



**HAL**  
open science

# Modéliser la diversité au cours du temps pour détecter le contexte dans un service de musique en ligne

Amaury L'Huillier, Sylvain Castagnos, Anne Boyer

## ► To cite this version:

Amaury L'Huillier, Sylvain Castagnos, Anne Boyer. Modéliser la diversité au cours du temps pour détecter le contexte dans un service de musique en ligne. *Revue des Sciences et Technologies de l'Information - Série TSI: Technique et Science Informatiques*, 2016. hal-01300419

**HAL Id: hal-01300419**

**<https://inria.hal.science/hal-01300419>**

Submitted on 2 May 2016

**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.

---

# Modéliser la diversité au cours du temps pour détecter le contexte dans un service de musique en ligne

Amaury L’Huillier, Sylvain Castagnos, Anne Boyer

LORIA – Campus scientifique B.P. 239

54506 Vandoeuvre-lès-Nancy CEDEX

{amaury.lhuillier,sylvain.castagnos,anne.boyer}@loria.fr

---

*RÉSUMÉ. De nombreuses études ont démontré que la prise en compte du contexte améliore la qualité des systèmes de recommandation. Cependant, les méthodes traditionnelles permettent d’inférer le contexte à l’aide de données personnelles (localisation, date, âge, etc.). Dans ce papier, nous proposons de détecter automatiquement les changements de contexte, sans connaissance sur les utilisateurs (contexte explicite), mais en fonction des caractéristiques communes aux items consultés (contexte implicite). Pour ce faire, nous proposons un modèle formel capable d’établir une correspondance entre les variations de diversité au cours du temps dans les parcours des utilisateurs et les changements de contexte. Ce modèle a été testé sur un corpus musical de plus de 200 000 écoutes. Pour valider la pertinence de notre modèle, nous avons cherché à retrouver des événements à partir des changements de contexte détectés : notre modèle a ainsi permis de retrouver 88% des fins de session.*

*ABSTRACT. Many studies have shown that taking into account the context improves the quality of recommender systems. However, traditional methods infer the context using personal data (location, date, age, etc.). In this paper, we propose to automatically detect context changes, without knowledge on users (explicit context), but based on common features of consulted items (implicit context). To do this, we propose a formal model which can establish a correspondence between the variation of diversity over time within the paths of users and context changes. This model has been tested on a musical corpus of more than 200,000 tracks. To validate the relevance of our model, we sought to retrieve events from the detected changes of context: our model has recovered 88% of session ends.*

*MOTS-CLÉS : diversité, contexte, recommandation, facteurs humains, modélisation utilisateur*

*KEYWORDS: diversity, context, recommendation, human factors, user modeling*

---

DOI:10.3166/HSP.x.1-36 © 2016 Lavoisier

## Introduction

Sur Internet, la quantité d'information à laquelle les utilisateurs ont accès ne cesse d'augmenter, il est donc 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 aux utilisateurs (généralement sous forme de liste) 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'Intelligence Artificielle, les systèmes de recommandation ont permis de fournir des recommandations de plus en plus précises, c'est-à-dire proches des préférences réelles de l'utilisateur. 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 dans le but de maximiser la satisfactions des utilisateurs (Jones, 2010). À titre d'exemple, 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. Comprendre les facteurs humains inhérents à la prise de décision permet donc d'adapter la stratégie de recommandation afin d'augmenter le taux d'acceptation et d'adoption du système par les utilisateurs.

Parmi ces facteurs humains (confiance, prise en compte du contexte, précision...), cet article se concentre sur le besoin en diversité ayant récemment été mis en avant comme étant une dimension importante dans les systèmes de recommandation. La diversité (en temps que métrique) est la mesure opposée à la similarité : plus des items sont similaires entre eux, moins ils sont divers entre eux, et réciproquement. 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, McClave, 2001). Plus 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, il n'est pas possible de détecter ou d'anticiper ces phases ou ces fins de session. Or, (Lathia *et al.*, 2010), (Castagnos *et al.*, 2013) ont observé que la diversité pouvait provoquer la méfiance des utilisateurs si cette dernière n'est pas proposée au bon moment. Il existe donc plusieurs phases (contextes) au sein d'une session et il est nécessaire d'adapter le niveau de diversité dans les recommandations en fonctions des différentes phases dans lesquelles se trouvent les utilisateurs. Le défi actuel consiste alors à identifier automatiquement les différents contextes jalonnant les séquences de consultations afin de pouvoir apporter le bon niveau de diversité en fonction de ce contexte.

En résumé, connaître le contexte courant de l'utilisateur permet d'estimer le niveau de diversité souhaitable. Ce papier prolonge ce raisonnement, en se demandant s'il n'existerait pas une réciprocity entre ces deux facteurs que sont le contexte et le

besoin en diversité. Notre problématique consiste à analyser l'évolution de la diversité dans les parcours des utilisateurs, afin de détecter automatiquement d'éventuels changements de contexte.

L'article est organisé en trois parties que nous allons présenter. Dans la première partie, un état de l'art est établi afin de rappeler les principales étapes d'un système de recommandation, et de présenter les travaux sur la diversité et le contexte. Le vocabulaire et la syntaxe que nous utiliserons dans cet article sont également explicités dans cette partie. 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 obtenu avec ce modèle afin de détecter/prédire des événements initiés par les utilisateurs. Notre étude repose sur un corpus de consultations musicales issues de l'application de *scrobbling*<sup>1</sup> du site Lastfm.fr<sup>2</sup>.

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. Du fait de la faible complexité algorithmique de notre modèle, cette première étape constituera une preuve de concept sur un corpus de grande taille. Par la suite, nous chercherons à trouver une correspondance entre les changements de contexte détectés par notre modèle et des événements particuliers tels que les fins de session. L'objectif consiste donc à expliquer les changements de contexte au moyen de l'évolution temporelle de la diversité. 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 un système de recommandation.

---

1. Le *scrobbling* consiste à récupérer les informations concernant les habitudes d'écoute des utilisateurs à l'aide d'un *plugin* installé sur leurs ordinateurs personnels, téléphones et tablettes.

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

## 1. État de l'art

De plus en plus présents sur les sites internet, les services de recherche et d'accès à l'information, et plus récemment dans les objets connectés, les systèmes de recommandation sont devenus des outils indispensables pour les utilisateurs. Leur but est simple : analyser et filtrer l'information disponible afin de proposer aux utilisateurs des items étant susceptibles de les intéresser ou de répondre à leurs besoins. Avec la quantité croissante et incommensurable de données disponibles de nos jours, offrir un service personnalisé et adapté à chaque utilisateur est un véritable enjeu. Afin de positionner notre travail parmi les travaux existants, nous avons organisé notre état de l'art comme suit : après avoir brièvement rappelé les grandes étapes du processus de recommandation, nous présentons les difficultés actuellement rencontrées dues à la non prise en compte des facteurs humains inhérents à la prise de décision dans les systèmes de recommandation. Par la suite, nous présentons un facteur humain en particulier, à savoir le besoin en diversité, ainsi que les raisons pour lesquelles ce dernier joue un rôle majeur dans de tels systèmes. Nous présentons ensuite la notion de contexte afin d'expliquer son rôle et son intégration dans les systèmes de recommandation. À l'issue de cet état de l'art, nous proposons notre propre définition de la notion de contexte sur laquelle se basent les travaux que nous présentons dans la deuxième partie de cet article.

### 1.1. Les systèmes de recommandation

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

- collecter les traces d'usage des utilisateurs ;
- transformer et synthétiser ces traces sous forme d'information de plus haut niveau (profils utilisateurs) ;
- utiliser les profils utilisateurs afin de trouver les recommandations adaptées à chaque profil à l'aide de méthodes d'Apprentissage Automatique ;
- présenter les recommandations à l'utilisateur de manière appropriée.

Les traces récupérées suite aux interactions entre l'utilisateur et le système sont des données dites brutes, c'est-à-dire qu'elles ne peuvent pas être exploitées directement par les algorithmes de recommandation. Il existe deux types de traces : les traces explicites et les traces implicites (Castagnos, 2008), (Jones, 2010). Les premières correspondent aux données fournies explicitement par les utilisateurs, comme les notes attribuées à un item (exprimées par un chiffre ou via une échelle de notation utilisant des étoiles par exemple), les votes attribués à un item (j'aime un peu, j'aime beaucoup, je n'aime pas...) ou encore les critiques fournies par les utilisateurs à propos d'un item. Les traces implicites sont, quant à elles, des informations relatives aux comportements des utilisateurs, comme le temps passé sur une page web ou le nombre de clics effectués. (Jawaheer *et al.*, 2014) ont récemment présenté un état de

l'art concernant les avantages et les inconvénients des traces implicites et explicites et mettent en avant les bénéfices des systèmes hybrides permettant d'utiliser conjointement ces deux types de traces.

Les traces sont ensuite transformées dans le but d'estimer les préférences des utilisateurs par rapport aux items qu'ils ont consultés. Les traces sous forme de notes sont des indicateurs directs de la préférence : plus la note est importante, plus l'intérêt est grand. À l'inverse les traces implicites nécessitent une étape de transformation supplémentaire afin de pouvoir inférer les préférences des utilisateurs sous la forme de notes. Les notes obtenues sont ensuite réunies dans une matrice utilisateurs-items également appelée modèle de préférences contenant l'ensemble des profils utilisateurs. Les traces implicites et explicites peuvent être réunies dans un seul modèle ou constituer deux modèles différents.

