

Louvain School of Management

Les systèmes de recommandation enferment-ils les utilisateurs, au cours du temps, dans leurs préférences dominantes ?

Auteur : Flore Vancompernelle Vromman
Promoteur : François Fouss
Année académique 2019-2020
Master [120] Ingénieur de gestion, à finalité spécialisée

Résumé

Les systèmes de recommandation occupent une place grandissante dans notre société. Ces algorithmes se basent, notamment, sur les goûts et attitudes préexistantes des utilisateurs pour leur proposer des recommandations. Mais finalement, *les systèmes de recommandation n'enfermeraient-ils pas les utilisateurs, au cours du temps, dans leurs préférences dominantes ?* C'est la question à laquelle nous essayons de répondre à travers ce mémoire.

Pour ce faire, nous nous penchons, tout d'abord, sur les avis et études des nombreux chercheurs dans le domaine. Certains, tels que Pariser (2011), affirment l'existence d'une bulle filtrante. Alors que d'autres (Hosanagar, Fleder, Lee & Buja, 2014) ont démontré que les systèmes de recommandation permettaient d'élargir le champ des possibles des utilisateurs.

Afin de confronter ces différents constats, nous avons choisi de réaliser notre propre étude. Alors, nous analysons et comparons plusieurs systèmes de recommandation sur l'évolution temporelle des mesures de diversité et de nouveauté obtenues au sein des recommandations fournies par ces algorithmes. Nous testons également plusieurs comportements de réponse potentiels des utilisateurs face aux recommandations. Toute cette phase d'expérimentations est réalisée grâce au langage de programmation Python.

En analysant les résultats, il n'est pas possible de confirmer ou d'infirmer la présence d'un enfermement des utilisateurs mais nous pouvons, toutefois, fournir une réponse nuancée à notre question de recherche. Un algorithme établissant des recommandations de manière aléatoire élargit le plus le champ des possibles pour les utilisateurs par rapport à d'autres algorithmes, à l'inverse d'un système de recommandation favorisant les items populaires qui a alors plus tendance à les enfermer. Nous constatons également que les algorithmes basés sur les similarités entre les utilisateurs ou les items proposent des diversités et nouveautés assez faibles au départ mais les augmentent au cours du temps jusqu'à atteindre un certain point de stagnation. Finalement, certains algorithmes étant conçus pour incorporer plus de diversité et de nouveauté ont effectivement tendance à moins enfermer les utilisateurs dès le départ par rapport à des algorithmes plus classiques même s'ils n'élargissent pas plus le champ des possibles pour les utilisateurs au cours du temps.

Remerciements

Lors de la réalisation de ce mémoire, j'ai pu compter sur l'aide de nombreuses personnes que j'aimerais à présent remercier.

Tout d'abord, je remercie mon promoteur, le professeur François Fouss, qui m'a accompagnée et conseillée tout au long de cette année académique en se rendant toujours disponible pour répondre à mes questions et en me prodiguant des critiques constructives me permettant de mener à bien ce travail.

De plus, j'aimerais remercier l'assistante Elora Fernandes pour son encadrement et ses précieux conseils.

Ensuite, je tiens particulièrement à remercier ma maman, Bernadette Vromman, ainsi que Madame Brigitte Bocquet pour le temps qu'elles ont consacré à la relecture de mon mémoire.

Finalement, j'adresse un grand merci à mes proches qui m'ont apporté un soutien sans faille au cours de mes études et spécialement lors de cette dernière année.

