préférences des utilisateurs Pazzani and Billsus (2007). Pour ce faire, le système doit posséder des informations sur les items permettant d'établir des similarités inter-items. Le processus permettant de caractériser les ressources peut être de diverses natures. Il peut être automatique dans le cas où le système extrait les caractéristiques des items en analysant le contenu. Il peut être le fruit d'annotations manuelles d'experts⁴ du domaine qui peut amener à la création d'ontologies permettant une modélisation de toutes les connaissances d'un domaine et de leurs relations. Le processus de caractérisation peut également s'appliquer sur les communautés d'utilisateurs en leur attribuant des étiquettes (*tags*) Celma (2008).

Cependant, utiliser uniquement le contenu peut ne pas suffire à proposer de bonnes recommandations, utiliser également les préférences des utilisateurs peut alors permettre d'améliorer la qualité des recommandations. Par exemple, si l'on se place dans le domaine cinématographique, les attributs utilisés pour caractériser les films sont le(s) genre(s) du film, sa date de sortie, le réalisateur, les acteurs, son box-office... Ainsi, un utilisateur ayant apprécié le film

4. Le site <http://www.pandora.com/> est réputé pour l'aboutissement du projet *Music Genome Project* où chaque musique possède plus de 400 aspects musicaux déterminés par des experts

“Avatar” pourra se voir recommander des films du même réalisateur ou du même genre selon son profil. En effet, si l'utilisateur a apprécié “Avatar” mais qu'il n'a apprécié aucun des films du réalisateur alors le système devra se baser sur des similarités en rapport avec le genre ou les acteurs du film plutôt que sur les films du même réalisateur Chen and Pu (2012). La qualité des recommandations dépend donc en grande partie du profil utilisateur. S'il n'est pas suffisamment complet, alors le système ne saura pas si il faut recommander des films du même réalisateur ou des films du même genre ou bien avec les deux attributs réunis.

Proposer des recommandations à partir de la similarité entre l'item apprécié et les items recommandés n'est pas une garantie de succès. Par exemple, sur le site IMBD⁵ le célèbre film *Highlander* (premier su nom) est noté en moyenne 7,2 sur 10 alors que la suite du film du même réalisateur et avec les mêmes acteurs n'est noté que 3,9 sur 10. Un système basé sur le contenu recommandera le deuxième opus de la saga à un utilisateur ayant apprécié le premier. Ainsi, bien que le second film soit très similaire au premier et que la recommandation soit précise, elle n'apportera pas pour autant la satisfaction de l'utilisateur. Cela souligne fait qu'il ne suffit pas de proposer des items similaires pour proposer de bonnes recommandations et que les facteurs humains doivent être intégrés dans les cycle de recommandation .

Discussion : Le filtrage par contenu et le filtrage collaboratif sont deux approches possédant chacune des avantages et des inconvénients. Afin de maximiser les avantages tout en minimisant les inconvénients, certaines méthodes hybrides utilisent une combinaison de ces deux techniques. Par exemple, dans un premier temps le filtrage par contenu est utilisé lorsqu'un item ou un utilisateur n'a pas assez de notes, puis le filtrage collaboratif est ensuite utilisé pour fournir des recommandations lorsque les profils sont suffisamment complets Burke (2002).

1.1.3 Limites des systèmes de recommandation et facteurs humains

Comme nous l'avons vu précédemment, chaque méthode de recommandation possède ses avantages et ses inconvénients. Quels que que soient leurs avantages et inconvénients, tous les systèmes de recommandation ont partagé depuis le début de leur création une caractéristique commune : ils ont tous été construits et évalués à partir de leur précision. Cependant, il existe d'autres facteurs que les systèmes de recommandation doivent être en mesure d'intégrer afin de fournir les recommandations les plus adaptées aux utilisateurs. Nous allons voir dans un premier temps que la satisfaction des utilisateurs ne dépend pas uniquement de la précision des recommandations et dans un second temps que les facteurs humains ont une part importante à jouer dans les recommandations.

Les recherches dans le domaine des systèmes de recommandation se sont longtemps attachées à minimiser la valeur de la RMSE⁶. Plus la valeur de la RMSE est basse, plus le système est capable de prédire les notes d'un utilisateur et donc plus le système sera précis dans ses recommandations. Cependant, avoir une grande précision ne suffit pas à satisfaire les utilisateurs. A titre d'exemple, les algorithmes les plus précis ayant remporté le concours Netflix⁷ ne seront pas utilisés à des fins de recommandation thenextweb.com (2012). Cela est dû, comme l'explique McNee et al. (2006) au fait que la précision des recommandations n'est pas un critère

5. <http://www.imdb.com/> est un site cinématographique international dont les bases de données de votes et de films sont couramment utilisées à des fins de recherches, notamment dans le domaine des systèmes de recommandation

6. La RMSE(*Root-Mean-Square Error.*) ou erreur quadratique moyenne est une mesure d'erreur entre la prédiction du système et les votes réels des utilisateurs.

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

exclusif pour la satisfaction des utilisateurs et que des recommandations moins précises ne sont pas moins appréciées par les utilisateurs. Par ailleurs, Jones (2010) démontre qu’une différence de précision de 10 % entre 2 systèmes de recommandation n’est pas perçue par les utilisateurs. Dans le même temps, Pu et al. (2009) ont montré que la façon de présenter les recommandations et la capacité à les expliquer aux utilisateurs peuvent doubler le taux d’acceptation des recommandations. Cela illustre bien le fait que certains facteurs humains inhérents à la prise de décision doivent être intégrés dans les systèmes de recommandation. Ainsi, certains travaux cherchent à identifier les facteurs permettant de maximiser les taux d’acceptation et d’adoption des systèmes par les utilisateurs, et donc indirectement leur satisfaction. Le fruit de ces recherches est synthétisé dans une adaptation du modèle *Technology Acceptance Model* (TAM) Davis (1989) par Jones (2010). Le modèle originel qui s’intéressait à la manière dont les utilisateurs perçoivent un système informatique a été adapté au domaine de la recommandation (Figure 3). Ce modèle TAM adapté par Jones (2010) s’intéresse à deux points en particulier qui peuvent être traduits en deux questions :

- Est-ce que le système est facile d’utilisation ? (ease of use)
- Est-ce que le système est utile pour l’utilisateur ? (usefulness)

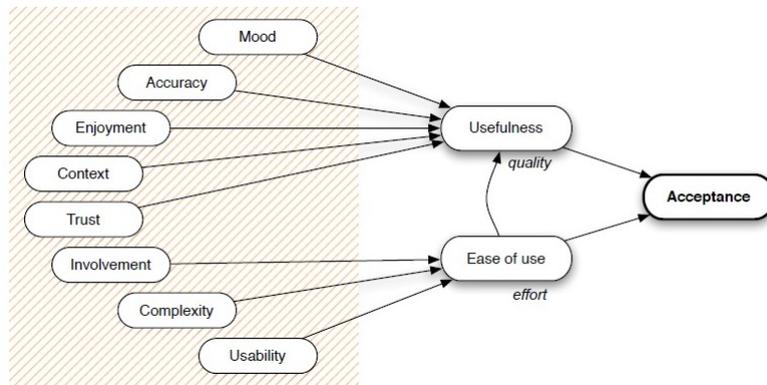


FIG. 3 – Technology Acceptance Model

A titre d’exemple, d’après les études utilisateurs réalisées par Jones (2010), la facilité d’utilisation d’un système intervient au moment de l’interaction entre l’utilisateur et le système. Si l’utilisateur ne comprend pas comment interagir avec les recommandations qui lui sont proposées, ce dernier risque de ne pas utiliser le système. Le même risque est encouru si les recommandations proposées ne sont pas adaptées à l’utilisateur. Par exemple, si un utilisateur recherche des livres sur les bases de la programmation et que le système propose des ouvrages destinés à des experts en programmation, alors le système ne sera pas utile pour l’utilisateur puisqu’il n’aura pas cerné le contexte de l’utilisateur.

Parmi les facteurs humains existants, les principaux facteurs qu’un système doit s’efforcer de prendre en compte lors de la phase de recommandation sont la confiance apportée par le système, la diversité des items proposés, la présentation des recommandations, les explications sur les recommandations (pourquoi cet item est recommandé), les facteurs culturels et le contexte. Certains travaux se sont également focalisés sur le processus de décision des utilisateurs et son intégration dans les systèmes de recommandation Chen et al. (2013).

Outre la phase de recommandations, les facteurs humains interviennent également dans la phase de constitution des profils utilisateurs. Cosley et al. (2003), ont démontré qu’il existait une inconstance dans la notation des items et qu’il pouvait y avoir jusqu’à 40 % de variations dans les notes attribuées par un utilisateur sur un même item à des moments différents. En

réaction à ce constat, Jones et al. (2011) proposent une méthode alternative à la notation qui est la comparaison (je préfère l’item A à l’item B) et obtient une meilleure stabilité dans les préférences des utilisateurs. Castagnos et al. (2013) observent que la diversité est une dimension importante à prendre en compte lors de la constitution du profil utilisateur. En effet, si les items votés par un utilisateur sont trop similaires entre eux alors les recommandations seront de mauvaises qualités (recommandations sclérosées).

Parmi les autres facteurs humains qui ont récemment été investigués, les deux principaux sont la diversité et le contexte. Étant donné que ces facteurs sont au cœur de nos travaux, ils seront détaillés dans les parties qui suivent.

1.2 La diversité

1.2.1 Définition et rôle

La diversité telle qu’elle a été définie pour la première fois dans le domaine des systèmes de recommandation par Smyth and McClave (2001) est la mesure opposée à la similarité. Plus des items sont similaires entre eux, moins il sont divers et réciproquement. La diversité a été utilisée dans un premier temps afin de proposer des alternatives aux utilisateurs dans les n -recommandations qui leurs étaient proposées. Par exemple, un utilisateur cherche une maison à louer pour les vacances et ses critères sont d’avoir 3 chambres, d’être à moins de 500 mètres de la plage et d’avoir un loyer inférieur à 450 euros par semaine. Si un système de recommandation propose cinq offres qui correspondent exactement à ces critères mais que ces cinq offres se trouvent toutes dans le même immeuble, alors si l’utilisateur n’est pas satisfait par la situation géographique de l’immeuble, aucune des recommandations ne sera utile. C’est pourquoi l’algorithme mis au point par Smyth and McClave (2001) consiste à trouver les offres les plus similaires à la demande de l’utilisateur tout en étant diverses entre elles. Cette première approche démontre donc l’importance de la diversité et certains auteurs tels que Zhang and Hurley (2008), Lathia (2010) parleront même de frustration chez les utilisateurs lorsque la diversité n’est pas suffisamment présente.

En plus de proposer des alternatives, la diversité s’avère être un moyen de réagir face à une mauvaise recommandation McGinty and Smyth (2003). En effet, si un système recommande un item qui n’est pas apprécié, le système a tout intérêt à réagir de manière à proposer un item avec des caractéristiques diverses pour éviter que la prochaine recommandation soit trop similaire à la première (ce qui risquerait de nouveau de ne pas satisfaire l’utilisateur). Au cours d’une étude utilisateur, ils démontreront tout comme Lathia et al. (2010), que la diversité ne doit pas être utilisée à chaque cycle de recommandation.

Certains auteurs distinguent la diversité extrinsèque et intrinsèque. La diversité extrinsèque permet de pallier l’incertitude due au langage. Par exemple, si un utilisateur fait des recherches sur le mot “jaguar”, ne sachant pas si l’utilisateur fait référence à l’animal ou à la voiture, proposer des recommandations diverses permet de prévoir ces deux éventualités Radlinski et al. (2009). La diversité intrinsèque permet, quant à elle, d’éviter une redondance des items recommandés Clarke et al. (2008).

Partant des travaux de Häubl and Murray (2003) sur les processus de prise de décision dans les systèmes de recommandation, Castagnos et al. (2010) s’intéressent à l’évolution du besoin en diversité au cours du temps lors d’une démarche d’achat sur un site de e-commerce. La figure 4 issue de leurs travaux nous montre que la diversité des items recommandés ne doit pas être la même en fonction de l’étape du processus d’achat dans lequel se trouve l’utilisateur.