Les profils utilisateurs ainsi établis sont utilisés afin de calculer les recommandations. Pour ce faire, il existe deux grandes familles d'algorithmes, à savoir le filtrage par contenu et le filtrage collaboratif. Le filtrage par contenu repose sur l'exploitation des informations des items consultés, afin de recommander des items similaires à ceux ayant été appréciés par un utilisateur (Kurki *et al.*, 1999). Le filtrage collaboratif se base quant à lui sur l'exploitation de la communauté d'utilisateurs d'un système, afin de proposer à un utilisateur des items ayant été appréciés par des personnes lui étant similaires (Goldberg *et al.*, 1992). Il existe d'autres méthodes de filtrage tels que les systèmes de recommandation basés sur la confiance (Abdul-Rahman, Hailes, 2000), ou sur les ontologies (Middleton *et al.*, 2001). Afin d'augmenter la précision des recommandations, certains systèmes combinent plusieurs approches de filtrage (Burke, 2002).

Une fois les recommandations déterminées par le système, celles-ci doivent être présentées à l'utilisateur. La plupart du temps, les recommandations sont présentées sous la forme d'une liste verticale ou horizontale positionnée aux abords de l'item en cours de consultation. Comme le montrent certaines études utilisateurs, une présentation adaptée des recommandations permet d'augmenter significativement le taux d'acceptation des recommandations (Jones, 2010). La présentation des recommandations ne concerne pas uniquement l'aspect esthétique, mais également l'organisation et l'explication des recommandations.

**Discussion :** Bien qu'elles soient essentielles au fonctionnement d'un système de recommandation, les étapes de fonctionnement que nous venons de présenter ne peuvent suffire à fournir des recommandations personnalisées. Un système de recommandation doit être en mesure de s'adapter dynamiquement aux besoins des utilisateurs. Afin d'y parvenir, les systèmes de recommandation doivent prendre en compte certaines variables telles que le contexte et les facteurs humains (Jones, 2010). Nous allons voir dans un premier temps quelles sont les limites rencontrées par les systèmes actuels, puis nous présenterons deux approches essentielles à la personnalisation : la prise en compte du contexte et du besoin en diversité des utilisateurs.

## 1.2. Limites des systèmes de recommandation et facteurs humains

Les systèmes de recommandation à base de filtrage par contenu, collaboratif ou hybride possèdent des avantages et des inconvénients étant propres à leurs caractéristiques ou à leur architecture (Su, Khoshgoftaar, 2009), (Lops *et al.*, 2011). Cependant, quels que soient leurs avantages et inconvénients, tous les systèmes de recommandation ont partagé depuis leur création une caractéristique commune : ils ont tous été construits et évalués autour de la métrique de précision. Les recherches dans le domaine des systèmes de recommandation se sont longtemps attachées à minimiser la valeur de l'erreur quadratique moyenne (RMSE) afin de fournir des recommandations précises. Cependant, avoir une grande précision ne suffit pas à satisfaire les utilisateurs. À titre d'exemple, les algorithmes les plus précis ayant remporté le concours Netflix<sup>3</sup> n'ont pas été 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 suffisant pour expliquer la satisfaction des utilisateurs et que des recommandations moins précises ne sont pas nécessairement 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) montrent 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. Les travaux de recherche se sont alors focalisés sur l'identification de ces facteurs permettant à la fois de maximiser l'acceptation et l'adoption des systèmes par les utilisateurs, et donc indirectement leur satisfaction.

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. Par exemple, l'adaptation du modèle *Technology Acceptance Model* (TAM) (Davis, 1989) par (Jones, 2010) atteste de cette volonté d'intégrer les facteurs humains dans les systèmes de recommandation. Le modèle originel s'intéressant à la manière dont les utilisateurs perçoivent un système informatique a été adapté au domaine de la recommandation (Figure 1). Cette adaptation du modèle TAM s'intéresse à deux points en particulier pouvant être traduit 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*)

Outre la phase de recommandation, 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

---

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

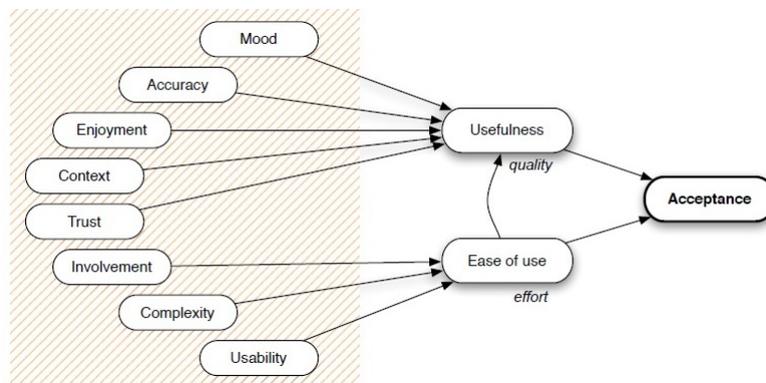


Figure 1. Technology Acceptance Model (Jones, 2010)

jusqu'à 40 % de variation 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 reposant sur la comparaison (je préfère l'item A à l'item B) et obtient une meilleure stabilité dans les préférences des utilisateurs. Par ailleurs, (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 mauvaise qualité (recommandations sclérosées).

Parmi les autres facteurs humains récemment étudiés, les deux principaux sont la diversité et le contexte. Étant donné que ces facteurs sont au cœur de nos travaux, ils sont détaillés dans les sous-sections suivantes.

### 1.3. La diversité

#### 1.3.1. Définition et rôle

La diversité, telle qu'elle a été définie pour la première fois dans le domaine des systèmes de recommandation par (Smyth, McClave, 2001) est la mesure opposée à la similarité. Plus des items sont similaires entre eux, moins ils sont divers et réciproquement. La diversité a été utilisée dans un premier temps afin de proposer des alternatives aux utilisateurs dans les recommandations proposées. C'est pourquoi l'algorithme mis au point par (Smyth, 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 a permis d'illustrer l'importance de la diversité, et certains auteurs tels que (Zhang, 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 efficace de réagir face à une mauvaise recommandation (McGinty, Smyth, 2003). En effet, si un système recommande un item n'étant pas apprécié, le système a tout intérêt à réagir de manière à proposer un item avec des caractéristiques diverses pour éviter que la pro-

chaîne recommandation ne soit trop similaire à la première (auquel cas, la nouvelle recommandation risquerait de nouveau de ne pas satisfaire l'utilisateur). Au cours d'une étude utilisateur, ils démontrèrent également tout comme (Lathia *et al.*, 2010), que la diversité ne doit pas être présente 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, 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 2 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.

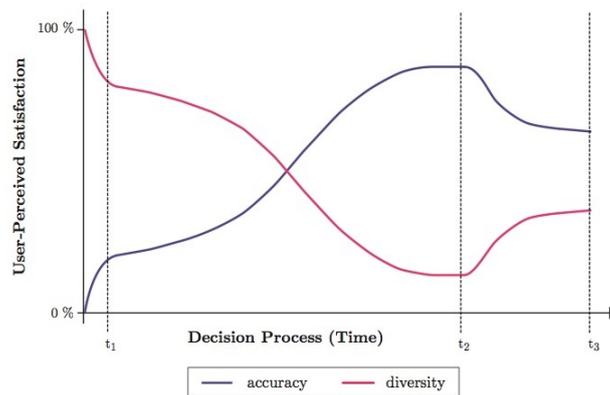


Figure 2. Modèle temporel du besoin en diversité (Castagnos *et al.*, 2010)

La dimension temporelle peut être étudiée de plusieurs manières, sur le court terme et sur le long terme. (Lathia *et al.*, 2010) se sont intéressés à la prise en compte de la diversité dans le temps à l'échelle d'un profil utilisateur, minimisant la similarité des items recommandés pour les profils les plus complets afin d'éviter de scléroser les recommandations. À l'inverse, 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 tenons à souligner qu'il faut distinguer la diversité en tant que métrique et le besoin en diversité en tant que facteur humain. Le besoin en diversité peut être extrapolé à partir de l'étude de l'évolution de la diversité. La Figure 2 décrit l'évolution du besoin en 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 large 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, c'est-à-dire proche de ses attentes et préférences. Enfin, le besoin en diversité augmente à la fin de la session d'achat, pour permettre à l'utilisateur de conforter son choix final en le distinguant des autres alternatives valables.

Poursuivant leurs travaux sur le rôle et l'impact de la diversité, (Castagnos *et al.*, 2013) démontrent que cette dimension est effectivement perçue par les utilisateurs dans les systèmes de recommandation. Leur étude révèle en outre que la diversité, bien qu'elle semble améliorer la satisfaction des utilisateurs et permette de résoudre certains problèmes dus au démarrage à froid (*cold-start*), peut également être source de méfiance ou d'incompréhension si elle est utilisée à mauvais escient. Ces conclusions corroborent les travaux de (Jones, 2010) selon lesquels le mode de présentation ainsi que la diversité des recommandations influent sur la confiance accordée au système par les utilisateurs.

Ainsi, bien que la diversité dans les recommandations corresponde à un besoin réel, la quantité de diversité à introduire, le mode de présentation et le moment où la diversité doit être maximisée ou minimisée posent de nombreuses questions s'inscrivant actuellement au cœur des recherches (Hasan *et al.*, 2014), (Liang *et al.*, 2014).