Table des matières

Liste des figures	VII
Liste des annexes.....	X
Introduction	1
Présentation de la question de recherche.....	1
Systèmes de recommandation	2
Bulles filtrantes	3
Chapitre 1 : Revue de littérature	5
Introduction	5
1. Systèmes de recommandation	6
1.1. Fonctionnement.....	6
1.2. Types d’algorithmes.....	7
1.3. Intérêts pour les entreprises.....	8
1.4. Intérêts pour les consommateurs.....	9
2. Mesurer les performances des systèmes de recommandation.....	10
2.1. Les métriques étudiant l’exactitude	10
2.2. Au-delà de l’exactitude	14
3. Conséquences des bulles filtrantes	16
3.1. D’où proviennent ces bulles filtrantes ?.....	16
3.2. Fragmentation et/ou homogénéisation des individus ?.....	17
3.3. Les dangers des bulles filtrantes	18
4. Solutions pour contrer les bulles filtrantes	22
4.1. Introduire la diversité dans les algorithmes	22
4.2. Impliquer les utilisateurs	23
5. Incorporation de la notion de temps	25
5.1. Cremonesi et Turrin (2010).....	25

5.2.	Nguyen et al. (2014)	26
5.3.	Lathia (2010).....	27
5.4.	Castagnos et al. (2014).....	28
5.5.	Nikolov et al. (2015)	29
5.6.	Szlavik et al. (2011)	30
5.7.	Conclusions de ces études.....	32
	Conclusion.....	33
	Chapitre 2 : méthodologie	34
	Introduction	34
1.	Outils utilisés.....	35
2.	Choix et analyse de la base de données.....	36
2.1.	Présentation et division de la base de données	36
2.2.	Conversion des données en une matrice	37
2.3.	Résumé du travail sur les données	39
3.	Étude de plusieurs algorithmes	40
3.1.	Algorithme de filtrage collaboratif item-based.....	41
3.2.	Algorithme de filtrage collaboratif user-based	44
3.3.	Algorithme de filtrage collaboratif intégrant une plus grande diversité au sein des recommandations	46
3.4.	Algorithme de filtrage collaboratif intégrant une plus grande nouveauté au sein des recommandations	49
3.5.	Algorithme établissant des recommandations aléatoires	51
3.6.	Algorithme établissant des recommandations populaires.....	51
3.7.	Algorithme de filtrage collaboratif avec factorisation matricielle.....	52
4.	Métriques pour évaluer les recommandations.....	56
4.1.	La diversité.....	57
4.2.	La nouveauté.....	59
5.	Simulation du comportement des utilisateurs	63

5.1. Composantes humaine et algorithmique & les utilisateurs acceptent toutes les recommandations	65
5.2. Composantes humaine et algorithmique & les utilisateurs acceptent les recommandations avec une certaine probabilité	66
5.3. Composante humaine & les utilisateurs n'acceptent aucune recommandation	67
5.4. Composante algorithmique & les utilisateurs acceptent toutes les recommandations	68
5.5. Composante algorithmique & les utilisateurs acceptent les recommandations avec une certaine probabilité	69
Conclusion.....	70
Chapitre 3 : Analyse des résultats	71
Introduction	71
1. Expérimentations et prise de décisions	72
2. Présentation des résultats obtenus	74
3. Influence du nombre de recommandations	76
4. Évolution temporelle	79
4.1. Analyse générale	79
4.2. Analyse par scénario	84
4.3. Conclusion	88
5. Étude de l'enfermement des utilisateurs dans leurs préférences dominantes	89
5.1. Algorithme établissant des recommandations aléatoires	90
5.2. Algorithme établissant des recommandations populaires	90
5.3. Algorithme de filtrage collaboratif item-based	91
5.4. Algorithme de filtrage collaboratif user-based	93
5.5. Algorithme de filtrage collaboratif introduisant une plus grande nouveauté au sein des recommandations	95
5.6. Algorithme de filtrage collaboratif introduisant une plus grande diversité au sein des recommandations	96

Conclusion des analyses.....	98
Conclusion.....	100
Bref résumé de l'étude	100
Limites et pistes de recherche	102
Bibliographie.....	104