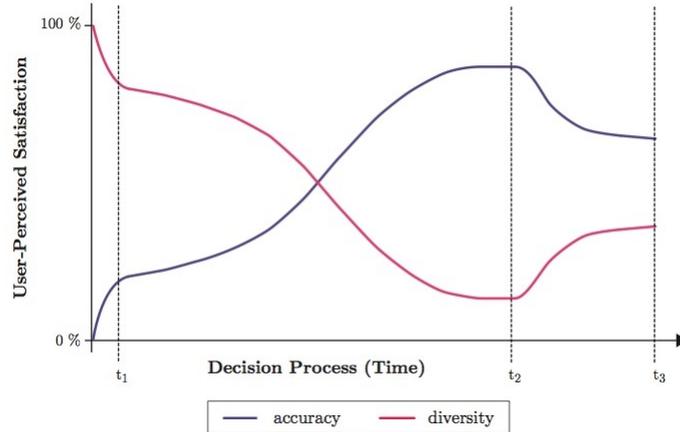


FIG. 4 – Time dependent Accuracy vs Diversity Model

Bien que Lathia et al. (2010) se soient intéressés à la prise en compte de la diversité dans le temps à l’échelle d’un profil utilisateur en cherchant à minimiser la similarité des items recommandés pour les profils les plus complets afin d’éviter de scléroser les recommandations, les travaux de Castagnos et al. (2010) sont les premiers à s’intéresser à l’évolution du besoin en diversité au cours du temps à l’échelle d’une unique session. Nous rappelons qu’il faut distinguer la diversité qui est une métrique et le besoin en diversité qui est un facteur humain. Le besoin en diversité peut être extrapolé à partir de l’étude de l’évolution de la diversité. Une grande diversité dans les items consultés indique un besoin en diversité accru et réciproquement. La figure 4 décrit l’évolution de la diversité au cours du temps et cela nous montre que le besoin en diversité est plus important dans certaines phases du processus d’achat. Lors d’une démarche d’achat, les utilisateurs commencent par “balayer” une grande gamme de produits afin de pouvoir se faire une idée des produits existants, de leurs prix et de leurs caractéristiques. C’est la raison pour laquelle le besoin en diversité en début de session est important. Par la suite, le besoin en diversité diminue à mesure que l’utilisateur commence à s’orienter vers un produit de plus en plus précis. Et enfin, le besoin en diversité augmente à la fin de la session d’achat, car cela permet aux utilisateurs de comparer le produit ou les produits ayant le plus retenu leur attention par rapport aux autres produits.

Poursuivant leurs travaux, Castagnos et al. (2013) se sont intéressés à la perception de la diversité au cours du temps et démontrent que la diversité est effectivement perçue par les utilisateurs dans les systèmes de recommandation. Leur étude démontre que, bien qu’elle semble améliorer la satisfaction des utilisateurs et permette de résoudre certains problèmes dus au démarrage à froid (*cold-start*), elle peut également être source de méfiance ou d’incompréhension dans certains cas. Ces conclusions corroborent les travaux de Jones (2010) qui a démontré l’importance de l’interface dans la présentation des recommandations. En effet, le mode de présentation ainsi que la diversité des recommandations influent sur la confiance que les utilisateurs vont accorder au système.

Ainsi, bien que la diversité dans les recommandations soit un réel besoin, la quantité de diversité à introduire, le mode de présentation et le moment où la diversité doit être maximisée posent de nombreuses questions.

1.2.2 Intégration de la diversité dans les systèmes de recommandation

Comme nous venons de le voir, la diversité apporte de nombreux avantages et nous allons à présent présenter les méthodes permettant d’introduire et d’évaluer la diversité au sein des

recommandations.

Même si elle n'est pas volontairement introduite, une certaine part de diversité est tout de même présente dans les recommandations via sérendipité⁸ Murakami et al. (2008). Par exemple, dans le cas d'un système reposant sur du filtrage collaboratif, de la diversité peut être proposée à l'utilisateur cible si les profils utilisateurs utilisés pour la recommandation sont suffisamment divers. Cependant, les méthodes visant à intégrer de la diversité dans les systèmes de recommandation sont majoritairement adaptées pour les systèmes basés sur le contenu Agrawal et al. (2009), Bradley and Smith (2001), Department and Bridge (2001), Shimazu (2001), McSherry (2002). Comme nous l'avons présenté précédemment, la diversité telle qu'elle a été proposée par Smyth and McClave (2001) est une mesure opposée à la similarité. La similarité entre deux items i_1 et i_2 se calcule à partir de la moyenne pondérée des similarités sur les n attributs communs à ces deux items, comme le montre l'équation suivante :

$$Similarity_A(i_1, i_2) = \frac{\sum_{j=1..m} w_j * sim_{attribute=j}(i_1, i_2)}{\sum_{j=1..m} w_j} \quad (1)$$

Smyth and McClave (2001) poursuivent leurs travaux et proposent deux autres mesures de diversité. Ils proposent une formule permettant de calculer la dissimilarité présente dans une classe C composée de m items dont la valeur obtenue est appelée *Diversity* :

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

La seconde formule qu'ils proposent permet de calculer la diversité apportée par un item par rapport à une classe C composée de m items, la valeur obtenue est appelé *Relative Diversity* (RD) et se calcule avec l'équation suivante :

$$RD^u(i, C) = \begin{cases} 0 & \text{si } C = \{i\}, \\ \frac{\sum_{j=1..m} (1 - sim_A(i, c_j))}{m} & \text{sinon.} \end{cases} \quad (3)$$

Par la suite, Ziegler et al. (2005) proposeront une métrique similaire afin de calculer la similarité des items contenus dans une classe C de cardinalité m . Cette métrique est appelé ILS (Intra List Similarity) et la formule permettant de la calculer est définie par l'équation ci-dessous :

$$ILS(C) = \frac{\sum_{i \in C, j=1..m-1} \sum_{i_k \in C, k=j+1..m} Similarity(i_j, i_k)}{\frac{m*(m-1)}{2}} \quad (4)$$

Bradley and Smith (2001) sont les premiers à proposer un algorithme incluant la métrique de diversité dans les recommandations. L'algorithme qu'ils ont développé est un algorithme glouton borné dont le but est de proposer aux utilisateurs des items similaires à l'item cible (la requête de l'utilisateur) tout en étant divers entre eux. Le principe de leur algorithme est le suivant : premièrement, l'algorithme sélectionne un ensemble d'item de taille K où les items sont les plus similaires à l'item cible (selon l'équation 1). Puis dans un second temps, l'algorithme procède à un reclassement des recommandations. Pour ce faire, l'algorithme crée un sous-ensemble appelé top-N (où $N < K$) en sélectionnant itérativement à chaque étape l'item offrant la meilleure qualité respectivement calculée selon l'équation 5 :

$$Quality(i, t, C) = Similarity(i, t) * RelDiversity(t, C) \quad (5)$$

8. la sérendipité est la découverte par le hasard

Utilisant ce principe de reclassement des recommandations, plusieurs adaptations seront présentées afin de proposer un top-N à la fois précis par rapport à la requête et suffisamment divers. Radlinski et al. (2009) proposent trois méthodes différentes de reclassement, Zhang and Hurley (2008) adopteront une approche binaire afin de conserver une similarité raisonnable tout en maximisant la diversité dans le top-N proposé à l'utilisateur. Certains auteurs tels que Schafer et al. (2002), Jahrer et al. (2010) parleront de méta-systèmes dont le principe repose sur la sélection d'items de différentes listes obtenues à partir de métriques de similarités différentes. L'introduction de la diversité a également été investiguée dans le cadre de systèmes à base de filtrage collaboratif. La différence majeure par rapport au filtrage par contenu est qu'il n'est pas possible de se servir des caractéristiques des items et de ce fait, les métriques de diversité sont calculées à partir des notes. La métrique de l'ILS de Ziegler et al. (2005) utilisée pour le reclassement du top-N fut la première approche visant à intégrer de la diversité dans le filtrage collaboratif. D'autres travaux tels que ceux de Said et al. (2012), Boim et al. (2011) et Castagnos et al. (2014) s'attacheront à adapter les méthodes de clustering afin de pouvoir injecter de la diversité.

De manière plus générale, Foulonneau et al. (2014) présentent certaines stratégies visant à augmenter la diversité lors de la recommandation :

- augmenter la taille de la liste des item recommandés ;
- ajouter des items choisis aléatoirement ;
- mélanger les stratégies de recommandation (systèmes hybrides) ;
- trouver des clusters proches des clusters d'intérêt d'une personne donnée, ce qui permet de recommander des items différents tout en minimisant le risque de proposer des items qui n'intéressent pas l'utilisateur Onuma and Faloutsos (2009) ;
- sélectionner des items dans la longue traîne⁹ qui aient le moins de chance de déplaire ;
- compenser le manque de notation (*sparsity*) en notant des clusters d'items plutôt que de les noter individuellement Park and Tuzhilin (2008).

Malgré l'impact positif de la diversité sur la satisfaction des utilisateurs, la dimension temporelle et donc le contexte ne sont pas pris en compte dans les systèmes actuels de recommandations à base de diversité .

1.3 Le contexte

Tout comme le besoin en diversité, le contexte est un facteur dont l'importance dans la satisfaction utilisateur a été démontrée par de nombreuses études Leake and Scherle (2001), Adomavicius et al. (2005), Palmisano et al. (2008), Sieg et al. (2007). Cependant, le contexte est une notion relativement complexe et son intégration dans les systèmes de recommandation est devenue un champ d'investigation à part entière. Nous allons donc dans un premier temps définir d'après l'état de l'art ce qu'est le contexte puis nous en présenterons les enjeux ainsi que les méthodes permettant son intégration dans les systèmes de recommandation.

1.3.1 Définition

Le contexte est un concept multidimensionnel utilisé dans de nombreux domaines tels que la médecine, les lois ou encore l'économie. De ce fait, il existe une pléthore de définitions du contexte et Bazire and Brezillon (2005) concluront après avoir examiné plus de 150 définitions :

9. La longue traîne définit pour un grand ensemble de ressources le sous-ensemble des ressources qui sont très peu consultées. La longue traîne a pour opposé la tête qui constitue les ressources réellement utilisées.

“il est difficile de trouver une définition pertinente et satisfaisante pour chaque discipline”. De manière générale, Webster (1980) définit le contexte comme étant l’ensemble des conditions et des circonstances qui affectent quelque chose. Dans le domaine des systèmes de recommandation, le “quelque chose” de la définition précédente correspond à l’action effectuée par l’utilisateur et “les circonstances et conditions” correspondent aux variables que le système cherche à déterminer. Castagnos (2008) définit ainsi le contexte dans les systèmes de recommandations “comme l’ensemble des circonstances d’utilisation et des attributs de l’usager pouvant influencer sur son comportement”.

En utilisant le contexte, un système essaye donc de comprendre pourquoi deux utilisateurs similaires ont réagi différemment face à une même recommandation. Par exemple, dans le cas d’un site de location d’appartements de vacances, si un utilisateur cherche une maison à la montagne, les recommandations choisies ne seront pas les mêmes en hiver et en été (pour des critères de recherche identiques). En effet, l’hiver les utilisateurs sont plus susceptibles de rechercher un appartement à proximité des pistes de ski alors que l’été ils cherchent sans doute à être au calme et un peu excentré du centre de la ville¹⁰. De ce fait, la saison constitue un exemple de variable permettant de déterminer le contexte.

1.3.2 Contexte et systèmes de recommandation

Tout comme il existe de nombreuses définitions du contexte, les domaines dans lesquels vont s’appliquer les recommandations vont également donner lieu à des notions de contexte différentes. Ainsi, le contexte dans le domaine du e-commerce ou des systèmes de recommandation mobiles ne sera pas défini de la même manière. Nous allons à titre d’exemple présenter le contexte tel qu’il est décrit dans chacun des deux domaines précédemment cités. Cela permet dans un premier temps de présenter quelques unes des variables qui peuvent être utilisées mais aussi de montrer que les notions de contexte peuvent être relativement différentes.