### 1.3.2. 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.

Dans certains cas, la diversité n'est pas introduite volontairement mais est tout de même présente dans les recommandations via la sérendipité<sup>4</sup> (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. Dans d'autre cas, il s'agit de contrôler précisément le niveau de diversité apporté par un système de recommandation. 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, Smith, 2001), (Department, Bridge, 2001), (Shimazu, 2001), (McSherry, 2002). Comme nous l'avons précisé précédemment, la diversité telle qu'elle a été proposée par (Smyth, McClave, 2001) est une mesure opposée à la similarité. Calculer la similarité entre deux items implique de posséder des informations sur ces derniers. Ces informations sont appelées des attributs. Par exemple, dans le cas où les items sont des films, le genre (action, aventure, comédie, drame...), la durée, l'année de sortie, le nombre d'entrées au cinéma sont des attributs. La similarité entre deux items  $i_1$  et  $i_2$

---

4. La sérendipité est la découverte par le hasard.

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 :

$$Similarité_A(i_1, i_2) = \frac{\sum_{j=1..m} w_j * similarité_{attribut=j}(i_1, i_2)}{\sum_{j=1..m} w_j} \quad (1)$$

La nature des attributs étant variée (numérique, textuelle, temporelle...) et dépendante du domaine applicatif, les formules de similarité inter-attributs doivent être adaptées en conséquence. Nous présentons dans la deuxième partie de cet article les formules que nous avons mises au point pour des items de type musicaux.

(Smyth, 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 Diversité :

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

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 appelée Diversité relative (RD) et se calcule avec l'équation suivante :

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

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

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

(Bradley, 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'items 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 calculée selon l’équation 5 :

$$Qualité(i, t, C) = Similarité(i, t) * RD(t, C) \quad (5)$$

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, 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) et (Jaher *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 étudiée 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. De ce fait, les métriques de diversité sont calculées à partir des notes. La métrique 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 phase de recommandation :

- augmenter la taille de la liste des items recommandés ;
- ajouter des items choisis aléatoirement ;
- mélanger les stratégies de recommandation (systèmes hybrides) ;
- trouver des clusters proches des clusters d’intérêt d’une personne donnée, permettant alors de recommander des items différents tout en minimisant le risque de proposer des items peu intéressants pour l’utilisateur (Onuma, Faloutsos, 2009) ;
- sélectionner des items dans la longue traîne<sup>5</sup> ayant le moins de chances de déplaire ;
- compenser le manque de notes (*sparsity*) en notant des clusters d’items plutôt que de les noter individuellement (Park, Tuzhilin, 2008).

Notons que malgré l’impact positif de la diversité sur la satisfaction des utilisateurs, la dimension temporelle, ainsi que le contexte, ne sont pas pris en compte dans les systèmes actuels de recommandations à base de diversité.

---

5. La longue traîne (*long tail*) définit pour un grand ensemble d’items, le sous-ensemble des items très étant peu consultés. A l’inverse, la tête est l’ensemble des items abondamment consultés.

#### 1.4. 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, 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.

##### 1.4.1. Définition

Le contexte est un concept multidimensionnel utilisé dans de nombreux domaines tels que la médecine, le droit ou encore l'économie. De ce fait, il existe une pléthore de définitions du contexte et (Bazire, Brezillon, 2005) concluront après avoir examiné plus de 150 définitions : "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 affectant 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'utilisateur pouvant influencer sur son comportement".

##### 1.4.2. Contexte et systèmes de recommandation

Tout comme il existe de nombreuses définitions du contexte, les recommandations s'inscrivent dans de nombreux domaines applicatifs, donnant 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 pouvant 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 côté de l'utilisateur courant, les objets présents autour de ce dernier ainsi que les changements de ces deux éléments (Schilit, Theimer, 1994). (Brown, 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 la définition suivante du contexte : "toute information permettant une caractérisation et dont la connaissance s'avère 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 consti-

tue 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 plus fidèlement possible ces derniers, permettant ainsi d'augmenter la satisfaction des utilisateurs à travers les recommandations proposées. Tout comme pour les systèmes de recommandation mobiles, des informations de natures plus concrètes et explicites comme l'heure, la météo et l'entourage de l'utilisateur à recommander sont également utilisées pour améliorer les recommandations. Par exemple, (Oku *et al.*, 2006) utiliseront de telles informations afin de fournir des recommandations dans un restaurant.

Quel que soit le domaine d'application, la prise en compte du contexte passe par une phase importante de récupération d'informations. Selon (Adomavicius, Tuzhilin, 2011), il existe trois moyens de collecter des informations contextuelles :

- la collecte explicite, dont les informations récupérées ont été obtenues par des interactions directes sous forme de questions. Par exemple, le site en ligne *Stereomood*<sup>6</sup> demande explicitement à l'utilisateur de renseigner son humeur afin de proposer des playlists adaptées ;
- la collecte implicite a pour but de récupérer les informations d'un utilisateur issues des interactions avec un système comme par exemple sa géolocalisation, l'heure à laquelle le système est utilisé, ou le temps et le nombre de consultations des items ;
- 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, adolescent, enfant...) d'un utilisateur 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. La plupart du temps, les contextes sont représentés sous une forme d'ontologies (Chen, Chen, 2014) afin de les utiliser dans les systèmes de recommandation. La construction de l'ontologie peut être automatique (Schickel-Zuber, Faltings, 2007) (avec l'inconvénient d'être coûteux et complexe), mais le plus fréquemment, elle nécessite une intervention humaine. De plus, l'utilisation d'ontologie aboutit à des systèmes de recommandation *ad hoc*. Par ailleurs, essayer de connaître le contexte de recherche d'une personne peut parfois s'avérer 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

---

6. [www.stereomood.fr](http://www.stereomood.fr)

utilisateurs (Cranor, 2005).

### 1.5. Définitions

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 cet article. En effet, nous nous inspirons des définitions proposées pour l'élicitation des préférences (les préférences implicites et explicites) (Castagnos, 2008) afin de les étendre à la notion de contexte. Nous proposons ces nouvelles définitions du contexte car nous pensons que tout comme les préférences des utilisateurs, le contexte peut lui aussi prendre plusieurs formes :

**Contexte implicite** : ensemble des caractéristiques communes aux items consultés pendant un certain laps de temps. Par exemple, dans le domaine du cinéma, les caractéristiques peuvent être le genre des films, les réalisateurs ou encore les acteurs des films consultés. Un contexte implicite peut être dans ce cas la consultation de films qui ne sont que des films de type "thriller" et qui sont réalisés par Martin Scorsese.

**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, moyens à disposition), etc. Comme nous l'avons vu précédemment, la plupart du temps, les critères du contexte explicite sont représentés sous la forme d'une ontologie et peuvent être en conflit avec le respect de la vie privée des utilisateurs.

Dans cet article, nous ferons également référence à la notion d'événement que nous définissons ainsi :

**Événement** : toute action effectuée par un utilisateur dont nous avons la trace ou que nous pouvons inférer.

### 1.6. 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 les besoins réels des utilisateurs et intègrent de plus en plus les facteurs humains. Parmi les facteurs humains existants, le besoin en diversité fait partie des champs d'investigation actuels dont les

travaux se sont avérés concluants et prometteurs. Proposer des recommandations diverses améliore la satisfaction des utilisateurs en permettant d'éviter une sclérose des recommandations (absence de nouveauté), de combler les incertitudes et de proposer des alternatives dans les recommandations. Cependant, les dimensions quantitatives et temporelles de la diversité restent des points à éclaircir, d'autant que ces derniers semblent être corrélés. En effet, comme cela a été démontré, la quantité de diversité requise par les utilisateurs évolue au cours du temps. À 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 cet article.

Par ailleurs, (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 les contextes des utilisateurs. Comme nous l'avons défini, un contexte implicite correspond aux caractéristiques communes partagées par les items consultés pendant un laps de temps. Sur la base de cette définition, un changement de contexte implicite doit donc se traduire par une modification remarquable du niveau de diversité. Nous faisons l'hypothèse selon laquelle ce changement peut être détecté par un modèle analysant l'évolution de la diversité au cours du temps. Un tel modèle pourrait alors permettre de détecter les changements de contexte implicite.

En résumé, notre objectif est d'enrichir le modèle utilisateur en étant capable de détecter automatiquement le contexte implicite sur la base de la diversité. Nous chercherons également à voir s'il existe un lien entre les contextes implicites détectés par le modèle que nous allons présenter et les contextes explicites existants. Cela suppose dans un premier temps de mettre au point un modèle permettant de modéliser l'évolution de la diversité au cours du temps (le modèle DANCE), 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 ("*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 illustrons dans un premier temps la capacité de DANCE à modéliser la diversité au cours du temps. Dans un deuxième temps, nous expliquons pourquoi et comment notre modèle peut être utilisé afin de détecter et de caractériser les contextes. Nous présentons les performances obtenues sur la base d'un corpus de plus de 200 000 écoutes. Enfin, nous démontrons 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

Les hypothèses défendues au cours de cet article 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 n'ayant 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.

#### 2.1.1. Notation

Nous introduisons dans cette section la notation que nous utilisons pour présenter notre modèle.