Liste des figures

Figure 1.1. Nombre d'items pertinents/ impertinents, recommandés/ non-recommandés.....	12
Figure 2.1. Exemple de données	37
Figure 2.2. Représentations possibles des données dans une matrice.....	38
Figure 2.3. Exemple de matrice de données sur laquelle nos algorithmes se basent pour établir des recommandations.	40
Figure 2.4. Matrice de similarité cosinus entre les items	42
Figure 2.5. Liste des k=3 plus proches voisins pour chaque film	42
Figure 2.6. Établir les recommandations pour l'utilisateur 1 selon l'algorithme item-based ..	43
Figure 2.7. Matrice de similarité cosinus entre les utilisateurs	44
Figure 2.8. Liste des k=3 plus proches voisins pour chaque utilisateur.....	45
Figure 2.9. Établir les recommandations pour l'utilisateur 1 selon l'algorithme user-based...	46
Figure 2.10. Mesures de diversité calculées pour différents ensembles	48
Figure 2.11. Degrés de popularité et de nouveauté des items de la liste initiale.....	50
Figure 2.12. Nombre d'utilisateurs ayant vu chacun des films	51
Figure 2.13. Exemple de valeurs r_{ui} estimées pour l'utilisateur 1	54
Figure 2.14. Matrice de données de la première période temporelle	56
Figure 2.15. Matrice de données de la seconde période temporelle.....	57
Figure 2.16. Matrice de données regroupant les 2 périodes temporelles	57
Figure 2.17. Similarités cosinus entre les items	58
Figure 2.18. Diversité présente dans chacune des listes de recommandations	58
Figure 2.19. Degrés de popularité des différents films à la première période de temps.	61
Figure 2.20. Nouveauté présente dans chacune des listes de recommandations.....	61
Figure 2.21. Matrice de données de la première période de temps	64
Figure 2.22. Matrice de données de la seconde période de temps	64
Figure 2.23. Listes de recommandations fournies par notre algorithme à la première période de temps.	64
Figure 2.24. Matrice de données intégrant les caractéristiques du scénario 1	66
Figure 2.25. Matrice de données intégrant les caractéristiques du scénario 2	67
Figure 2.26. Matrice de données intégrant les caractéristiques du scénario 3	68
Figure 2.27. Matrice de données intégrant les caractéristiques du scénario 4	68
Figure 2.28. Matrice de données intégrant les caractéristiques du scénario 5	69

Figure 3.1. Évolution de la diversité, pour chaque scénario, obtenue par l’algorithme item-based lorsque 20 recommandations sont établies.....	74
Figure 3.2. Résultats de diversité et moyennes de ceux-ci pour l’algorithme item-based lorsque 20 recommandations sont établies.....	75
Figure 3.3. Moyenne générale des résultats de diversité obtenus par l’algorithme item-based lorsque 20 recommandations sont établies.....	76
Figure 3.4. Graphique reprenant les diversités moyennes générales obtenues pour chaque algorithme en ayant établi 10, 20 et 100 recommandations.....	77
Figure 3.5. Graphique reprenant les nouveautés moyennes générales obtenues pour chaque algorithme en ayant établi 10, 20 et 100 recommandations.....	77
Figure 3.6. Moyennes, pour chaque période de temps, des mesures de diversité obtenues par l’algorithme item-based lorsque 20 recommandations sont établies.....	79
Figure 3.7. Évolution de la diversité dans le temps pour chaque algorithme lorsque 20 recommandations sont établies + Diversité globale.....	80
Figure 3.8. Calcul de l’évolution de la diversité pour l’algorithme item-based lorsque 20 recommandations sont établies.....	81
Figure 3.9. Évolution de la diversité globale.....	81
Figure 3.10. Évolution de la nouveauté dans le temps pour chaque algorithme lorsque 20 recommandations sont établies + Nouveauté globale.....	83
Figure 3.11. Évolution de la nouveauté globale.....	83
Figure 3.12. Évolution de la diversité, pour chaque scénario, obtenue par l’algorithme introduisant plus de diversité lorsque 20 recommandations sont établies.....	86
Figure 3.13. Évolution de la diversité, pour chaque scénario, obtenue par l’algorithme item-based lorsque 10 recommandations sont établies.....	87
Figure 3.14. Éléments en faveur d’un élargissement du champ des possibles pour les utilisateurs avec l’algorithme aléatoire.....	90
Figure 3.15. Éléments en faveur d’un élargissement du champ des possibles pour les utilisateurs avec l’algorithme populaire.....	91
Figure 3.16. t-tests entre les mesures globales et les résultats de diversité/nouveauté obtenus en moyenne à chaque période par l’algorithme item-based.....	92
Figure 3.17. Éléments en faveur d’un élargissement du champ des possibles pour les utilisateurs avec l’algorithme item-based.....	93
Figure 3.18. t-tests entre les mesures globales et les résultats de diversité/nouveauté obtenus en moyenne à chaque période par l’algorithme user-based.....	93