Dans les systèmes de recommandation mobiles, les variables initialement utilisées étaient l’identité des personnes se trouvant à cotés de l’utilisateur courant, les objets présents autour de ce dernier ainsi que les changements de ces deux éléments Schilit and Theimer (1994). Brown and Chen (1997) incluront par la suite la date, la saison et la température. L’humeur de l’utilisateur (également appelée statut émotionnel) est également utilisée par Dey et al. (2001) qui proposeront une définition plus fine du contexte : “toute information qui peut caractériser, et qui est pertinente dans l’interaction entre l’utilisateur et l’application”.

Dans le domaine du e-commerce et de la personnalisation des recommandations, les informations contextuelles utilisées sont différentes. Ainsi l’intention d’achat constitue une variable représentative du contexte et une intention d’achat différente donnera lieu à des actions et des choix différents selon Palmisano et al. (2008). En effet, un utilisateur ne va pas se comporter de la même manière s’il achète un produit pour lui ou pour faire un cadeau. Différents profils existent donc en fonction de l’intention d’achat et le système cherche à déterminer le profil de l’utilisateur, permettant ainsi d’augmenter la satisfaction des utilisateurs. Tout comme les systèmes de recommandation mobiles, des informations de natures plus concrètes comme l’heure, la météo et l’entourage de l’utilisateur à recommander sont également utilisées pour améliorer les recommandations dans un restaurant Oku et al. (2006).

10. On comprend alors le rôle de la diversité, puisque cette dernière permet de proposer des appartements situés à différents endroits de manière à pouvoir couvrir le maximum de cas possibles tout en étant proche de la demande de l’utilisateur.

Quel que soit le domaine d'application, la prise en compte du contexte passe par une phase importante qui est la récupération d'informations. Selon Adomavicius and Tuzhilin (2001), il existe trois moyens de collecter des informations contextuelles qui sont :

- la collecte explicite, dont les informations récupérées ont été obtenues par des interactions directes sous forme de questions. Par exemple, le site de que en ligne Stereomood¹¹ demande explicitement à l'utilisateur de renseigner son humeur afin de proposer des recommandations qui y sont adaptées.
- la collecte implicite a pour but de récupérer les informations issues des interactions avec un système comme par exemple la géolocalisation de l'utilisateur, l'heure à laquelle l'utilisateur utilise le système...
- l'inférence, dont le principe consiste à essayer de déterminer à partir de méthodes statistiques (comme les classifieurs Naïfs de Bayes) des situations caractéristiques en utilisant une base d'apprentissage. Par exemple, il est possible d'inférer l'identité (adulte, enfant...) d'une personne qui regarde la télévision en fonction de la fréquence à laquelle les chaînes sont "zappées".

Le contexte joue un rôle incontestable dans l'amélioration des performances d'un système de recommandation en terme de satisfaction utilisateur. Étant donnée la complexité de cette notion, il faut bien souvent expliciter les contextes possibles afin de les caractériser. Le plus souvent les contextes sont représentés sous une forme d'ontologie Chen and Chen (2014) afin de les utiliser dans les systèmes de recommandation. Cependant, cela nécessite une intervention humaine afin de construire l'ontologie, ce qui aboutit à des systèmes de recommandation ad hoc (non génériques, c'est-à-dire non transférables d'un domaine à un autre). Par ailleurs, connaître le contexte de recherche d'une personne peut parfois être intrusif. Par exemple, si l'on récupère l'adresse *IP* d'un compte utilisateur, il est possible d'inférer si ce dernier est chez lui ou sur son lieu de travail. Même si les recommandations proposées sont plus pertinentes en utilisant certaines informations, il ne faut pas négliger le respect de la vie privée des utilisateurs.

Suite à l'état de l'art que nous venons de présenter, nous introduisons deux nouvelles définitions du contexte que nous utiliserons dans la deuxième partie de ce mémoire :

Contexte implicite : ensemble des traits communs aux items consultés pendant un certain laps de temps.

Contexte explicite : ensemble des critères permettant de caractériser explicitement la situation dans laquelle se trouve l'utilisateur courant. Il peut s'agir de critères temporels (date et heure, jour de la semaine, période de l'année...), spatiaux (localisation, destination ...), d'éléments d'expertise (compétences, moyen à disposition...), etc.

La plupart du temps, les critères du contexte explicite sont représentés sous la forme d'une ontologie et sont assez intrusifs (manque de respect de la vie privée).

1.4 Discussion

Bien plus que de simples suggestions, les recommandations sont devenues à l'heure actuelle des aides indispensables aux utilisateurs. Après s'être uniquement focalisés sur la précision des items recommandés durant de nombreuses années, les systèmes de recommandation se centrent de plus en plus sur le réel besoin des utilisateurs et intègrent de plus en plus les facteurs humains. Parmi les facteurs humains existants, le besoin en diversité fait partie des champs

11. www.stereomood.fr

d'investigations actuels dont les travaux se sont avérés concluants et prometteurs. Proposer des recommandations diverses améliore la satisfaction des utilisateurs en permettant d'éviter les recommandations sclérosées (ne jamais proposer de nouveauté), de pallier aux incertitudes et de proposer des alternatives dans les recommandations. Cependant, les dimensions quantitative et temporelle de la diversité sont des points qui restent à éclaircir d'autant que ces derniers semblent être corrélés. En effet, comme il a été démontré, la quantité de diversité requise par les utilisateurs évolue au cours du temps. A l'heure actuelle, aucun modèle permettant de modéliser l'évolution de la diversité des items consultés au cours du temps n'a été proposé. De ce fait, répondre à la question "Comment évolue la diversité des items consultés au cours du temps?" constitue une question de recherche à laquelle nous essayerons de répondre dans la deuxième partie de ce mémoire.

Castagnos et al. (2013) ont démontré qu'il existe une relation entre certaines phases du processus de décision et la diversité désirée par les utilisateurs. Partant de cette conclusion, nous posons donc l'hypothèse selon laquelle l'évolution de la diversité des items consultés peut être utilisée afin de comprendre les usages et le contexte des utilisateurs. Comme nous l'avons défini, un contexte implicite correspond à un laps de temps pendant lequel les items consultés partagent des caractéristiques communes. Un changement de contexte implicite doit donc se traduire par un changement de diversité que notre modèle pourrait être à même de détecter. Un tel modèle pourrait permettre de détecter les changements de contexte. De plus, sachant que la diversité des items consultés augmente en début et en fin de session dans la cadre du e-commerce, il serait possible de tester si cette affirmation se vérifie dans un autre domaine. Cela suppose donc dans un premier temps de mettre au point un modèle permettant de modéliser l'évolution de la diversité au cours du temps et dans un deuxième temps de se servir de ce modèle afin de caractériser les usages des utilisateurs.

2 Modèle DANCE : Diversity And Natural Context Elicitation

Comme nous l'avons précédemment introduit, le modèle que nous proposons a pour but de modéliser l'évolution de la diversité au cours du temps et de détecter les changements de contexte en analysant les usages des utilisateurs L'Huillier et al. (2014). Nous avons choisi de nommer ce modèle DANCE, qui est l'acronyme de "Diversity And Natural Context Elicitation" car ce dernier repose sur l'analyse de la diversité en utilisant des techniques empruntées au domaine du traitement automatique des langues.

Dans cette partie, nous allons présenter dans un premier temps l'aspect modélisation temporelle de la diversité de notre modèle. Dans un deuxième temps, nous expliquerons pourquoi et comment notre modèle peut être utilisé afin de détecter et de caractériser les contextes. Nous présenterons également les performances qui ont été obtenues sur la base d'un corpus de plus de 200.000 écoutes. Enfin, nous démontrerons à l'aide d'un algorithme génétique que notre modèle est stable et qu'il peut être paramétré afin d'améliorer ses performances de détection.

2.1 Modélisation temporelle de la diversité

2.1.1 Notation

Afin de faciliter la lecture de cette partie, nous allons dans cette section définir les termes que nous allons utiliser.

$U = \{u_1, u_2, \dots, u_n\}$ est l'ensemble des utilisateurs.

C_u représente l'utilisateur courant.

$I = \{i_1, i_2, \dots, i_m\}$ est l'ensemble des items consultés par l'utilisateur courant.

$C_u = \{c_1^u, c_2^u, \dots, c_k^u\}$ est l'ensemble des items consultés par l'utilisateur courant pour un historique de taille k .

L'historique d'un utilisateur peut être décomposé sous la forme de tuple :

$\langle c_{t-h}^u, \dots, c_{t-2}^u, c_{t-1}^u, c_t^u \rangle$ où t représente le pas de temps courant.

$A = \{a_1, a_2, \dots, a_h \mid \text{card}(A) \ll \text{card}(I)\}$ est l'ensemble des attributs que possède une musique. $\text{sim}_a(c_i^u, c_j^u)$ où $a \subseteq A$ est la similarité entre deux musiques pour l'attribut a .

2.1.2 Description du modèle

Lors de son interaction avec un système informatique, l'utilisateur est amené à consulter plusieurs items au cours du temps. L'historique de ses consultations étant stocké par le système, il est donc possible de modéliser les usages des utilisateurs. L'ensemble des consultations d'un utilisateur peut être vu comme une suite d'items contigus. Les approches reposant sur un modèles de Markov d'ordre k pour estimer l'intérêt – ou observer les caractéristiques – d'un item particulier (celui en cours de consultation ou le prochain le plus probable) relativement à ses prédécesseurs apparaissent alors naturelles Bonnin (2010). Dans ce genre de modèles, seules les k dernières observations sont utilisées pour la prédiction des événements futurs. De ce fait, nous assumons que la consultation de l'item courant dépend uniquement des items précédemment consultés dans un laps de temps que nous appellerons la fenêtre d'observation (ou encore l'historique). La base de notre modèle est de calculer la diversité apportée par un item cible par rapport à une fenêtre d'observation, c'est-à-dire de calculer sa *relative diversity*

(RD). La figure ci-dessous est un exemple tiré d'un corpus de musique permettant d'illustrer le fonctionnement de notre modèle.

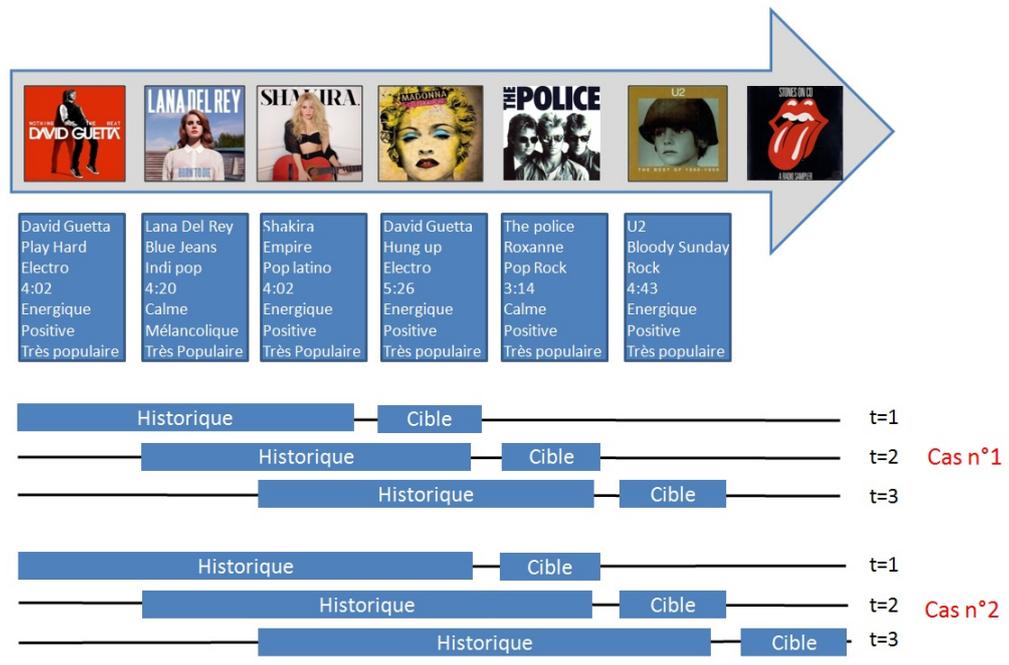


FIG. 5 – Principe du modèle DANCE

A chaque pas de temps, la RD de l'item courant est calculée par rapport à la fenêtre d'observation. La taille de la fenêtre d'observation est un paramètre qui peut être défini si l'on souhaite utiliser un historique plus ou moins grand pour calculer la RD. Par exemple, sur la figure 5, le cas n° 1 représente une fenêtre d'observation de taille 3 alors que dans le cas n° 2, la fenêtre d'observation est de taille 4. La RD d'un item peut être calculée soit avec l'ensemble des attributs soit avec un ou plusieurs attributs.