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

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

$C_k^u = \{c_1^u, c_2^u, \dots, c_k^u\}$  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 d'un tuple :

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

$A = \{a_1, a_2, \dots, a_h\}$  est l'ensemble des attributs d'un item.

#### 2.1.2. Description du modèle

Lors de notre état de l'art, nous avons constaté qu'il est possible de calculer la diversité existante entre deux items, dans un ensemble d'items, ou la diversité apportée par un item relativement à un ensemble d'items (Smyth, McClave, 2001), (Ziegler *et al.*, 2005). Cependant, il n'existe aucun modèle permettant de modéliser l'évolution de la diversité au cours du temps. Nous avons donc mis au point le premier modèle permettant de calculer la diversité apportée par chacun des items dans une séquence de consultation.

Pour notre modèle, nous considérons l'ensemble des consultations d'un utilisateur comme une suite d'items contiguë. Les approches reposant sur un modèle 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 supposons 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). Notre modèle calcule pour chaque

pas de temps, la diversité apportée par cet item relativement à l’historique de taille  $k$  précédemment consulté. Pour ce faire, nous utilisons la métrique de diversité relative (RD) que nous avons présenté lors de l’état de l’art :

$$RD(i, C_u) = \begin{cases} 0 & \text{si } C = \{ \}, \\ \frac{\sum_{j=1..m} (1 - \text{sim}_a(c_i^u, c_j^u))}{m} & \text{sinon.} \end{cases} \quad (6)$$

La Figure 3 illustre le fonctionnement de notre modèle pour une séquence de consultations musicales. Chaque morceau de musique (item) est composé d’attributs tels que le genre, la durée ou la popularité à partir desquels la similarité et RD sont calculées.

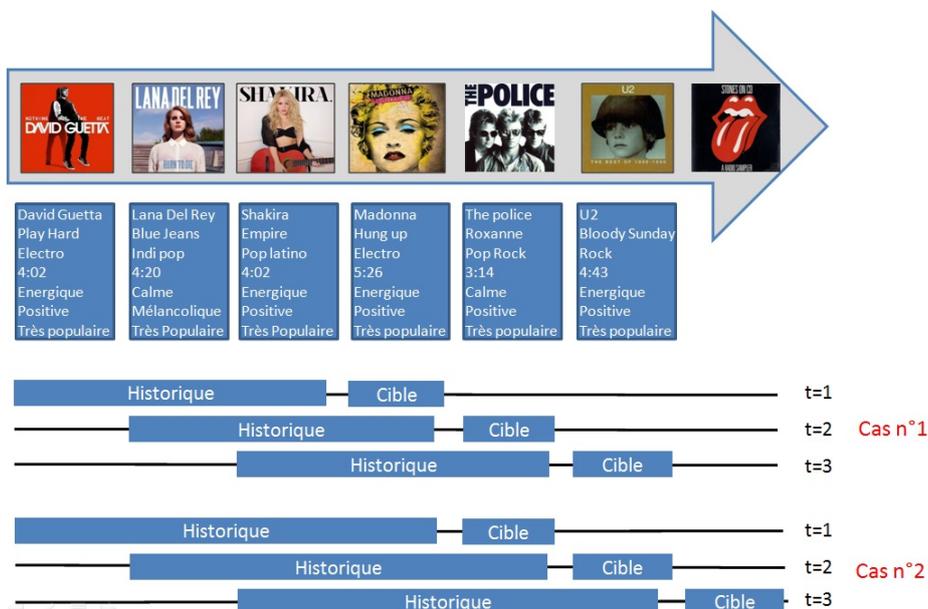


Figure 3. Principe du modèle DANCE

La cible  $c_t^u$  correspond à l’item courant pour lequel RD est calculée. L’historique correspond aux  $k$  dernières consultations  $\langle c_{t-k}^u, \dots, c_{t-2}^u, c_{t-1}^u \rangle$  par rapport auxquelles va être calculée. Dans le cas n° 1 (cf. Figure 3), la diversité apportée par l’item cible est calculée sur un historique de taille 3 alors que dans le cas n° 2, l’historique est de taille 4.

RD étant calculée à partir des attributs des items, il est possible de modéliser l’évolution de la diversité d’une séquence de consultations pour un seul attribut ou pour une combinaison d’attributs donnée.

Afin de pouvoir calculer RD, il est nécessaire de définir les formules de similarité par

attribut en fonction du type d’item sur lequel le modèle sera utilisé. Dans le cadre de cet article, nous avons utilisé un corpus musical et de ce fait, nous avons adapté nos formules de similarité en conséquence. Les formules que nous avons mises au point afin de calculer les similarités sur des attributs musicaux sont présentées dans la partie suivante.

### 2.1.3. Adaptation du modèle au domaine de la musique

Dans cette section, nous présentons la démarche que nous avons suivie afin d’appliquer notre modèle sur un corpus musical.

Nous avons utilisé un corpus de musique car les musiques nous offrent de nombreux avantages. Premièrement, la durée de consultation des items est relativement courte (contrairement aux films par exemple), impliquant de ce fait qu’il existe des séquences de consultations pour lesquelles le contexte explicite ne change pas. En effet, contrairement au cas des films pour lesquels le contexte explicite (par exemple les conditions dans lesquelles le film est vu : lieu, ambiance, entourage...) peut fortement varier d’un item à l’autre, les musiques sont moins soumises à cet effet. Deuxièmement, les sessions de consultation 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, une musique possède un temps de consultation (sa durée) indépendant de l’utilisateur (bien qu’elle puisse être écoutée en entier ou non, une musique ne peut pas être “consommée” plus rapidement d’un utilisateur à l’autre, contrairement à un site web).

Afin d’obtenir un nombre suffisant de consultations musicales, nous avons utilisé l’API de lastfm.fr<sup>7</sup> pour récupérer les écoutes de 1 000 utilisateurs entre juin 2005 et octobre 2013. Une écoute correspond à un tuple 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 sous forme d’un *timestamp*. 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’Echonest<sup>8</sup>. 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 être récupérées par l’API d’Echonest<sup>9</sup>. Pour chaque musique, nous avons ainsi pu récupérer 13 attributs nous permettant de calculer une valeur RD. Les détails sur le calcul de la valeur des attributs ne nous sont pas fournis. A titre d’exemple, nous savons que l’attribut “popularité” est un indice compris entre 0 et 1 témoignant de la popularité générale d’un groupe de musique. La “hottness”

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

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

9. Parmi les 1 000 utilisateurs initiaux, bon nombre d’entre eux avaient écouté des artistes inconnus par l’API d’Echonest. Le manque d’informations sur une musique rendant difficile le calcul de RD, nous avons choisi de supprimer les utilisateurs ayant trop de musiques inconnues.

d'un artiste, quant à elle, détermine si cet artiste est populaire au moment où la requête est faite. Par exemple, le groupe "The Beatles" a une popularité très élevée (popularité = 0,90) et une actualité assez bonne (hotttness=0,77) alors que "Rihanna" sera moins populaire (popularité = 0,84) mais plus actuelle (hotttness = 0,83). Pour plus de détails sur les attributs que nous avons utilisés, nous vous invitons à visiter le site d'Echonest<sup>8</sup>.

Le corpus utilisé pour l'ensemble des calculs est composé de 100 utilisateurs ayant écouté 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 :

Tableau 1. Statistiques du corpus

Attributs	musique							artiste	
	durée	tempo	mode	loudness	énergie	hotttness	danceability	hotttness	familiarity
max	4 194	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

Pour les items de notre corpus, il existe 5 types d'attributs : les attributs sous forme de listes, les attributs binaires, les attributs numériques, les attributs sous forme d'intervalles (années d'activité des artistes) et les attributs géographiques (coordonnées GPS des artistes). Nous avons donc défini les formules suivantes :

FORMULE DE SIMILARITÉ POUR LES ATTRIBUTS SOUS FORME DE LISTES

$$Similarité_{attribut=a}(c_t^u, c_{t-1}^u) = \frac{card(c_t^u.a \cap c_{t-1}^u.a)}{\min(card(c_t^u.a), card(c_{t-1}^u.a))} \quad (7)$$

Cette formule permet de calculer la similarité entre deux musiques pour les attributs "artistes similaires" et "termes" (genre) associés aux artistes.  $card(c_t^u.a)$  correspond au nombre d'éléments caractéristiques de l'item consulté par l'utilisateur au temps  $t$  pour l'attribut  $a$ . Par exemple, dans le cas où l'on travaille sur les termes associés aux artistes,  $card(c_t^u.a)$  correspond au nombre de termes associés à cet artiste pour la musique écoutée au temps  $t$ . L'utilisation du minimum de la cardinalité au dénominateur est utilisé car les artistes ne possèdent pas tous le même nombre de termes.

FORMULE DE SIMILARITÉ POUR LES INTERVALLES

$$Similarité_{attribut=a}(c_t^u, c_{t-1}^u) = \frac{card(c_t^u.a \cap c_{t-1}^u.a)}{\max(card(c_t^u.a), card(c_{t-1}^u.a))} \quad (8)$$

La formule permettant de calculer la similarité pour des attributs sous la forme d'intervalles (pour les années d'activité des artistes dans notre cas) repose sur l'intersection des intervalles des items considérés. Cependant, nous avons décidé d'utiliser la valeur maximale de la cardinalité du nombre d'années au dénominateur car nous partons du

principe que deux artistes sont exactement similaires si et seulement s'ils ont été en activité pendant la même période exactement <sup>10</sup>.