Figure 3.19. t-tests entre les résultats obtenus par les algorithmes item-based et user-based..	94
Figure 3.20. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme user-based.....	94
Figure 3.21. t-tests entre les résultats obtenus par l'algorithme introduisant plus de nouveauté et celui introduisant plus de diversité.....	95
Figure 3.22. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme introduisant plus de nouveauté.....	96
Figure 3.23. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme introduisant plus de diversité.....	97
Figure 3.24. Classement des algorithmes quant à l'enfermement des utilisateurs qu'ils provoquent.....	98

Liste des annexes

Annexe 1 : Expérimentations

Annexe 2 : Code Python

Annexe 3 : Tableaux et graphiques de résultats

Introduction

Présentation de la question de recherche

À travers ce mémoire, nous avons choisi d'étudier la problématique suivante : « Les systèmes de recommandation enferment-ils les utilisateurs, au cours du temps, dans leurs préférences dominantes ? ».

Il s'agit d'un sujet plus que jamais d'actualité au vu de l'utilisation d'internet toujours croissante et de la quantité astronomique de données à laquelle nous faisons face (Alani, Tawfik, Saeed & Anya, 2018 ; Skourletopoulos, Mastorakis, Mavromoustakis, Dobre & Pallis, 2018). Les systèmes de recommandation permettent alors aux utilisateurs de faire un tri facilement parmi cette multitude d'informations, en se basant notamment sur leurs goûts (Helberger, Karppinen & D'acunto, 2018 ; Pariser, 2011 ; Zanker, Rook & Jannach, 2019).

Mais à force d'établir des recommandations pour les utilisateurs, ces systèmes ne les enfermeraient-ils pas dans leurs préférences dominantes ? C'est une question sur laquelle de nombreux chercheurs se sont penchés et que nous allons également essayer d'approfondir. Dans cet objectif, nous aurons l'occasion d'étudier le point de vue de ces chercheurs quant à l'enfermement potentiel des utilisateurs ainsi que les conséquences et avantages que cela peut avoir à la fois pour les consommateurs eux-mêmes, mais aussi pour les entreprises qui mettent en place ce genre de systèmes. De plus, nous présenterons différentes études qui ont été réalisées dans ce domaine.

À la suite de cette revue de la littérature, nous serons alors en mesure de réaliser notre propre étude temporelle concernant l'enfermement potentiel des utilisateurs par les systèmes de recommandation. Pour ce faire, nous présenterons tout d'abord la méthodologie que nous comptons appliquer pour ensuite passer à la phase expérimentale où nous aurons l'occasion de coder en Python différents systèmes de recommandation. Finalement, nous réaliserons une analyse des résultats obtenus afin de répondre, au mieux, à notre question de recherche.

Mais avant d'entamer tout cela, nous souhaitons définir certains éléments de notre question de recherche, à savoir les systèmes de recommandation, ainsi que l'enfermement des utilisateurs dans leurs préférences dominantes que nous caractériserons par la notion de bulle filtrante.

Systemes de recommandation

De nos jours, de plus en plus d'entreprises ont recours aux systemes de recommandation. Ceux-ci sont consideres par les consommateurs comme les sources de recommandation les plus influentes, devançant même celles faites par les pairs et les experts. Pourtant, ils les perçoivent aussi comme ayant moins d'expertise et étant moins dignes de confiance (Senecal & Nantel, 2004).