2.1.3 Corpus utilisé

Afin de tester notre modèle, nous avons utilisé l'API de lastfm.fr¹² pour récupérer les écoutes de 1 000 utilisateurs entre juin 2005 et octobre 2013. Une écoute correspond à un tuple¹³ composé du pseudonyme de l'utilisateur, du nom de l'artiste et de la musique ainsi que de la date et l'heure d'écoute sur forme d'un *timestamp*¹⁴. Ce corpus comprenait à l'origine vingt millions d'écoutes pour des utilisateurs provenant de différents pays. Dans le but de récupérer des informations sur les musiques et les artistes, nous avons utilisé l'API d'Echonest¹⁵. Cependant, avant de récupérer des informations sur les musiques, nous avons fait le choix de réduire le nombre d'utilisateurs. Ce choix a été fait dans le but de ne garder que les utilisateurs ayant un nombre d'écoutes suffisamment important et dont les informations sur les musiques pourraient

12. <http://www.lastfm.fr/api>

13. Un tuple constitue une écoute mais pas l'inverse

14. Le *timestamp* correspondant au nombre de secondes écoulées depuis le 1er janvier 1970 à minuit UTC. Cette valeur permet donc d'obtenir une valeur précise et universelle de la date et de l'heure d'un événement.

15. <http://developer.echonest.com/>

être récupérées par l’API d’Echonest¹⁶. Pour chaque musique, nous avons ainsi pu récupérer les informations suivantes (cf. annexe A pour plus d’informations sur ces attributs) :

- des valeurs réelles : la durée, le tempo, le mode (mineur ou majeur) et “l’amplitude” *loudness* de la musique, les coordonnées de la ville et les années d’activité de l’artiste ;
- des indices compris entre 0 et 1 : “l’énergie” *energy*, “la popularité” (*hotttness*) et la “dançabilité” (*dancability*) de la musique ainsi que la popularité et la familiarité *familiarity* de l’artiste ;
- des listes : les artistes similaires et les genres associés à l’artiste (rock, rap, classique...).

Le corpus utilisé pour l’ensemble des calculs effectués dans ce mémoire est composé de 100 utilisateurs ayant écoutés 204 360 musiques dont 40 877 musiques uniques et 5 558 artistes uniques. Les caractéristiques des écoutes sont résumées dans le tableau 1 ci-dessous :

Attributs	musique							artiste	
	durée	tempo	mode	loudness	energy	hotttness	danceability	hotttness	familiarity
max	4194	239	1	41,76	0,99	0,91	0,98	0,98	0,91
min	12	35	0	0,08	0,00002	0,0008	0,04	0,11	0,08
moyenne	218,10	128,27	0,57	7,69	0,76	0,33	0,43	0,61	0,63
écart type	84,28	30,30	0,49	4,19	0,21	0,15	0,17	0,12	0,13

TAB. 1 – Statistiques du corpus

La formule que nous avons utilisée pour calculer la RD est la formule telle qu’elle a été définie dans l’équation 3 (cf. supra, p.11). Nous avons dû cependant adapter les formules permettant de calculer les similarités avec les divers attributs de notre corpus. Nous avons donc utilisé les formules suivantes :

FORMULE DE SIMILARITÉ POUR LES ATTRIBUTS SOUS FORME DE LISTE

$$sim_{attribute=a}(c_t^u, c_{t-1}^u) = \frac{card(c_t^u.a \cap c_{t-1}^u.a)}{min(card(c_t^u.a), card(c_{t-1}^u.a))} \quad (6)$$

Cette formule de similarité permet de calculer le nombre d’artistes et de termes communs aux deux listes, plus ce nombre est grand et plus la similarité est importante.

FORMULE DE SIMILARITÉ POUR LES ANNÉES D’ACTIVITÉ

$$sim_{attribute=a}(c_t^u, c_{t-1}^u) = \frac{card(c_t^u.a \cap c_{t-1}^u.a)}{max(card(c_t^u.a), card(c_{t-1}^u.a))} \quad (7)$$

Tout comme la formule de similarité pour les attributs sous forme de liste, la formule pour calculer la similarité sur les années d’activité calcule le nombre d’années communes aux deux artistes.

FORMULE DE SIMILARITÉ POUR LES ATTRIBUTS BINAIRES (MODE)

$$sim_{attribute=a}(c_t^u, c_{t-1}^u) = \begin{cases} 1 & \text{si } c_{t-1}^u.a = c_t^u.a, \\ 0 & \text{sinon.} \end{cases} \quad (8)$$

16. Parmi les 1 000 utilisateurs initiaux, bon nombre d’entre eux avaient écouté des artistes inconnus par l’API d’Echonest. De nombreux noms d’artistes ou de musique étaient également erronés. Le manque d’informations sur une musique rendant impossible de calculer la RD, nous avons choisi de supprimer les utilisateurs ayant trop de musiques inconnues.

Étant donné que le mode ne prend que 2 valeurs qui sont 0 et 1, si les musiques ont le même mode alors elles sont exactement similaires (sur l'attribut mode) sinon elles sont exactement dissimilaires.

FORMULE DE SIMILARITÉ POUR LES ATTRIBUTS NUMÉRIQUES

$$sim_{attribute=a}(c_t^u, c_{t-1}^u) = e^{-10 * \left(\frac{c_t^u \cdot a - c_{t-1}^u \cdot a}{max_a - min_a} \right)^2} \quad (9)$$

Dans la formule 8, les valeurs minimum et maximum permettent de normaliser les similarités. Nous avons mis au point cette formule afin que les similarités ne soient pas uniquement dépendante de la différence existante entre les valeurs des attributs. Par exemple, une différence de durée de 5 minutes entre deux musiques ne doit pas avoir le même impact sur la similarité en fonction de la durée de ces dernières. Dans le cas où les deux musiques durent 30 et 35 minutes, la similarité entre les musiques doit être importante alors que dans le cas où les musiques durent 3 et 8 minutes, la similarité doit être plus faible car la différence de durée est importante par rapport à la durée des deux musiques.

FORMULE DE SIMILARITÉ POUR LES COORDONNÉES

$$sim_{attribute=a}(c_t^u, c_{t+1}^u) = 1 - \frac{distance(c_t^u, c_{t+1}^u)}{max\ distance} \quad (10)$$

La similarité pour l'attribut "coordonnées" se calcule à partir de la distance qui sépare les deux artistes. Cette distance est calculée à partir de coordonnées grâce à la géodésique qui permet de calculer la distance minimale entre deux points sur une surface (la terre dans notre cas). Le dénominateur correspond à la distance maximum qui existe entre deux artistes, c'est-à-dire 20 037,5 km (la moitié de la circonférence de la terre).

Remarque : Nous avons utilisé un corpus de musique car les musiques sont des items qui offrent de nombreux avantages. Premièrement, la durée de consultation des items est relativement courte (contrairement aux films par exemple) et de ce fait, les sessions sont composées d'un nombre d'items suffisamment important pour tester notre modèle de détection. Par ailleurs, les musiques possèdent de nombreux attributs (numériques pour la plupart) pouvant être obtenus via des API spécialisées. Enfin, contrairement aux ressources textuelles par exemple, une musique possède un temps de consultation qui est indépendant de l'utilisateur.

2.1.4 Résultats

Afin de modéliser l'évolution de la diversité au cours du temps, nous avons développé un programme en JAVA. Nous avons également mis au point une interface graphique nous permettant de tester diverses configurations de notre modèle. La figure 6 ci-dessous est une capture d'écran de notre programme.

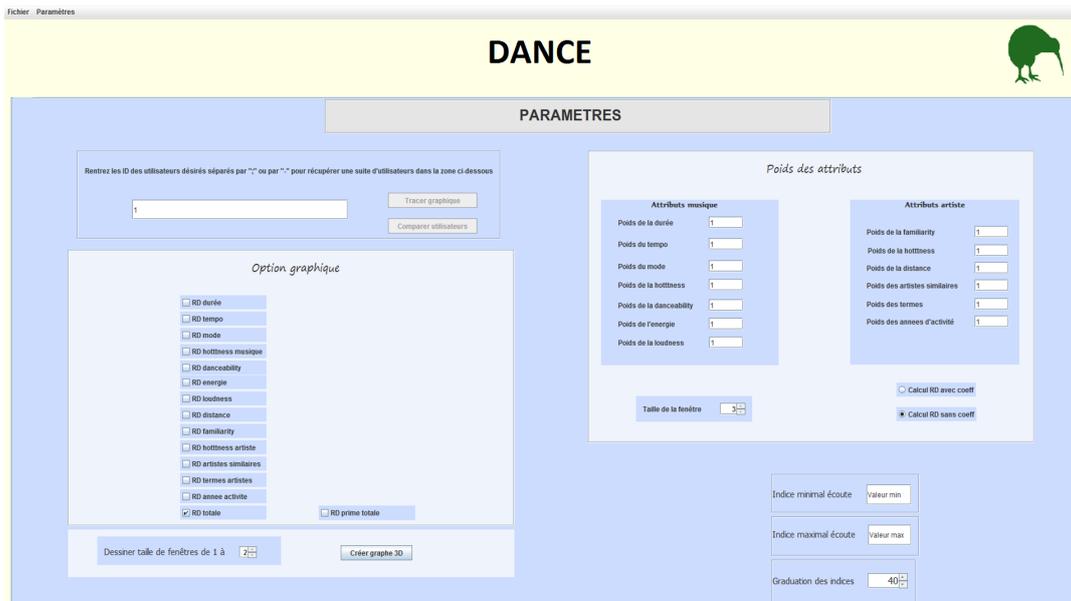


FIG. 6 – Interface du modèle DANCE

L'interface se divise en deux parties. La partie de gauche permet de sélectionner les attributs avec lesquels la RD sera calculée. Il est ainsi possible d'utiliser tous les attributs (nous parlerons alors de RD totale), de faire des combinaisons d'attributs ou de n'utiliser qu'un seul attribut. Pour chaque attribut sélectionné, il est possible de calculer la valeur des nombres dérivés de la RD (que nous appellerons RD') associée à cet attribut. La partie de droite permet quant à elle d'attribuer des poids différents aux attributs lors du calcul de la RD et de configurer la taille de la fenêtre d'observation. En utilisant les poids, il est par exemple possible de privilégier la prise en compte des artistes similaires et de diminuer celle de la durée des musiques dans le calcul de la RD.

Le graphique¹⁷ de la figure 7 ci-dessous représente l'évolution de la RD et de la RD' pour l'attribut "tempo" de l'utilisateur 1 pour 125 de ses écoutes.