FORMULE DE SIMILARITÉ POUR LES ATTRIBUTS BINAIRES

$$Similarité_{attribut=a}(c_t^u, c_{t-1}^u) = \begin{cases} 1 & \text{si } c_{t-1}^u \cdot a = c_t^u \cdot a, \\ 0 & \text{sinon.} \end{cases} \quad (9)$$

De part sa nature binaire, le mode ne prend que 2 valeurs : 0 ou 1. Si les musiques ont le même mode, alors elles sont exactement similaires sinon elles sont exactement dissimilaires (sur cet attribut).

FORMULE DE SIMILARITÉ POUR LES ATTRIBUTS NUMÉRIQUES

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

Les valeurs minimum ( $min_a$ ) et maximum ( $max_a$ ) permettent de normaliser les similarités en bornant le calcul par rapport au corpus. En effet, une différence de 5 minutes ne doit pas avoir le même impact sur la similarité en fonction du maximum et du minimum des durées des musiques du corpus. L'utilisation de la fonction exponentielle permet de considérer d'avantage les faibles différences de durée dans le calcul de la similarité.

FORMULE DE SIMILARITÉ POUR LES DISTANCES

$$Similarité_{attribut=a}(c_t^u, c_{t-1}^u) = 1 - \frac{distance(c_t^u, c_{t-1}^u)}{max\ distance} \quad (11)$$

La similarité pour l'attribut "coordonnées" se calcule à partir de la distance séparant les deux artistes. Cette distance est déterminée à partir des coordonnées (latitude et longitude) et en utilisant une formule géodésique permettant de calculer la distance minimale entre deux points sur la surface d'une sphère (la Terre dans notre cas). Le dénominateur correspond à la distance maximale entre deux artistes sur terre, c'est-à-dire 20 037,5 km (la moitié de la circonférence de la Terre).

#### 2.1.4. Résultats sur le corpus musical

En utilisant le corpus et les formules que nous avons présentés précédemment, nous avons modélisé l'évolution de la diversité au cours du temps pour chaque utilisateur. Nous avons donc calculé à chaque pas de temps la valeur de RD. Nous avons également calculé les valeurs dérivées de RD que nous appellerons RD' :

$$RD'(c_t^u) = RD(c_t^u) - RD(c_{t-1}^u) \quad (12)$$

10. Un artiste ayant été actif de 1986 à 1990 possède une similarité proche mais pas égale à 1 par rapport à un artiste ayant été actif de 1986 à 1998.

Le graphique de la Figure 4 ci-dessous représente l'évolution de RD et de RD' pour l'attribut "tempo" d'un utilisateur pour 125 de ses écoutes.

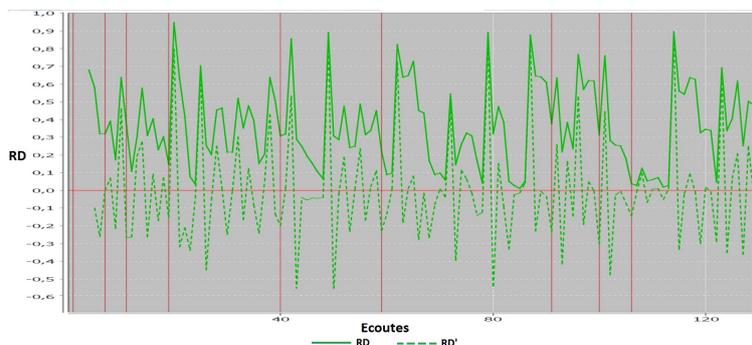


Figure 4. Évolution de RD et RD' sur l'attribut "tempo" pour une fenêtre de taille 3

Sur ce graphique, l'axe des abscisses représente les items consultés et l'axe des ordonnées représente la valeur de RD et de RD' de chaque item. Les traits pleins représentent la RD et les pointillés représentent RD'. Chaque trait vertical en rouge symbolise le début d'une session d'écoute. Nous pouvons constater que RD n'est pas une valeur stable et qu'elle évolue fortement au cours du temps. Une croissance continue de RD nous indique que les items consultés apportent de plus en plus de diversité par rapport à l'historique considéré et inversement. Bien que RD soit relativement instable, nous constatons qu'il existe certains moments où la diversité varie peu, nous indiquant que les items consultés n'apportent pas beaucoup de diversité.

A l'inverse, certaines consultations font chuter ou augmenter drastiquement RD. Lorsque RD diminue ou reste faible, l'utilisateur spécialise sa séquence de consultations (c'est-à-dire que les items consultés possèdent beaucoup d'attributs pour lesquels leurs valeurs sont similaires). Lorsque RD est importante, l'utilisateur a un comportement éclectique, il est ouvert à plusieurs tendances, et les items consultés sont divers.

L'évolution de RD peut également être visualisée en trois dimensions comme le montre la Figure 5 obtenue à partir de RD totale (prise en compte de tous les attributs). Ce graphique nous permet de voir l'impact de la taille de la fenêtre d'observation sur la valeur de RD. La taille de la fenêtre ne semble pas modifier de manière significative les valeurs de RD. Malgré cela, les variations de RD sont plus prononcées pour un historique de taille 1. 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. Pour la suite de nos calculs, nous utiliserons une fenêtre de taille 3, car elle offre un bon compromis entre détection des variations importantes et sensibilité au bruit (faibles variations de diversité).

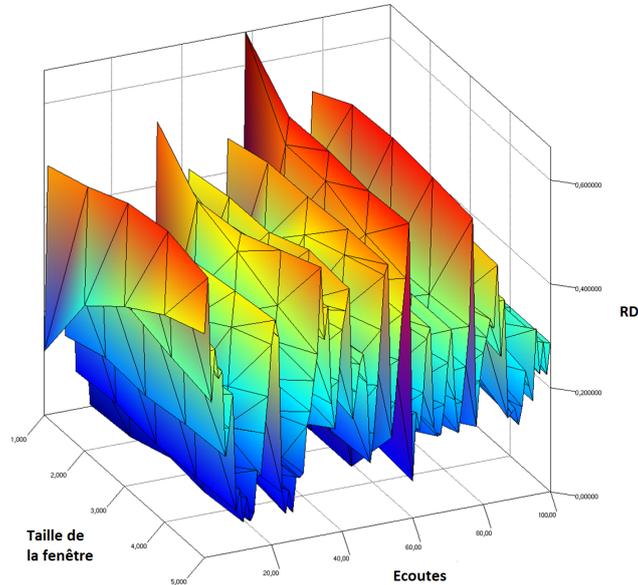


Figure 5. Évolution de RD totale selon différentes tailles de fenêtres

**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 transposé dans d'autres domaines. Les formules de similarité par attribut 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 que nous avons fixée (cf. section 2.1.1).

Même si nous avons fixé la taille de la fenêtre à 3 pour les raisons citées précédemment, la taille de la fenêtre optimale ne pourra être déterminée qu'à l'issue de la confrontation de notre modèle avec des utilisateurs en situation réelle de recommandation. Cette confrontation constitue l'une de nos perspectives. En effet, nous avons besoin du retour des utilisateurs afin de savoir si notre paramétrage est trop ou pas assez sensible pour la détection des contextes implicites.

Comme nous l'avons défini précédemment, au sein d'un contexte (qu'il soit implicite ou explicite), les items consultés partagent des caractéristiques communes. À 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 implicite à un autre.

## 2.2. Détection du contexte : approche naïve

Un contexte implicite 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 implicite est relativement faible. Au contraire, lors d'un changement de contexte implicite, les caractéristiques des items consultés n'étant plus les mêmes, la valeur RD apportée par les items du nouveau contexte est plus importante. C'est sur ce principe que nous avons basé notre modèle, afin de détecter les changements de contexte implicite.

Notre modèle cherche donc à détecter les consultations réunissant ces deux conditions :

1. RD de l'item cible est supérieure à RD de l'item précédent et de l'item suivant, cela signifie qu'il y a un pic de diversité ( $C_1$ );
2. la variation de diversité apportée par l'item courant (RD') doit être supérieure à un seuil fixé dans notre modèle, cela signifie que la diversité apportée par l'item cible est importante ( $C_2$ ).

Le pseudo-code simplifié de l'algorithme 1 ci-dessous présente le fonctionnement de notre modèle.

---

### Algorithme: 1 Pseudo-code simplifié du modèle DANCE

---

```

sessions_détectées, musiques_passées_détectées, contextes_détectés = 0
k = taille de la fenêtre d'observation
pour tous les items i faire
  calculer RD(i,k)
  RD'(i) = RD'(i) - RD'(i-1)
  si RD(i-1) < RD(i) > RD(i+1) et RD'(i) > seuil alors
    si i est un début de session alors
      sessions_détectées ++
    sinon si i a été passé alors
      musiques_passées_détectées ++
    sinon
      contextes_détectés ++
  fin si
fin si
retourner sessions_détectées, musiques_passées_détectées, contextes_détectés

```

---

Notre corpus initial est divisé en deux corpus contenant tous les items (c'est-à-dire toutes les musiques écoutées), mais ne contenant pas les mêmes informations :

- Le **corpus d'apprentissage** dans lequel nous connaissons pour chaque utilisateur les items consultés, les informations sur ces items (i.e. leurs attributs) et le *timestamp* de ces consultations ;
- Le **corpus de test** contenant la liste des sessions d'écoute et l'information déterminant si une musique a été passée ou non (les sessions et les musiques passées ont

pu être déterminées grâce aux *timestamps*).