En 2013, une étude réalisée par l'entreprise McKinsey demontrait que 35% de ce qu'achetaient les consommateurs sur Amazon provenait des recommandations faites par les algorithmes de ce géant du web (MacKenzie, Meyer & Noble, 2013). Selon Pariser (2011), Amazon a réussi à vendre des billions de dollars en prédisant ce par quoi les utilisateurs étaient intéressés et en mettant en avant des produits spécifiques pour chacun des consommateurs. En ce qui concerne Netflix, les chiffres sont encore plus impressionnants : 75% de ce que regardent les utilisateurs du site résulte de recommandations proposées par celui-ci (Netflix Technology Blog, 2012). Mais les systemes de recommandation ne s'arrêtent pas à Amazon et Netflix, ils sont présents dans une multitude de domaines différents. Par exemple, dans le secteur de la musique avec notamment Spotify. N'oublions pas non plus les services d'informations. Selon une étude réalisée en 2007 (Das, Datar, Garg & Rajarm, 2007, cités dans Hosanagar et al., 2014), les recommandations établies par Google news permettaient d'augmenter le nombre d'articles vus de 38%. Les réseaux sociaux n'échappent pas non plus aux systemes de recommandation, ceux-ci y sont même très présents. De nombreuses études ont été réalisées concernant les recommandations établies sur les réseaux sociaux tels que YouTube, Facebook, Twitter, etc. (Abisheva, Garcia & Schweitzer, 2016 ; Nikolov, Oliveira, Flammini & Menczer, 2015 ; Sumpter, 2018)

Plusieurs façons de décrire un système de recommandation existent. Dès lors, nous allons approfondir cette notion à travers différentes définitions. Tourwé (2012) expose un système de recommandation comme étant un système de filtrage qui prédit la préférence attribuée par un utilisateur à un item (i.e. produit, service, élément qui peut être recommandé) qu'il n'avait pas encore considéré. Dans ce cas, il ne s'agit donc pas de recommander à l'utilisateur un produit qu'il avait de toute façon prévu de consommer car son achat ne serait alors pas dû au système de recommandation. Selon d'autres auteurs, les systemes de recommandation « ont pour but de faciliter la recherche et l'accès à l'information en proposant des items adaptés aux besoins et préférences des utilisateurs. » (Castagnos, Brun & Boyer, 2014, p.3). Les systemes de

recommandation placent l'utilisateur dans un endroit confortable entouré de ses idées, produits et personnes préférées (Pariser, 2011). Helberger et al. (2018) partagent le même avis sur le fait que la liste des informations proposées à un utilisateur est personnalisée en fonction de ses goûts. Ils ajoutent également que les informations présentées dépendent des préférences personnelles de l'utilisateur, de ses choix précédents et des préférences de ses amis. De plus, ils indiquent que des décisions stratégiques et commerciales interviennent aussi dans les algorithmes de recommandation mis en place par les entreprises.

Bulles filtrantes

Comme nous venons d'en parler, les systèmes de recommandation proposent des items en fonction des préférences des utilisateurs. Sumpter (2018, p 133) nous dit que les recherches sur Facebook et Google sont faites pour nous montrer seulement ce que nous voulons voir. Mais, il se peut alors que les utilisateurs se retrouvent enfermés dans leurs préférences dominantes. Pariser (2011) est le premier à mettre un nom sur cet environnement unique qui entoure l'utilisateur, il parle alors de « filter bubble » que nous traduirons par *bulle filtrante*. Il caractérise cette bulle par trois dynamiques. Premièrement, chaque utilisateur y est seul et est donc séparé des autres. Deuxièmement, l'utilisateur ne sachant pas quelles sont les caractéristiques utilisées pour trier les informations, Pariser décrit la bulle filtrante comme invisible ainsi que les critères sur lesquels se basent les algorithmes de recommandation. Troisièmement, il expose le fait que les utilisateurs ne choisissent pas d'entrer dans cette bulle (Pariser, 2011, pp. 9-10).

Geschke, Lorenz et Holz (2019), quant à eux, définissent les bulles filtrantes comme le résultat de différents processus de recherche de l'information, de perception et de sélection. Ils pensent que les utilisateurs ne reçoivent qu'une infime partie des informations disponibles en fonction de leurs attitudes préexistantes.