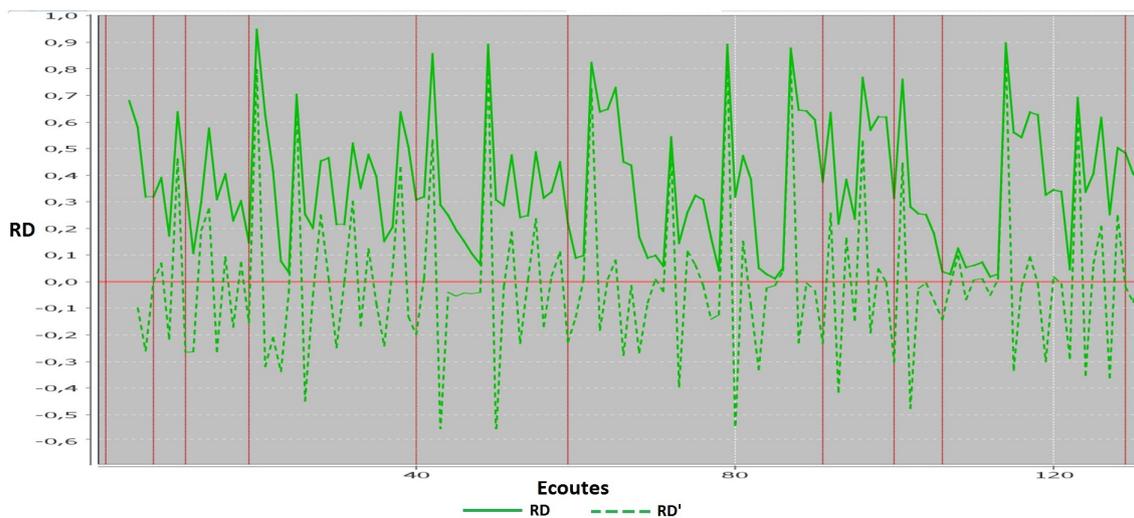


FIG. 7 – Évolution de la RD et RD' sur l'attribut "tempo" pour une fenêtre de taille 3

17. Ce graphique a été obtenu à l'aide de la librairie java *jmathplot* : <http://jzy3d.org/>

Sur ce graphique, l'axe des abscisses représente les items consultés et l'axe des ordonnées représente la valeur de la RD de chaque item. Nous pouvons constater que la RD n'est pas une valeur stable et qu'elle évolue fortement au cours du temps. Une croissance de la RD nous indique que les items consultés apportent de plus en plus de diversité par rapport à l'historique considéré et inversement. Bien que la RD soit relativement instable, nous constatons qu'il existe certains moments où la diversité oscille peu, nous indiquant que les items consultés n'apportent pas beaucoup de diversité. A l'inverse, certaines consultations font chuter ou augmenter drastiquement la RD. Lorsque la RD est basse, l'utilisateur spécialise sa séquence de consultation (c'est-à-dire que les items consultés sont très similaires). Lorsque la RD est haute, l'utilisateur a un comportement éclectique (il est ouvert à plusieurs tendances, et les items consultés sont divers).

L'évolution de la RD peut également être visualisée en trois dimensions comme le montre le graphique ci-dessous qui a été obtenu à partir de la RD totale.

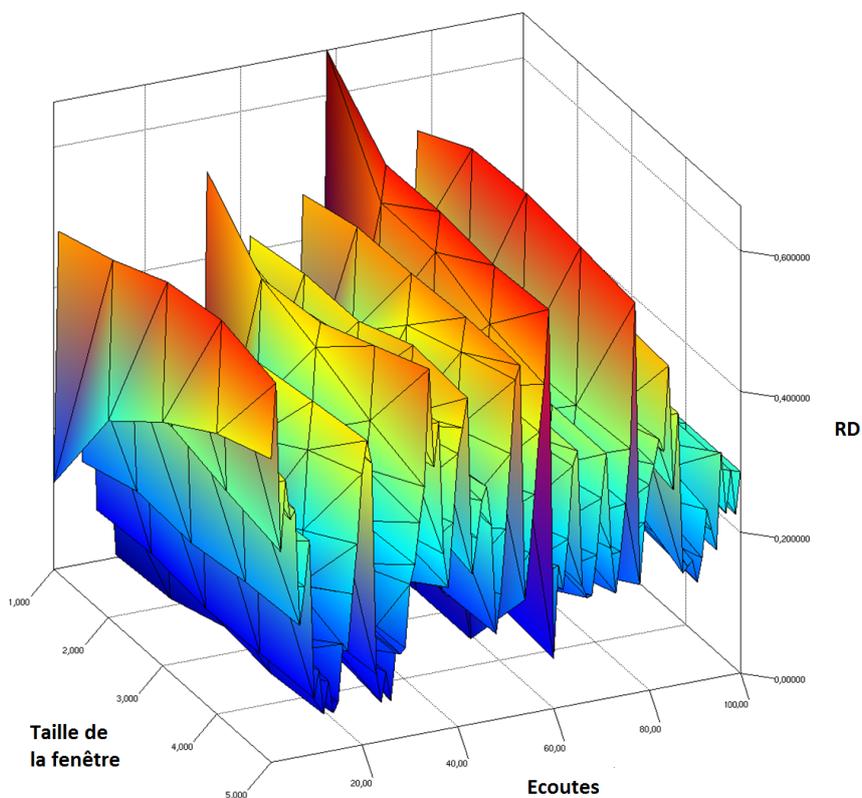


FIG. 8 – Evolution de la RD totale selon différentes tailles de fenêtres

Ce graphique nous permet de voir l'impact de taille de la fenêtre d'observation sur la valeur de la RD. La taille de la fenêtre ne semble pas modifier de manière significative les valeurs de la RD. Malgré cela, les variations de RD sont plus prononcées pour un historique de taille 1. Cela n'est pas surprenant, car si l'on calcule la diversité apportée par un item par rapport à un seul autre, cela revient à calculer la diversité entre les deux items. De manière générale, la diversité apportée par un item par rapport à un ensemble d'items est plus faible que la diversité apportée par un item par rapport à un seul item. En prenant une fenêtre d'observation trop petite, les variations trop importantes de RD risquent de générer trop de bruit alors qu'une fenêtre trop grande pourrait masquer une variation importante de diversité.

Discussion : Comme nous venons de le présenter, le modèle DANCE permet de modéliser l'évolution de la diversité au cours du temps. Cela nous permet donc de valider **H1**. Bien qu'un corpus de musique ait été utilisé, notre modèle présente l'avantage d'être générique et peut être utilisé dans d'autres domaines. Les formules de similarité par attributs entre les items devront cependant être adaptées en fonction du domaine choisi. Le modèle DANCE possède l'avantage d'avoir une complexité en temps constant $O(k)$, k étant la taille de l'historique qui est fixé (cf. section 2.1.1).

Comme nous l'avons défini précédemment, au sein d'un contexte (qu'il soit implicite ou explicite) les items consultés partagent des caractéristiques communes. A partir de notre modèle, nous pouvons savoir quand la diversité des items consultés varie fortement et nous pouvons donc utiliser ces variations afin de détecter le passage d'un contexte à un autre.

2.2 Détection du contexte : approche naïve

Un contexte étant un laps de temps pendant lequel les items consultés partagent des caractéristiques communes, la diversité apportée par les items consultés dans un même contexte est relativement faible. Au contraire, lors d'un changement de contexte, les caractéristiques des items consultés n'étant plus les mêmes, la RD apportée par les items du nouveau contexte est plus importante. C'est selon ce principe que nous avons basé notre modèle afin de détecter les changements de contexte.

Notre modèle cherche donc à détecter les items réunissant deux conditions :

1. la RD de l'item cible est supérieure à la RD de l'item précédent et de l'item suivant, ce qui signifie qu'il y a un pic de diversité ;
2. la variation de diversité apportée par l'item courant (sa RD') doit être supérieure à un seuil fixé par les paramètres de notre modèle, ce qui signifie que la diversité apportée par l'item cible est importante.

Algorithm 1 Pseudo-code simplifié du modèle DANCE

```

nb_session_detecté = 0
musiques_passées_detectées = 0
contexte_detectés = 0
k = taille de la fenêtre d'observation
pour Tous les items i faire
    Calculer RD(i,k) et RD'(i,k)
    si RD(i-1) < RD(i) > RD(i+1) et RD'(i) > seuil alors
        si i est un début de session alors
            sessions_detectées ++
        sinon si i a été passé alors
            musiques_passées_detectées ++
        sinon
            contextes_detectés ++
    fin si
fin si
fin pour
retourner nb_session_detectées, nb_musiques_passées_detectées, nb_contexte_detectés

```

2.2.1 Paramètres utilisés

Afin de pouvoir tester les performances de détection de notre modèle, nous avons calculé les sessions d'écoute de chacun des utilisateurs de notre corpus. En effet, l'API que nous avons utilisé ne nous a pas permis d'obtenir explicitement les moments où se situent les fins de sessions. Connaissant les heures exactes auxquelles les musiques ont été écoutées grâce aux *timestamp*, ainsi que les durées des musiques, nous avons pu extraire les sessions d'écoute de notre corpus. Nous considérons qu'une session d'écoute est constituée d'un enchaînement de musiques et qu'il n'y a pas d'interruption de plus de 15 minutes entre 2 écoutes. A l'aide des informations précédemment citées, nous avons également été en mesure de savoir si les musiques ont été entièrement écoutées. Nous considérons qu'une musique a été passée si cette dernière n'a pas été écoutée pendant au moins 70 % de sa durée. Notre corpus comporte 18 412 sessions d'écoute et 15 008 musiques ont été passées. Une session est composée en moyenne de 11 musiques et dure en moyenne 40 minutes.

Cette première approche est dite naïve car, les paramètres que nous avons utilisés ne sont pas optimisés. Tous les attributs ont un poids fixé à 1 pour le calcul de la RD et le seuil de RD' est fixé à 0,05 (cf. supra, 2^{ème} condition pour détecter un changement de contexte, section 2.2, p.21).

2.2.2 Résultats

Les résultats que nous avons obtenus avec les paramètres décrits précédemment sont résumés dans le tableau ci-dessous :

	Nombre existants	Nombre détecté	Pourcentage
Sessions	18 412	14 253	77,41 %
Musiques passées	15 008	2 632	17,53 %
Contexte implicite	-	12 925	-

TAB. 2 – Résultats de l'approche naïve

Cette première approche de notre modèle permet de détecter 77,41 % des sessions existantes. Bien que le nombre de sessions retrouvées soit important, il convient de préciser que notre modèle ne permet pas d'arriver à 100 % de détection. Notre modèle basé sur les changements de diversité, nous permet d'induire des changements de contexte. Or, s'il n'y a pas de changement de contexte entre deux sessions, notre modèle ne pourra pas détecter la nouvelle session. Notre modèle n'a pas pour but de détecter les sessions des utilisateurs mais les contextes. La détection des sessions nous a permis à partir du postula selon lequel il existe un changement de contexte entre deux sessions, de vérifier que l'on pouvait détecter ces changements de contexte. De plus, cela nous permet de validé dans un autre domaine le constat effectué par Castagnos et al. (2010) sur le fait que la diversité est plus importante en début et en fin de session.

Nous constatons également que le nombre de maximums locaux que nous avons obtenus (32 665, ceux correspondants à un contexte implicite et ceux correspondants à une fin de session) est supérieur au nombre de sessions existantes de notre corpus. Cela suggère ainsi qu'il existe plusieurs contextes au sein d'une même session.

Le nombre de musiques passées qui sont détectées par notre modèle est relativement faible : seuls 17,53 % de détections des musiques passées correspondent à un fort pic de diversité. Ce chiffre n'est pas très surprenant dans le sens où les raisons qui peuvent pousser un utilisateur à ne pas écouter une musique entièrement peuvent être multiples : la musique n'est pas appréciée, la musique est appréciée mais l'utilisateur n'avait pas envie de l'écouter entièrement, l'utilisateur

à envie d’écouter autre chose car cette musique lui a fait penser à une autre musique qu’il souhaite écouter... Cependant, ces résultats ne sont en rien dépréciatifs de notre modèle, nous avons juste souhaité vérifier si l’on pouvait détecter un autre type d’événement que les fins de sessions.