Le but de l'algorithme 1 est de détecter les changements de contexte implicite se produisant lorsque les deux conditions  $C_1$  et  $C_2$  sont réunies. Une fois un changement de contexte implicite détecté à partir des informations du corpus d'apprentissage, l'algorithme regarde si ce changement est dû à un événement en utilisant les informations contenues dans le corpus de test.

La finalité de notre algorithme n'est pas de détecter les fins de session et les musiques passées, puisque ces dernières peuvent être obtenues par l'analyse des traces. Notre algorithme est un moyen de répondre à la question suivante :

**Existe-t-il des concordances entre les événements (révèlateurs potentiels des changements de contexte explicite) et les changements de contexte implicite détectés par notre algorithme ?**

L'algorithme 1 est donc un moyen de répondre à cette question en cherchant à voir si les contextes implicites détectés et les changements de contexte explicite (induits entre autre par les fins de session) coïncident.

### 2.2.1. Paramètres utilisés

Afin de pouvoir tester les performances de détection de notre modèle, nous avons déterminé les sessions d'écoute de chacun des utilisateurs de notre corpus. En effet, l'API que nous avons utilisée ne nous a pas permis d'obtenir explicitement les moments où se situent les fins de session (*i.e.* les déconnexions ou moments d'inactivité). Connaissant les heures exactes auxquelles les musiques ont été écoutées grâce aux *timestamp*, ainsi que les durées des musiques, nous avons pu inférer les sessions d'écoute de notre corpus. Nous considérons qu'une session d'écoute est constituée d'un enchaînement de musiques pour lesquelles il se passe moins de 15 minutes entre la fin d'une musique et le début de la suivante.

À 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 abandonnée (passée) si la durée d'écoute est inférieure ou égale à 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 RD et le seuil de RD' est fixé à 0,05 (cf. supra, 2<sup>ème</sup> 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 ci-dessus sont résumés dans le Tableau 2 :

Tableau 2. Résultats de l'approche naïve

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

Avec cette approche, nous avons cherché à vérifier si les changements de contexte identifiés par notre algorithme correspondaient à des événements particuliers tels que les fins de session ou les chansons passées par les utilisateurs.

Notre modèle basé sur les variations de diversité nous permet d'induire des changements de contexte implicite. Or, s'il n'y a pas de changement de contexte implicite entre deux sessions, notre modèle ne pourra pas détecter la nouvelle session car selon notre définition du contexte implicite, la diversité varie très peu au sein d'un même contexte. Ceci est la raison pour laquelle il convient de préciser que notre modèle ne permet pas d'arriver à 100 % de détection de fin de session. Cette première approche de notre modèle permet de détecter 77,41 % des sessions existantes.

Comme nous l'avons mentionné précédemment, notre modèle n'a pas pour but de détecter les sessions des utilisateurs mais les contextes implicites. La détection des sessions nous permet, à partir du postulat selon lequel il existe un changement de contexte explicite entre deux sessions, de vérifier que l'on pouvait détecter ces changements de contexte explicite.

Nous constatons que nous avons obtenu 12 925 maximums locaux ne correspondant ni à un changement de session, ni à une musique passée. Ces maximums locaux nous permettent de poser l'hypothèse qu'il existe plusieurs contextes implicites au cours d'une même session et que de tels changements de contexte implicite sont détectables par notre modèle.

Le nombre de musiques passées détectées par notre modèle est relativement faible : seuls 17,53 % des musiques passées ont été à l'origine d'un fort pic de diversité. Cela signifie qu'une musique n'est pas nécessairement passée car elle ne correspond pas au contexte implicite courant. Les raisons pouvant pousser un utilisateur à ne pas écouter une musique entièrement peuvent être multiples : la musique correspond au contexte implicite mais n'est pas appréciée, la musique est appréciée mais l'utilisateur n'avait pas envie de l'écouter entièrement ou à ce moment précis, l'utilisateur a envie d'écouter autre chose car cette musique lui a fait penser à une autre musique, etc.

### 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 (notamment les changements de sessions dont la détection peut être due à un changement de contexte explicite étant à l'origine du changement remarquable des caractéristiques des items consultés). L'hypothèse **H2** est donc validée. En outre, il est raisonnable de s'interroger sur le choix des attributs à prendre en compte lors de la mise en place d'un tel modèle. Ainsi, bien que

l'utilisation de l'attribut "coordonnées" soit surprenante, cela n'en est pas moins intéressant. Utiliser cette information nous a permis de montrer dans un premier temps que la diversité pouvait être calculée à partir de n'importe quelle source d'information. De plus, dans certains cas, cet attribut peut avoir du sens. Par exemple, nous pourrions imaginer dans une séquence de consultations, que le seul attribut possédant une évolution en diversité faible soit les coordonnées si l'utilisateur n'écoute que des groupes locaux. Il faudra alors davantage tenir compte de cet attribut pour les recommandations, puisque qu'il semble être important pour cet utilisateur dans ce contexte implicite.

### 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 implicite. La configuration utilisée précédemment avec l'approche naïve attribuait le même poids à chaque attribut dans le calcul de 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 explicite 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 l'attribut "durée" pour le calcul de RD. Afin de pouvoir vérifier si certains attributs sont plus discriminants que d'autres, nous devons tester différentes configurations des poids des attributs de notre algorithme afin de voir quelles en sont les conséquences sur les performances de détections. Nous souhaitons donc approfondir notre analyse autour de trois questions de recherche :

1. Peut-on privilégier la détection de certains événements en changeant la configuration des poids de notre algorithme ?
2. L'algorithme est-il stable si on modifie sa configuration (i.e. les poids élevés sont-ils toujours les mêmes) ?
3. Y-a-t-il une configuration optimale (i.e. toujours meilleure 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 entre 0 et 1 et sachant que nous disposons de 13 attributs, tester toutes les configurations de poids possibles reviendrait à tester  $10^{26}$  configurations. Étant donné cet ordre de grandeur, il est impossible d'envisager de tester toutes ces configurations. 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. En utilisant un algorithme génétique, nous ne cherchons pas à trouver le meilleur paramétrage possible mais nous nous servons de cette approche afin de balayer un large espace de combinaisons dans le but de savoir s'il est possible ou non d'améliorer les performances de notre modèle.

Les algorithmes génétiques ont été mis au point dans les années soixante pour répondre à des problèmes ayant de très nombreuses combinaisons (dont le plus étudié est le problème du voyageur de commerce (Sonam, Puneet, 2014)). Ce type d'algorithmes fait partie des algorithmes évolutionnistes, c'est à dire que la solution est trouvée via la sélection des individus offrant les meilleures performances.

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. Nous rappelons que pour le calcul de RD, nous utilisons auparavant les valeurs de RD de chacun des attributs, où chacun des attributs avait le même poids dans le calcul de RD. En utilisant des poids différents, certains attributs impacteront plus ou moins fortement le calcul de RD. Un chromosome correspond à la valeur des poids attribués à chacun des attributs. Un gène correspond au poids d'un attribut. Ainsi, chaque chromosome possède treize gènes. La performance de chaque chromosome par rapport au problème posé est évaluée à l'aide d'une fonction de fitness (phase d'évaluation). Dans notre cas, la performance est la capacité à détecter certains événements au travers des contextes implicites que nous détectons. 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 n'étant présent pas dans la population initiale. Une nouvelle génération de chromosomes sera donc créée à partir 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.

Bien qu'ils reposent sur des bases communes, il existe différentes variantes au sein des mécanismes que nous venons de présenter (Mallipeddi *et al.*, 2011). 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 reposant sur un tirage aléatoire entre tous les individus mais où les individus ayant les meilleurs scores ont plus de chance d'être sélectionnés que les autres.

Il convient de préciser que nous ne cherchons pas à mettre au point l'algorithme génétique nous permettant d'obtenir les meilleures performances de détection possibles, mais que nous utilisons un algorithme génétique afin de pouvoir répondre aux questions de recherche que nous avons citées précédemment.

### 2.3.2. Paramètres utilisés

Lors de la mise au point de notre algorithme génétique, nous avons fait en sorte que ce dernier ne converge pas trop rapidement vers un maximum local, tout en favorisant les chromosomes ayant la plus grande probabilité de succès.