Outre le fait que notre modèle offre de bonnes performances en terme de détection de contextes (explicites et implicites), il est intéressant d’étudier le moment où ont été réalisées les détections de contextes explicites. Les temps auxquels ont été réalisées les détections sont résumés dans le tableau ci-dessous :

Moment détection	t - 2	t - 1	t	t + 1	t + 2	total
Nombre détection	2 894	2 647	7 793	2 187	2 403	17 924

TAB. 3 – Type de détection

Dans le tableau 3, $t+n$ signifie que nous regardons dans le futur d’un pas de temps de n et $t-n$ signifie que nous regardons dans le passé d’un pas de temps de n . Détecter une fin de session à $t+n$ revient à pouvoir l’anticiper alors que la détecter à $t-n$ revient à le détecter avec n temps de retard. Nous pouvons constater que la majorité des détections se fait au moment $t=0$, c’est-à-dire que la fin de session est immédiatement détectée. Les fins de sessions peuvent également être anticipées. Cependant, les performances d’anticipations sont plus faibles que les détections à l’instant t . Cela peut s’expliquer par le fait que le comportement d’un utilisateur ne peut être anticipé que dans une certaine mesure si l’on ne regarde que l’évolution de la diversité. Inversement, notre modèle détecte également les changements de contexte *a posteriori*. Cela est dû au fait que la diversité apportée par le nouveau contexte n’est pas caractérisée par le premier item de ce contexte mais par les suivants. Comme cela a été proposé dans Castagnos et al. (2010), il y a d’avantage de diversité en début et en fin de session avec une évolution progressive. Si l’on n’est pas capable de détecter à $t=0$, on peut alors détecter (anticiper ou mesurer après coup) très rapidement.

2.2.3 Discussion

Cette première approche du modèle DANCE nous a permis de démontrer qu’il est possible d’utiliser l’évolution temporelle de la diversité afin de détecter certains événements du comportement des utilisateurs. L’hypothèse **H2** est donc validée. En plus d’être en mesure de détecter les changements de contexte, l’étude de l’évolution de la diversité peut être utilisée dans une certaine mesure afin de prédire la fin d’une session de consultation. Cette capacité de détection et d’anticipation était l’un des obstacles rencontrés par Castagnos et al. (2010). Bien qu’ils aient proposé un modèle décrivant les différentes phases de consultations et les variations de diversité souhaitables associées dans un système de recommandation aux cours d’une session, il leur était impossible de prédire les fins de sessions. A terme, notre modèle pourrait donc être utilisé dans des cycles de recommandation, afin de répondre aux besoins en diversité des utilisateurs sans que ces derniers aient besoin de le demander explicitement et ce sans connaissance explicite du contexte courant des utilisateurs (ce qui nécessiterait de posséder des informations pouvant nuire à la vie privée des utilisateurs).

2.3 Étude de la stabilité et optimisation des paramètres

Nous avons montré qu’il est possible d’utiliser l’évolution temporelle de la diversité pour détecter les changements de contexte. La configuration utilisée attribuait le même poids à

chaque attribut dans le calcul de la RD. Or, certains attributs peuvent être plus discriminants que d'autres pour détecter un changement de contexte. Par exemple, le passage d'une musique très énergique à une musique très calme sera sans doute plus révélateur d'un changement de contexte que le passage d'une musique longue à une musique courte. Si tel est le cas, le poids de l'attribut "énergie" devra être plus important que celui de la durée. Les buts de cette partie sont donc multiples et peuvent être énumérés sous forme de questions de recherche qui sont les suivantes :

1. Peut-on privilégier la détection de certains événements en changeant les poids ?
2. L'algorithme est-il stable si on modifie le paramétrage (i.e les poids élevés sont-ils toujours les mêmes) ?
3. Y-a-t-il une configuration optimale (i.e toujours meilleur que les autres) ?

2.3.1 Utilisation d'un algorithme génétique

Pour répondre à ces questions de recherche, nous devons faire varier un grand nombre de paramètres. En considérant que les poids des attributs sont des nombres à deux décimales compris qu'entre 0 et 1 et sachant que nous disposons de 13 attributs, tester toutes les configurations possibles reviendrait à tester $1.14 \cdot 10^{26}$ configurations. Étant donné cet ordre de grandeur, il est impossible d'envisager de tester toutes les configurations possibles. Sachant que nous nous trouvons face à un problème d'exploration d'un espace de très grande taille, utiliser un algorithme génétique semble être une solution adéquate afin d'optimiser les poids des attributs.

Les algorithmes génétiques ont été mis au point dans les années 60 pour répondre à des problèmes similaires (notamment le problème du voyageur de commerce Sonam and Puneet (2014)). Ce type d'algorithmes fait partie des algorithmes évolutionnistes, ce qui signifie que la solution est trouvée via la sélection des individus qui offrent les meilleures performances, tout comme la théorie de l'évolution sur laquelle ces algorithmes sont basés.

Le principe des algorithmes génétiques consiste à initialiser aléatoirement une population où chaque individu ou chromosome représente une solution possible au problème (phase d'initialisation). Dans notre cas, un chromosome représente une combinaison des poids de nos treize attributs. Les poids des attributs correspondent aux gènes des chromosomes. Ainsi, chaque chromosome possède treize gènes. Ensuite, chaque chromosome est évalué en fonction de son score par rapport au problème (phase d'évaluation). La phase d'évaluation est réalisée à l'aide d'une fonction de *fitness* dont le but est d'attribuer à chaque chromosome un score en fonction de sa performance. Dans notre cas, la performance est la capacité à détecter certains événements. Les chromosomes les plus performants, c'est-à-dire ceux ayant le meilleur score sont utilisés pour créer la génération suivante (phase de reproduction). Lors de la phase de reproduction, deux chromosomes vont échanger une partie de leurs gènes selon un pourcentage défini par un seuil d'enjambement afin de créer un nouveau chromosome. Enfin, chaque chromosome issu de la phase de reproduction peut muter ou non selon une probabilité définie dans les paramètres du modèle. La mutation correspond au changement aléatoire d'un gène d'un chromosome et permet d'introduire un gène qui n'était pas dans la population initiale. Une nouvelle génération de chromosomes sera donc créée à l'aide de la population de départ et ces derniers seront de nouveau évalués. Le processus est ensuite répété jusqu'à convergence de la solution ou lorsque le nombre de générations maximum fixé est atteint.

La figure 9 ci-dessous illustre les différentes étapes de fonctionnement d'un algorithme génétique que nous avons décrit précédemment :

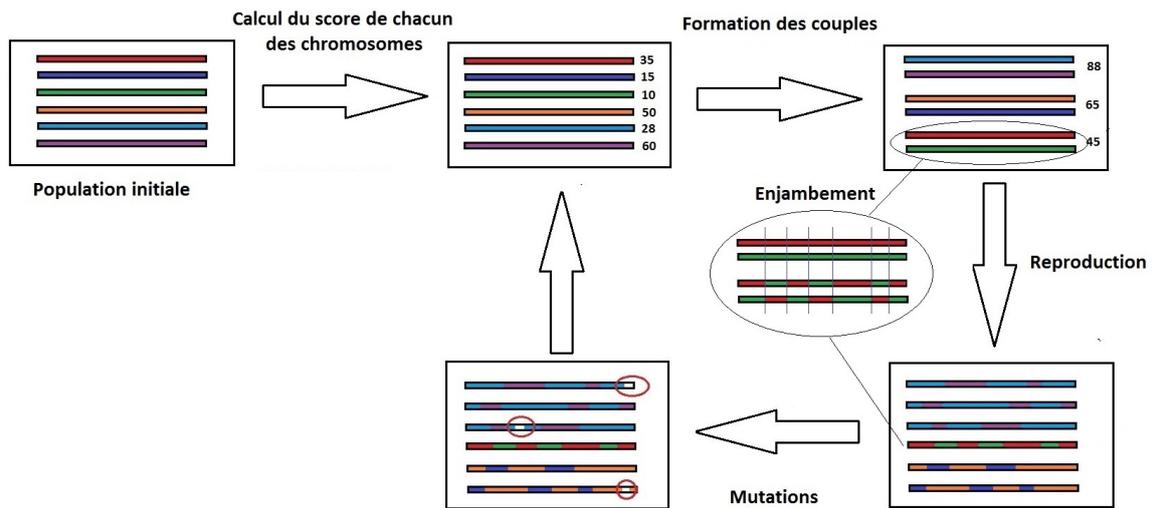


FIG. 9 – Fonctionnement d’un algorithme génétique

Bien qu’ils reposent sur des bases communes, il existe différentes variantes au sein des mécanismes que nous venons de présenter. Par exemple, certains algorithmes génétiques ne vont utiliser que les individus ayant les meilleurs scores pour créer la génération suivante, d’autres vont utiliser le principe de la roulette qui consiste en un tirage aléatoire entre tous les individus mais où les individus ayant les meilleurs scores ont plus de chance d’être sélectionnés. Cependant, ce mémoire n’étant pas destiné à présenter les différents modèles d’algorithmes génétiques, nous ne rentrerons pas dans les détails et nous nous contenterons de présenter les paramètres que nous avons utilisés afin de concevoir notre propre algorithme génétique.

2.3.2 Paramètres utilisés

Les paramètres que nous avons utilisés pour nos études sont les suivants :

- **Initialisation de la population** : la population initiale comporte 500 chromosomes dont les treize gènes ont été choisis aléatoirement entre une valeur minimale et maximale (qui correspondent aux poids des coefficients des attributs) ;
- **Évaluation des chromosomes (fonction de *fitness*)** : le score d’un chromosome est déterminé par le nombre de détections effectuées sur l’événement choisi. Par exemple, dans le cas où l’on cherche à maximiser les détections des sessions, le score du chromosome évalué correspond au nombre de changements de sessions que ce dernier aura permis de retrouver ;
- **Formation des couples** : les couples sont formés aléatoirement parmi la population de chromosomes et chaque couple possède un score qui est la somme des scores des chromosomes qui le composent (cf. figure 9). Ensuite, selon une loi de Poisson avec un coefficient de $2,12^{18}$, nous avons généré une liste du nombre de fils devant être créés par chaque couple. Plus le score du couple est important et plus le nombre de fils qu’il devra généré sera grand. Ce mécanisme permet de privilégier, au sein de la population ; les gènes qui permettent d’obtenir de bons scores sans pour autant tomber dans une forme d’eugénisme qui serait néfaste pour la convergence de l’algorithme (car l’algorithme convergerait rapidement mais sans pour autant offrir de bonnes performances).

18. Le coefficient $2,12$ a été choisi de manière à perpétuer la population sans pour autant qu’elle grandisse trop rapidement.

- Reproduction : Lors de la reproduction, les deux chromosomes vont s'échanger une partie de leurs gènes à hauteur de 70 % et les parents ne seront pas inclus dans la génération suivante. Ainsi, seuls les nouveaux chromosomes issus de la génération précédente sont conservés.
- Mutations : Chaque nouveau chromosome peut muter avec une probabilité de $\frac{1}{13}$. Si un chromosome mute, alors un de ses gènes sera de nouveau généré aléatoirement.

2.3.3 Résultats

La première hypothèse que nous avons voulu vérifier était de savoir si l'attribution de coefficients différents pouvait avoir un impact positif sur le nombre de sessions détectées (question de recherche n° 1 p.25). Pour ce faire, nous avons utilisé l'algorithme génétique que nous avons précédemment présenté. Les poids des attributs ne sont plus fixés à 1 mais sont des chromosomes dont les gènes sont compris entre 0 et 1. Les tests ont été effectués pour différents seuils de RD'. Les tableaux ci-dessous présentent les résultats qui ont été obtenus sur 50 générations de chromosomes :

	Seuil RD'	0	0,05	0,10	0,15	0,20
	Existants	Détectés	Détectés	Détectés	Détectés	Détectés
Fins de sessions	18 412	17 737	16 288	15 776	14450	12877
Contextes implicites	-	22 104	17 412	15776	14533	11075
Musiques passées	15 008	3 915	3 137	2 807	2448	2050

TAB. 4 – Résultats avec des coefficients compris entre 0 et 1

Nous pouvons constater dans un premier temps qu'un paramétrage des coefficients augmente les performances de détections de notre modèle. En effet, le nombre de sessions détectées passe de 14 253 (supra, p.27) pour des poids de 1 à 16 228 (tableau 4 ci-dessus) pour des poids compris entre 0 et 1. Cela confirme l'hypothèse selon laquelle la détection de certains événements peut être privilégié et que certains attributs sont plus déterminants que d'autres pour détecter les contextes (question de recherche n° 1, p.25).