Les paramètres que nous avons utilisés afin de pouvoir respecter les conditions précé-

demment citées 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 (correspondant aux poids des attributs pour le calcul de *RD*) ;
- **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 correspondant à la somme des scores des chromosomes le composant. Ensuite, selon une loi de Poisson avec un coefficient de 2,12<sup>11</sup>, 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 générer sera grand. Ce mécanisme permet de privilégier au sein de la population, les gènes permettant d'obtenir de bons scores sans pour autant tomber dans une forme d'eugénisme dont le principe serait néfaste pour la convergence de l'algorithme ;
- **Reproduction** : Lors de la reproduction, les deux chromosomes des couples formés précédemment vont générer des enfants en s'échangeant une partie de leurs gènes à hauteur de 70 % et les parents ne seront pas inclus dans la génération suivante ;
- **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 prennent les valeurs du chromosome ayant obtenu le meilleur score. Les tests ont été effectués pour différents seuils de *RD*. Le Tableau 3 ci-dessous présente les résultats obtenus sur 50 générations de chromosomes :

Tableau 3. Résultats avec des coefficients compris entre 0 et 1

	Evts existants	Seuil <i>RD</i>				
		0	0,05	0,10	0,15	0,20
Fins de session	18 412	17 737	<b>16 288</b>	15 776	14 450	1877
Contextes implicites	-	22 104	<b>17 412</b>	15 776	14 533	11 075
Musiques passées	15 008	3 915	3 137	2 807	2 448	2 050

Nous pouvons constater dans un premier temps que le paramétrage des coefficients augmente les performances de détections de notre modèle. En effet, le nombre de ses-

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

sions détectées passe de 14 253 (soit 77,41%) pour des poids de 1 (cf. supra, Tableau 2, p.24), à 16 228 (soit 88,46%) (Tableau 3 ci-dessus) pour des poids compris entre 0 et 1. Cela confirme l'hypothèse selon laquelle la détection de certains événements peut être privilégiée et que certains attributs sont plus déterminants que d'autres pour détecter les contextes (question de recherche n° 1, p.25).

Nous avons également essayé de paramétrer notre modèle afin de privilégier la détection des musiques ayant été passées. Cependant, les détections de ces dernières n'ont pas pu être améliorées, confirmant que les musiques passées sont difficilement détectables à l'aide de l'évolution de la diversité.

Le seuil de RD' impacte également significativement les performances de détection, comme le montrent les résultats du Tableau 3. Plus ce dernier est petit et plus les faibles variations de diversité sont détectées. Inversement, plus ce seuil est important, plus les changements de contexte implicite doivent être marqués par une variation de diversité importante pour être détectés. C'est pourquoi le nombre de contextes implicites détectés augmente lorsque l'on diminue le seuil de RD'. Ainsi, une faible valeur de RD' améliore la détection des fins de session, mais rend également notre algorithme plus sensible aux changements de contexte implicite de faibles amplitudes au sein des sessions. Le choix de la valeur de RD' se résume donc à trouver le meilleur compromis entre la sensibilité de la détection et la pertinence des contextes implicites détectés. À 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.

Enfin, 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. Nous avons également vérifié si la configuration des poids des attributs à la fin des 50 générations n'était pas due à l'initialisation de notre population. À l'issue de chaque lancement de notre algorithme génétique, nous avons classé les poids des attributs par ordre décroissant et nous avons récupéré la position de chaque attribut (plus le poids d'un attribut est grand, plus sa position sera petite et réciproquement). Le but est de vérifier si les attributs ayant les poids les plus importants sont les mêmes après chaque exécution et chaque initialisation.

Le Tableau 4 ci-dessous 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. Tableau 4). Par ailleurs, les attributs avec des poids faibles sont également souvent les mêmes. Précédemment, nous avons fait une remarque

Tableau 4. Rang des poids des attributs

Attribut	0		0,05		0,10	
	Moyenne	Écart type	Moyenne	Écart type	Moyenne	Écart type
Mode	<b>2,95</b>	0,39	<b>1</b>	0	<b>1</b>	0
Artistes similaires	<b>3,95</b>	0,22	<b>2,35</b>	0,49	<b>2</b>	0
Énergie (musique)	5,05	0,22	<b>2,65</b>	0,49	7,3	2,29
Années d'activités	7,25	0,44	<b>4,75</b>	1,74	<b>3</b>	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	<b>1,05</b>	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)	<b>2,05</b>	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

concernant les coordonnées et nous pouvons constater que dans le cas où nous cherchons à optimiser la détection des changements de session, cet attribut est l'un de ceux ayant le moins d'impact (d'où son faible poids). Cependant, dans le cas où l'on cherche à diversifier les recommandations, jouer sur les valeurs de cet attribut pourrait être essentiel. Cela valide l'hypothèse selon laquelle le modèle DANCE est stable (question de recherche n° 3, p.25) et qu'il existe une configuration optimale consistant à maximiser certains poids.

#### 2.3.4. Discussion

La conception d'un modèle reposant sur 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 par notre modèle DANCE avec des poids sur les attributs ont été améliorées par rapport à l'approche naïve. Pour un même seuil de RD' (0,05), 2 000 sessions supplémentaires ont pu être ainsi détectées (cela représente 88,41% de fins de session détectés si des poids sont attribués, contre 77,41% avec l'approche naïve). 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. Les résultats obtenus nous permettent donc de valider l'hypothèse **H3** car notre modèle est stable et peut effectivement être paramétré pour privilégier la détection de certains événements.

Lorsque nous avons classé les poids des attributs en fonction de leur rang, nous avons pu constater que certains attributs reviennent toujours en tête du classement

(“tempo”, “mode”, “artistes similaires” et “énergie”). Nous en avons conclu qu’ils étaient plus déterminants que d’autres dans le sens où, lorsque l’on cherche à maximiser la détection des changements de session, ce sont ces attributs ayant le plus d’impact. Ces attributs ont plus d’impact parce qu’ils témoignent d’une forte augmentation de diversité lors d’un changement de session contrairement à certains autres attributs (coordonnées, durée, hottness) ne semblant pas rendre compte de manière significative d’un changement de session. Donner du sens à la liste des attributs les plus déterminants est délicat car toute interprétation serait subjective. Néanmoins, les rangs fournis par notre algorithme génétique pourront être exploités pour fournir une explication à l’utilisateur courant. A titre d’exemple, il sera difficile de trouver une justification (autre que statistique) à l’adéquation entre variation d’“énergie” et fins de session. En revanche, il sera possible de dire à l’utilisateur qu’une nouvelle recommandation lui est faite pour tenir compte de son changement de comportement constaté vis-à-vis de l’attribut “énergie”.

Enfin, il est important de noter que nous ne proposons pas un modèle permettant de détecter les fins de session. En revanche, il existe un recouvrement important entre les fins de session et les changements de contexte implicite. De plus, au sein d’une session, il peut également y avoir un ou plusieurs changements de contexte implicite. Par exemple, lorsque l’utilisateur est lassé par la musique qu’il écoute et souhaite en changer. Notre modèle permet de détecter de fortes variations de diversité que nous attribuons à ces changements de contexte implicite au sein d’une session. Toutefois, notre corpus n’étant pas labélisé, nous ne pouvons pas évaluer la proportion de ces variations attribuables à des faux positifs. Cela ne remet pas en cause la qualité de notre modèle puisqu’il s’agit de variations de diversité avérées. Les tests utilisateur que nous réaliserons dans un futur proche nous permettront de distinguer automatiquement les changements de contexte implicite des faux positifs. Pour ce faire, nous intégrerons *a posteriori* les retours des utilisateurs comme fonction de récompense dans notre algorithme génétique.

### 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 cet article, 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 faible complexité algorithmique et d’autre part, d’être utilisé afin de détecter les changements de contexte

implicite des utilisateurs. Détecter rapidement un changement de contexte implicite permettra de réagir et d'adapter les recommandation en conséquence. La force de notre modèle réside dans sa capacité à pouvoir détecter les similarités et les dissimilarités existantes 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, seule une connaissance des items et de leur ordre de consultation suffisent à notre modèle. Notre modèle ne nécessite qu'un historique de petite taille pour détecter les changements de contexte implicite.

Nous avons donc proposé un modèle formel, innovant et générique pour comprendre le contexte implicite de l'utilisateur en observant le niveau de diversité au sein des séquences 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 session.

### 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 de nombreuses perspectives. 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, recommandations...). Dans un second temps, notre modèle pourrait être utilisé à des fins de recommandation afin de fournir des recommandations adaptées aux contextes des utilisateurs et pour mesurer leur satisfaction.

À 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. Il sera possible de personnaliser les recommandations sans pour autant être intrusif, puisque nous ne nous servons pas de données personnelles pour nous adapter au contexte. DANCE pourra être utilisé pour adapter les niveaux de diversité en fonction du besoin des utilisateurs, et également fournir des explications sur les recommandations (caractérisation du contexte implicite). Par exemple, si nous constatons qu'un utilisateur a des consultations très diverses (ou choisi des recommandations diverses), l'algorithme de recommandation pourra s'adapter à ce besoin et fournir des recommandations diverses. Inversement, si la personne semble très spécialisée dans ses consultations, l'algorithme pourra l'aider en identifiant les attributs déterminant ainsi que les plages de valeur adaptées à ses usages.

Comme nous l'avons précisé, le modèle peut être utilisé dans d'autres domaines comme

les réseaux sociaux professionnels tels que Yupeek<sup>12</sup> ou LinkedIn<sup>13</sup> où la recommandation occupe une place dominante. Appliquer notre modèle dans de tels domaines serait un moyen de confirmer le caractère générique de notre modèle. Une autre perspective consisterait à augmenter les capacités à décrire le comportement des utilisateurs. Une amélioration du modèle 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* de comportement des utilisateurs.