D'autre part, nous avons choisi de modifier notre fonction de *fitness* afin de voir si il était possible d'augmenter les capacités de prédiction de notre modèle. Nous avons donc modifié la fonction de *fitness* afin d'améliorer l'anticipation des détections des sessions. Pour ce faire, la fonction *fitness* était la fonction qui maximisait le nombre de détections effectuées à $t+1$ et $t+2$. Le seuil de RD' est fixé à 0,05 et les poids sont compris entre 0 et 1. Les résultats obtenus sont présentés dans le tableau 5 ci-dessous.

Moment détection	t - 2	t - 1	t	t + 1	t + 2	total
Nombre détection	3168	4036	6333	4249	4123	21909

TAB. 5 – Maximisation de la détection sessions à $t+1$ et $t+2$

Nous pouvons constater que notre modèle peut effectivement être paramétré afin d'augmenter ses capacités à anticiper les fins de sessions, car les nombres de sessions détectées à $t+1$ et $t+2$ passent respectivement de 3 356 et 3 900 (infra, p.28) à 4249 et 4123. Ces chiffres renforcent l'hypothèse selon laquelle le changement des poids peut permettre de privilégier la détection de certains événements (question de recherche n° 1, p.25).

Nous avons également essayé de paramétré notre modèle afin de privilégier la détection des musiques ayant été passées. Cependant, comme nous l'avons précédemment expliqué les musiques

qui ont été passées sont difficilement détectables à l'aide de l'évolution de la diversité et les détections de ces dernières n'ont pas pu être améliorées.

Le tableau 6 ci-dessous, résume les positions des détections effectuées pour différents seuil de RD' pour des poids compris entre 0 et 1.

Moment détection	seuil RD'				
	0	0,05	0,1	0,15	0,20
t-2	4 448	3 506	3 021	2 591	1 921
t-1	4 397	3 461	2 911	2 810	1 978
t	9 220	8 174	7 861	6 531	6 668
t+1	3 877	3 356	2 989	3 080	2 221
t+2	4 893	3 900	3 405	3 077	2 356
total	26 835	22 047	20 187	18 089	15 153

TAB. 6 – Positions des détections

Le seuil de RD' impacte également significativement les performances de détection, comme le montrent les résultats du tableau 4 et du tableau 6, plus ce dernier est petit et plus les faibles variations de contexte sont détectées. Inversement, plus ce seuil est important, plus les changements de contexte doivent être marqués par une diversité importante. C'est pourquoi dans un cas le nombre de contextes détectés augmente et dans l'autre cas le nombre de contextes détectés diminue. A l'heure actuelle, nous ne pouvons pas encore dire qu'une valeur de ce seuil est plus adaptée qu'une autre. Parmi les perspectives, nous envisageons de tester notre modèle en situation réelle sur des recommandations et de regarder si l'on peut augmenter la satisfaction des utilisateurs en prédisant leur besoin de diversité.

Afin de vérifier la stabilité de notre modèle (question de recherche n° 2, p.25), nous avons choisi de lancer plusieurs fois notre algorithme génétique. Pour ce faire, nous avons vérifié si la configuration des poids des attributs à la fin des 50 générations n'était pas due à l'initialisation de notre population. A l'issue de chaque lancement, nous avons classé les poids des attributs par ordre croissant et nous avons récupéré la position de chaque attribut. Le but est de vérifier si les attributs qui ont les poids les plus importants sont les mêmes après chaque exécution.

Seuil	0		0,05		0,10	
	Moyenne	Écart type	Moyenne	Écart type	Moyenne	Écart type
Mode	2,95	0,39	1	0	1	0
Artistes similaires	3,95	0,22	2,35	0,49	2	0
Énergie (musique)	5,05	0,22	2,65	0,49	7,3	2,29
Années d'activités	7,25	0,44	4,75	1,74	3	0
Termes (artiste)	8,45	2,25	7,35	2,6	5,1	2,04
Familiarity (artiste)	9,75	1,57	7,95	1,67	8,4	2,8
Tempo	1,05	0,22	8,7	2,83	10,05	2,64
Danceability (musique)	10,5	1,76	8,75	3,04	9,25	2,46
Loudness	7,8	2,41	9,15	2,6	9,55	2,25
Hotttness (musique)	2,05	0,22	9,20	2,74	8	2,99
Coordonnées	10,75	1,94	9,30	2,90	7,85	2,78
Durée	10,2	2,50	9,7	2,2	10,5	2,63
Hotttness (artiste)	11,25	1,41	10,15	2,6	8	2,99

TAB. 7 – Rang des poids des attributs

Le tableau 7 ci-dessus nous renseigne sur les statistiques des positions des treize attributs sur 20 initialisations. Les poids sont compris entre 0 et 1 et le tableau présente les moyennes et les écarts types des rangs des attributs sur 20 exécutions.

Il convient de constater que quel que soit le seuil de RD' utilisé, les attributs "Mode" et "Artistes similaires" sont les attributs ayant les poids les plus importants. Inversement, la "popularité" de l'artiste ainsi que ses coordonnées sont les attributs les moins discriminants dans la détection d'événements. Les attributs avec des poids élevés sont souvent les mêmes malgré de petites variations dans le classement de tête, d'où un écart type faible (cf. Tab.7). Par ailleurs, les attributs avec des poids faibles sont également souvent les mêmes. Cela tend à conforter l'hypothèse selon laquelle le modèle DANCE est stable (question de recherche n° 2, p.25) et qu'il existe une configuration optimale consistant à maximiser certains poids.

2.3.4 Discussion

La conception d'un modèle à base d'un algorithme génétique nous a permis de pouvoir tester de nombreuses combinaisons de poids des attributs. Dans un premier temps, nous avons pu confirmer l'hypothèse de recherche n° 1 car les performances de détections obtenues ont été améliorées. Pour un même seuil de RD' (0,05) 2 000 sessions supplémentaires ont pu être ainsi détectées. De plus, dans un second temps, nous avons également constaté qu'il est possible d'améliorer les capacités de prédictions de notre modèle.

Nous avons également démontré que l'hypothèse de recherche n° 2 est valide puisqu'il existe une stabilité dans les configurations des poids des attributs et ce, même après plusieurs exécutions de notre algorithme. Au fil de l'étude sur la stabilité de notre modèle, nous avons également mis en évidence le fait qu'il existe une configuration optimale de certains poids tels que le 'mode' et les 'artistes similaires'. Ce dernier constat nous a donc permis de valider notre hypothèse de recherche n° 3.

3 Conclusion et perspectives

3.1 Conclusion

Alors que l'impact positif de la diversité dans les systèmes de recommandation n'est plus à démontrer, de nombreuses problématiques se posent autour de cette dimension. Parmi les problématiques dominantes, la prise en compte du besoin en diversité chez les utilisateurs a récemment été étudiée par Castagnos et al. (2010). Ces derniers ont démontré que le besoin en diversité chez les utilisateurs n'est pas le même au cours du temps. Cependant, bien qu'ils aient proposé un modèle indiquant dans quel contexte la diversité des items recommandés devait être maximisée ou minimisée, détecter et caractériser ces contextes constituait un obstacle majeur. Dans ce mémoire, nous avons présenté le modèle DANCE et ce dernier permettra à terme d'outrepasser cet obstacle. Comme nous l'avons montré, le modèle DANCE permet d'une part, de modéliser l'évolution de la diversité au cours du temps avec une complexité algorithmique moindre et d'autre part, d'être utilisé afin de détecter les changements de contexte des utilisateurs. La force de notre modèle réside dans sa capacité à pouvoir détecter les similarités et les dissimilarités qui existent entre les différents contextes. En plus d'offrir de bonnes performances, notre modèle possède l'avantage d'être respectueux de la vie privée des utilisateurs. En effet, seules une connaissance des items et de l'heure à laquelle ces derniers ont été consultés suffisent à notre modèle, sur un historique de petite taille.

Dans ce mémoire, nous avons donc proposé un modèle formel, innovant et générique pour comprendre le contexte de l'utilisateur en observant le niveau de diversité au sein de la séquence de consultation. Il est considéré comme générique car applicable à tous les domaines, dès lors qu'il est possible de caractériser les items sous forme d'attributs. Nous avons éprouvé ce modèle sur la base d'un corpus musical de grande taille. En outre, nous avons proposé un algorithme génétique permettant à la fois d'optimiser les performances de notre modèle, et de vérifier sa stabilité. Nous avons également démontré la capacité de notre modèle à prédire des événements tels que les fins de sessions. Enfin, nous avons proposé deux nouveaux concepts qui sont le contexte explicite et implicite, en nous inspirant des définitions de l'état de l'art autour des préférences implicites et explicites Castagnos (2008), Jones (2010).

3.2 Perspectives

Le modèle DANCE est le premier modèle d'après la littérature à proposer une modélisation temporelle de la diversité. Il en résulte donc que les perspectives qui s'offrent sont nombreuses. Premièrement, nous souhaiterions pouvoir valider notre modèle à l'aide d'utilisateurs réels. En effet, le corpus que nous avons utilisé est incomplet et utilisé hors ligne. Nous ne savons par exemple pas comment les enchaînements de musiques ont été réalisés (à la main, liste de lecture personnelle ou communautaire, recommandation). Nous souhaiterions donc pouvoir maîtriser davantage les paramètres afin d'augmenter la précision de notre modèle. Dans un second temps, notre modèle pourrait être utilisé à des fins de recommandation afin de fournir des recommandations adaptées au contexte de l'utilisateur et à mesurer leur satisfaction.

Comme nous l'avons précisé, le modèle peut être utilisé dans d'autres domaines comme les réseaux sociaux professionnels tels que Yupeek¹⁹ ou LinkedIn²⁰ qui sont des domaines où la recommandation occupe une place dominante. Une autre perspective consisterait à augmenter les capacités de modélisation du comportement des utilisateurs. Une amélioration du modèle

19. <http://yupeek.com/>

20. <https://www.linkedin.com/home>

DANCE consisterait ainsi à caractériser les contextes des utilisateurs en fonction des caractéristiques des items contenus dans ces derniers. Nous envisageons également de nous servir de notre modèle afin de détecter certains *patterns* du comportement des utilisateurs.

Bilan personnel

Ce stage constitue à l'heure actuelle l'expérience professionnelle la plus enrichissante dont j'ai pu bénéficier. J'ai ainsi pu prendre le temps de découvrir les problématiques liées à ma thématique de recherche et de proposer une approche nouvelle permettant de modéliser l'évolution de la diversité au cours du temps. Durant ce stage, j'ai eu la chance de découvrir plusieurs facettes du métier de chercheur, incluant la synthèse de l'état de l'art, l'identification des verrous scientifiques, la définition d'hypothèses de recherche, la proposition d'un modèle, son implémentation dans un contexte d'optimisation large échelle, sa validation et la publication des résultats dans une conférences ACM²¹ internationale L'Huillier et al. (2014). Fort de ces expériences et des compétences que j'ai pu acquérir ces derniers mois, je souhaiterais poursuivre mes travaux en thèse.

21. *Association for Computing Machinery*

Bibliographie

Références

- Adomavicius, G., Sankaranarayanan, S., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender system using a multidimensional approach. *Transaction on Information System*, 23 :103–145.
- Adomavicius, G. and Tuzhilin, A. (2001). Contexte-aware recommender systems. *Recommender Systems Handbook*, pages 217–253.
- Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. (2009). Diversifying search results. In *Proceedings of the second ACM International Conference on Web Search and Data Mining, WSDM'09*, Barcelona, Spain.
- Bazire, M. and Brezillon, P. (2005). Understanding context before using it. In *Proceedings of the 5th International Conference on Modeling and Using Context, ICMUC'05*. Springer-Verlag.
- Boim, R., Milo, T., and Novgorodov, S. (2011). Diversification and refinement in collaborative filtering recommender. In *Proceedings of the 20 ACM international conference on Information and knowledge management, CIKM'11*, Glasgow, UK.
- Bonnin, G. (2010). *Vers des systèmes de recommandation robustes pour la navigation Web : inspiration de la modélisation statistique du langage*. These, Université Nancy II.
- Bradley, K. and Smith, B. (2001). Improving recommendation diversity. In *Irish Conference on Artificial Intelligence and Cognitive Science, AICS'01*, pages 85–94, San Francisco, CA, USA.
- Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98*, pages 43–52, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Brown, P. and Chen, X. (1997). Context-aware applications : from the laboratory to the marketplace. In *IEEE, IEEE'97*, pages 58–64.
- Burke, R. (2002). Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370.
- Castagnos, S. (2008). *Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information*. These, Université Nancy II.
- Castagnos, S., Brun, A., and Boyer, A. (2013). Utilité et perception de la diversité dans les systèmes de recommandation. In *CORIA 2013 - 10ème Conférence en Recherche d'Information et Applications*, Neuchâtel, Suisse.
- Castagnos, S., Brun, A., and Boyer, A. (2014). La diversité : entre besoin et méfiance dans les systèmes de recommandation. In *Information Interaction Intelligence, I3 '14*.
- Castagnos, S., Jones, N., and Pu, P. (2010). Eye-tracking product recommenders' usage. In *RecSys*, pages 29–36.
- Celma, Òscar. (2008). *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona.
- Chee, S. H. S., Han, J., and Wang, K. (2001). Rectree : An efficient collaborative filtering method. In *Proceedings of the Third International Conference on Data Warehousing and Knowledge Discovery, DaWaK '01*, pages 141–151, London, UK, UK. Springer-Verlag.
- Chen, G. and Chen, L. (2014). Recommendation based on contextual opinions. In Dimitrova, V., Kuflik, T., Chin, D., Ricci, F., Dolog, P., and Houben, G.-J., editors, *User Modeling, Adaptation, and Personalization*, volume 8538 of *UMAP '14*, pages 61–73. Springer International Publishing.

- Chen, L., de Gemmis, M., Felfernig, A., Lops, P., Ricci, F., and Semeraro, G. (2013). Human decision making and recommender systems. *ACM Trans. Interact. Intell. Syst.*, 3(3) :17 :1–17 :7.
- Chen, L. and Pu, P. (2012). Critiquing-based recommenders : survey and emerging trends. *User Model. User-Adapt. Interact.*, 22(1-2) :125–150.
- Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S., and MacKinnon, I. (2008). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 659–666, New York, NY, USA. ACM.
- Cosley, D., Lam, S. K., Albert, I., Konstan, J. A., and Riedl, J. (2003). Is seeing believing ? : How recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 585–592, New York, NY, USA. ACM.
- Davis, F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. In *MIS Quarterly* 13, ICCBR'01, pages 319–340.
- Department, D. and Bridge, D. (2001). Product recommendation systems : A new direction. In *Workshop on CBR in Electronic Commerce at The international Conference on Case-Based Reasoning*, ICCBR'01, pages 79–86.
- Dey, A., Abowd, G., and Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototype of context-aware applications. In *Human-Computer Interaction*, HCT'01, pages 97–166.
- Foulonneau, M., Groues, V., Naudet, Y., and Chevalier, M. (2014). Recommandeurs et diversité : Exploitation de la longue traîne et diversité des listes de recommandations. In *Les systèmes de recommandation*. Hermès, <http://www.editions-hermes.fr/>.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12) :61–70.
- Häubl, G. and Murray, K. B. (2003). Preference construction and persistence in digital marketplaces : The role of electronic recommendation agents. *Journal of Consumer Psychology*, 13(1) :75–91.
- Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA. ACM.
- Jahrer, M., Tscher, A., and Legenstein, R. (2010). Combining predictions for accurate recommender systems. In *ACM International Conference on Knowledge Discovery and Data Mining*, SIGKDD '10, pages 693–702, Washington, DC, USA.
- Jones, N. (2010). *User Perceived Qualities and Acceptance of Recommender Systems : The Role of Diversity*. These, Ecole polytechnique de Lausanne.
- Jones, N., Brun, A., Boyer, A., and Hamad, A. (2011). An exploratory work in using comparisons instead of ratings. In Huemer, C. and Setzer, T., editors, *EC-Web*, volume 85 of *Lecture Notes in Business Information Processing*, pages 184–195. Springer.
- Jones, N. and Pu, P. (2007). User technology adoption issues in recommender systems. In *Proceedings of the 2007 Networking and Electronic Commerce Research Conference*, page 379–394, Riva del Garda, Italy.
- Lathia, N. (2010). *Evaluating Collaborative Filtering Over Time*. PhD thesis, University of London, Department of Computer Science, University College London.
- Lathia, N., Hailes, S., Capra, L., and Amatriain, X. (2010). Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 210–217, New York, NY, USA. ACM.
- Leake, D. and Scherle, R. (2001). Towards context-based search engine selection. In *Proceedings of the 6th International conference on Intelligent user interfaces*, pages 109–112, New York, NY, USA. ACM.

- Lee, S., Yang, J., and Park, S.-Y. (2004). Discovery of hidden similarity on collaborative filtering to overcome sparsity problem. In *Discovery Science*, pages 396–402.
- L’Huillier, A., Castagnos, S., and Boyer, A. (2014). Understanding usages by modeling diversity over time. In *UMAP*.
- McGinty, L. and Smyth, B. (2003). On the role of diversity in conversational recommender systems. In *Proceedings of the Fifth International Conference on Case-Based Reasoning*, pages 276–290. Springer.
- McNee, S. M., Riedl, J., and Konstan, J. A. (2006). Being accurate is not enough : how accuracy metrics have hurt recommender systems. In *CHI ’06 : CHI ’06 extended abstracts on Human factors in computing systems*, pages 1097–1101. ACM.
- McSherry, D. (2002). Diversity-conscious retrieval. In *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, pages 219–233, London, UK.
- Murakami, T., Mori, K., and Orihara, R. (2008). Metrics for evaluating the serendipity of recommendation lists. In *Proceedings of the 2007 Conference on New Frontiers in Artificial Intelligence, JSAI’07*, pages 40–46, Berlin, Heidelberg. Springer-Verlag.
- Oku, K., Nakajima, J., and Uemura, S. (2006). Context-aware svm for context-dependent information recommendation. In *Proceedings of the 7th International Conference on Mobile Data Management*, page 109.
- Onuma, Kensuke, H. T. and Faloutsos, C. (2009). Tangent : a novel, “surprise me”, recommendation algorithm. page 657–666.
- Palmisano, C., Tuzhilin, A., and Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE Transactions on Knowledge and Data Engineering*, 20(11) :1535–1549.
- Park, Y. J. and Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18.
- Pazzani, M. J. and Billsus, D. (2007). The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg.
- Pu, P., Zhou, M., and Castagnos, S. (2009). Critiquing recommenders for public taste products. In *RecSys*, pages 249–252.
- Radlinski, F., Bennett, P. N., Carterette, B., and Joachims, T. (2009). Redundancy, diversity and interdependent document relevance. *SIGIR Forum*, 43(2) :46–52.
- Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens : An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW ’94*, pages 175–186, New York, NY, USA. ACM.
- Said, A., Kille, B., Brijnesh, J., and Albayrak, S. (2012). Increasing diversity through furthest neighbor-based recommendation. In *Proceedings of the Workshop on Diversity in Document Retrieval, WSDM’12*, Seattle, USA.
- Schafer, J., Konstan, J., and Riedl, J. (2002). Meta-recommendations systems : user-controlled integration of diverse recommendations. In *International Conference on Information And Knowledge Management*, pages 43–51.
- Schilit, B. and Theimer, M. (1994). Disseminating active map information to mobile hosts. In *IEEE network*, pages 22–32. ACM Press.
- Shardanand, U. and Maes, P. (1995). Social information filtering : Algorithms for automating “word of mouth”. pages 210–217. ACM Press.

- Shimazu, H. (2001). Expertclerk : Navigation shoppers buying process with the combination of asking and proposing. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI'01*, pages 1443–1450, Seattle, USA. ACM Press.
- Sieg, A., Mabasher, B., and Burke, R. (2007). Representing context in web search with ontological user profiles. In *Proceedings of the 6th International Conference on Modeling and Using Context*.
- Smyth, B. and McClave, P. (2001). Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning : Case-Based Reasoning Research and Development, ICCBR '01*, pages 347–361, London, UK, UK. Springer-Verlag.
- Sonam, K. and Puneet, G. (2014). An efficient solution of travelling salesman problem using genetic algorithm. In *International Journal of Advanced Research in Computer Science and Software Engineering - Volume 4, Issue 5*.
- Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009 :4 :2–4 :2.
- thenextweb.com (2012). Remember netflix’s \$1m algorithm contest ? well, here’s why it didn’t use the winning entry @ONLINE.
- Ungar, L. and Foster, D. (1998). Clustering methods for collaborative filtering. In *Proceedings of the Workshop on Recommendation Systems*. AAAI Press, Menlo Park California.
- Webster, N. (1980). Webster’s new twentieth century dictionary of the english language.
- Zhang, M. and Hurley, N. (2008). Avoiding monotony : Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys '08*, pages 123–130, New York, NY, USA. ACM.
- Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, pages 22–32, New York, NY, USA. ACM.

Annexes

A Définition des attributs Echonest

Les définitions des attributs telles qu’elles sont données par le Echonest sont les suivantes :

– **attributs pour les musiques :**

- *Durée* : indique la durée de la musique exprimée en secondes.
- *Mode* : indique la modalité (mineure ou majeure) d’une musique.
- *Energy* : indice caractérisant l’humeur dégagée par la musique, est-ce que cette dernière donne envie de “bouger” ou de rester calme ? Plus l’indice est grand et plus la musique est “énergique”.
- *Tempo* : mesure le tempo d’une musique (c’est à dire le nombre de battements par minute). Plus le tempo est élevé, plus le rythme de la musique est rapide.
- *Danceability* : indice caractérisant la facilité avec laquelle une personne peut danser sur une musique (par exemple, il est plus facile de danser sur une musique rock que sur un adagio). Plus l’indice est grand et plus la musique se prête à danser.
- *Loudness* : mesure l’intensité moyenne d’une musique en décibels (dB).
- *Hotttness* : indice caractérisant le “buzz” actuel d’une musique. Plus cet indice est élevé est plus la musique est populaire.

– **attributs pour les artistes :**

- *Hotttness* : indice caractérisant le “buzz” actuel d’un artiste. Plus cet indice est élevé est plus l’artiste est populaire.
- *Artistes similaires* : liste contenant les dix artistes les plus similaires par rapport à un artiste cible.
- *Termes* : listes de termes musicaux permettant de caractériser l’artiste. Chaque terme possède un poids et une fréquence. La fréquence d’un terme est en rapport avec le nombre de fois où ce terme est associé à un artiste, alors que le poids représente la capacité qu’a un terme à caractériser une musique par rapport à une ontologie et la mesure de TF-IDS²².
- *Familiarity* : indice caractérisant la popularité générale (indépendamment du temps) d’un artiste. Plus l’indice est grand, plus cet artiste est connu.
- *Coordonnées* : contient la latitude et la longitude de la ville de l’artiste.
- *Années d’activités* : date de début et de fin d’activité pour un artiste.

22. <http://fr.wikipedia.org/wiki/TF-IDF> : Le TF-IDF (de l’anglais Term Frequency-Inverse Document Frequency) est une méthode de pondération souvent utilisée en recherche d’information. Cette mesure statistique permet d’évaluer l’importance d’un terme contenu dans un document, relativement à une collection ou un corpus.