## Bibliographie

- Abdul-Rahman A., Hailes S. (2000, janvier). Supporting trust in virtual communities. In *Proceedings of the 33rd annual hawaii international conference on system sciences, 2000*, p. 9 pp. vol.1.
- Adomavicius G., Sankaranarayanan S., Sen S., Tuzhilin A. (2005). Incorporating contextual information in recommender system using a multidimensional approach. *Transaction on Information System*, vol. 23, p. 103–145.
- Adomavicius G., Tuzhilin A. (2011). Contexte-aware recommender systems. *Recommender Systems Handbook*, p. 217–253.
- Agrawal R., Gollapudi S., Halverson A., Ieong S. (2009). Diversifying search results. In *Proceedings of the second acm international conference on web search and data mining*. Barcelona, Spain.
- Bazire M., Brezillon P. (2005). Understanding context before using it. In *Proceedings of the 5th international conference on modeling and using context*. Springer-Verlag.
- Boim R., Milo T., Novgorodov S. (2011). Diversification and refinement in collaborative filtering recommender. In *Proceedings of the 20 acm internatioanal conference on information and knowledge management*. 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., Smith B. (2001). Improving recommendation diversity. In *irish conference on artificial intelligence and cognitive science*, p. 85–94. San Francisco, CA, USA.
- Brown P., Chen X. (1997). Context-aware applications: from the laboratory to the marketplace. In *Ieee*, p. 58–64.
- Burke R. (2002, novembre). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, vol. 12, n° 4, p. 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.

---

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

13. <https://www.linkedin.com/>

- Castagnos S., Brun A., Boyer A. (2013, avril). 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., Boyer A. (2014). La diversité : entre besoin et méfiance dans les systèmes de recommandation. In *Information interaction intelligence*.
- Castagnos S., Jones N., Pu P. (2010). Eye-tracking product recommenders' usage. In *Recsys*, p. 29–36.
- Chen G., Chen L. (2014). Recommendation based on contextual opinions. In V. Dimitrova, T. Kuflik, D. Chin, F. Ricci, P. Dolog, G.-J. Houben (Eds.), *User modeling, adaptation, and personalization*, vol. 8538, p. 61-73. Springer International Publishing.
- Clarke C. L., Kolla M., Cormack G. V., Vechtomova O., Ashkan A., Büttcher S. *et al.* (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*, p. 659–666. New York, NY, USA, ACM.
- Cosley D., Lam S. K., Albert I., Konstan J. A., 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*, p. 585–592. New York, NY, USA, ACM.
- Cranor L. (2005). Hey, that's personal! In L. Ardissono, P. Brna, A. Mitrovic (Eds.), *User modeling 2005*, vol. 3538, p. 4–4. Springer Berlin Heidelberg.
- Davis F. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. In *Mis quarterly* 13, p. 319–340.
- Department D., Bridge D. (2001). Product recommendation systems: A new direction. In *Workshop on cbr in electronic commerce at the international conference on case-based reasoning*, p. 79–86.
- Dey A., Abowd G., Salber D. (2001). A conceptual framework and a toolkit for supporting the rapid prototype of context-aware applications. In *Human-computer interaction*, p. 97–166.
- Foulonneau M., Groues V., Naudet Y., Chevalier M. (2014). Recommandeurs et diversité : Exploitation de la longue traîne et diversité des listes de recommandations. In G. Chartron, I. Saleh, G. Kembellec (Eds.), *Les systèmes de recommandation*, chap. 4. <http://www.editions-hermes.fr/>, Hermès.
- Goldberg D., Nichols D., Oki B. M., Terry D. (1992, décembre). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, vol. 35, n° 12, p. 61–70.
- Hasan M., Kashyap A., Hristidis V., Tsotras V. (2014). User effort minimization through adaptive diversification. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining*, p. 203–212. New York, NY, USA, ACM.
- Häubl G., Murray K. B. (2003). Preference construction and persistence in digital marketplaces: The role of electronic recommendation agents. *Journal of Consumer Psychology*, vol. 13, n° 1, p. 75–91.
- Jahrer M., Tscher A., Legenstein R. (2010). Combining predictions for accurate recommender systems. In *Acm international conference on knowledge discovery and data mining*, p. 693–702. Washington, DC, USA.

- Jawaheer G., Weller P., Kostkova P. (2014, juin). Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, vol. 4, n° 2, p. 8:1–8:26.
- 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., Hamad A. (2011). An exploratory work in using comparisons instead of ratings. In C. Huemer, T. Setzer (Eds.), *Ec-web*, vol. 85, p. 184–195. Springer.
- Kurki T., Jokela S., Sulonen R., Turpeinen M. (1999). Agents in delivering personalized content based on semantic metadata. In *Proc. 1999 aai spring symposium workshop on intelligent agents in cyberspace*, p. 84–93.
- Lathia N. (2010). *Evaluating Collaborative Filtering Over Time*. Thèse de doctorat non publiée, University of London, Department of Computer Science, University College London.
- Lathia N., Hailes S., Capra L., Amatriain X. (2010). Temporal diversity in recommender systems. In *Proceedings of the 33rd international acm sigir conference on research and development in information retrieval*, p. 210–217. New York, NY, USA, ACM.
- Leake D., Scherle R. (2001). Towards context-based search engine selection. In *Proceedings of the 6th international conference on intelligent user interfaces*, p. 109–112. New York, NY, USA, ACM.
- L’Huillier A., Castagnos S., Boyer A. (2014). Understanding usages by modeling diversity over time. In *Umap*.
- Liang S., Ren Z., Rijke M. de. (2014). Personalized search result diversification via structured learning. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining*, p. 751–760. New York, NY, USA, ACM.
- Lops P., Gemmis M. de, Semeraro G. (2011). Content-based recommender systems: State of the art and trends. In F. Ricci, L. Rokach, B. Shapira, P. B. Kantor (Eds.), *Recommender systems handbook*, p. 73–105. Springer.
- Mallipeddi R., Suganthan P., Pan Q., Tasgetiren M. (2011). Differential evolution algorithm with ensemble of parameters and mutation strategies. *Applied Soft Computing*, vol. 11, n° 2, p. 1679 - 1696. (The Impact of Soft Computing for the Progress of Artificial Intelligence)
- McGinty L., Smyth B. (2003). On the role of diversity in conversational recommender systems. In *Proceedings of the fifth international conference on case-based reasoning*, p. 276–290. Springer.
- McNee S. M., Riedl J., 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*, p. 1097–1101. ACM.
- McSherry D. (2002). Diversity-conscious retrieval. In *Proceedings of the 6th european conference on advances in case-based reasoning*, p. 219–233. London, UK.
- Middleton S. E., De Roure D. C., Shadbolt N. R. (2001). Capturing knowledge of user preferences: Ontologies in recommender systems. In *Proceedings of the 1st international conference on knowledge capture*, p. 100–107. New York, NY, USA, ACM.
- Murakami T., Mori K., Orihara R. (2008). Metrics for evaluating the serendipity of recommendation lists. In *Proceedings of the 2007 conference on new frontiers in artificial intelligence*, p. 40–46. Berlin, Heidelberg, Springer-Verlag.

- Oku K., Nakajima J., Uemura S. (2006). Context-aware svm for context-dependent information recommendation. In *Proceedings of the 7th international conference on mobile data management*, p. 109.
- Onuma H. T., Kensuke, Faloutsos C. (2009). Tangent: a novel, “surprise me”, recommendation algorithm. , p. 657–666.
- Palmisano C., Tuzhilin A., Gorgoglione M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, n° 11, p. 1535–1549.
- Park Y. J., Tuzhilin A. (2008). The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 acm conference on recommender systems*, p. 11–18.
- Pu P., Zhou M., Castagnos S. (2009). Critiquing recommenders for public taste products. In *Recsys*, p. 249–252.
- Radlinski F., Bennett P. N., Carterette B., Joachims T. (2009, décembre). Redundancy, diversity and interdependent document relevance. *SIGIR Forum*, vol. 43, n° 2, p. 46–52.
- Said A., Kille B., Brijnesh J., Albayrak S. (2012). Increasing diversity through furthest neighbor-based recommendation. In *Proceedings of the workshop on diversity in document retrieval*. Seattle, USA.
- Schafer J., Konstan J., Riedl J. (2002). Meta-recommendations systems : user-controlled integration of diverse recommendations. In *International conference on information and knowledge management*, p. 43–51.
- Schickel-Zuber V., Faltings B. (2007). Using hierarchical clustering for learning theontologies used in recommendation systems. In *Proceedings of the 13th acm sigkdd international conference on knowledge discovery and data mining*, p. 599–608. New York, NY, USA, ACM.
- Schilit B., Theimer M. (1994). Disseminating active map information to mobile hosts. In *Ieee network*, p. 22–32. 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*, p. 1443–1450. Seattle, USA, ACM Press.
- Sieg A., Mabasher B., 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., McClave P. (2001). Similarity vs. diversity. In *Proceedings of the 4th international conference on case-based reasoning: Case-based reasoning research and development*, p. 347–361. London, UK, UK, Springer-Verlag.
- Sonam K., 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., Khoshgoftaar T. M. (2009, janvier). A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, vol. 2009, p. 4:2–4:2.
- thenextweb.com. (2012). *Remember netflix’s \$1m algorithm contest ? well, here’s why it didn’t use the winning entry*. <http://thenextweb.com/media/2012/04/13/remember-netflixs-1m-algorithm-contest-well-heres-why-it-didnt-use-the-winning-entry/>.

Webster N. (1980). Webster's new twentieth century dictionary of the english language.

Zhang M., Hurley N. (2008). Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 acm conference on recommender systems*, p. 123–130. New York, NY, USA, ACM.

Ziegler C.-N., McNeer S. M., Konstan J. A., Lausen G. (2005). Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on world wide web*, p. 22–32. New York, NY, USA, ACM.