Sumpter (2018) propose un exemple sur la manière dont se forment les bulles filtrantes. Sur Google actualités, au plus un utilisateur clique sur des articles liés à un certain sujet, au plus des articles liés à ce sujet lui seront proposés. Comme ce sujet l'intéresse, il va probablement consulter les articles recommandés donc d'autant plus lui en seront suggérés sur ce sujet, et ainsi de suite. L'utilisateur se retrouvera alors de plus en plus enfermé dans une bulle filtrante.

Il existe un autre terme que les bulles filtrantes utilisé dans la littérature, il s'agit des « echo chambers » que nous traduirons simplement par *chambres d'écho*. Sumpter (2018, p 137)

distingue ces deux notions. Selon lui, les bulles filtrantes sont créées par les algorithmes de filtration alors que les chambres d'écho sont créées par les individus eux-mêmes. En effet, Geschke et al. (2019) sont du même avis. Ils indiquent que les chambres d'écho apparaissent suite aux mécanismes cognitifs, sans intervention nécessaire des systèmes de filtres technologiques. Mais, avec la présence de ces derniers, les auteurs renseignent que les chambres d'écho sont renforcées, c'est-à-dire qu'elles sont encore moins connectées les unes aux autres. Selon eux (Geschke et al., 2019), les chambres d'écho sont en réalité un phénomène social, et non pas technologique, au sein duquel les bulles filtrantes de plusieurs individus se recourent fortement.

Sunstein (2007) met en avant le danger lié à l'apparition des chambres d'écho. Il pense que c'est une conséquence sociétale indésirable car les individus sont tellement ancrés dans leurs opinions qu'il devient difficile d'obtenir un consensus au sein de la société et de mettre en relation des points de vue différents.

Chapitre 1 : Revue de littérature

Introduction

Notre problématique étant définie, nous allons à présent mettre en évidence différents points à ce propos à partir de la littérature. Tout d'abord, nous développerons plus en profondeur les systèmes de recommandation en abordant leur fonctionnement, les différents types d'algorithmes existants ainsi que les intérêts rencontrés pour les entreprises et les consommateurs. Ensuite, nous aborderons comment mesurer les performances des systèmes de recommandation et nous introduirons alors les notions de diversité et de nouveauté. Après cela, nous mettrons en évidence plusieurs conséquences des bulles filtrantes. À la suite de quoi, nous présenterons diverses solutions établies par plusieurs chercheurs afin d'éviter l'apparition de bulles filtrantes. Finalement, nous analyserons quelques études qui ont permis d'explorer la notion de temps à travers les systèmes de recommandation.

1. Systèmes de recommandation

Nous avons eu l'occasion d'aborder brièvement les systèmes de recommandation lors du chapitre introductif de ce travail. Nous allons maintenant approfondir ce sujet premièrement en développant le fonctionnement des systèmes de recommandation. Ensuite, nous verrons les différents types de filtrage qui se cachent derrière les algorithmes. Finalement, nous déterminerons les raisons pour lesquelles les entreprises ont recours à des systèmes de recommandation ainsi que les intérêts pour les utilisateurs.

1.1. Fonctionnement

Différents auteurs ont identifié le fonctionnement des systèmes de recommandation en 3 étapes (Castagnos et al., 2014 ; Pariser, 2011). La première consiste à collecter des informations sur les utilisateurs, celles-ci peuvent être implicites ou explicites (Castagnos et al., 2014). En général, les utilisateurs offrent aux entreprises une quantité importante d'informations les concernant en échange des services proposés (Pariser, 2011). Les données explicites sont celles transmises consciemment et volontairement par l'utilisateur. Par exemple, les notes attribuées aux produits et services, les réponses à des enquêtes, etc. Malheureusement, il se peut que ces données transmises consciemment par l'utilisateur soient biaisées notamment par les émotions extérieures qu'ils ressentent au moment de l'évaluation (Abisheva et al., 2016). Tandis que les données implicites sont collectées par l'entreprise sans que l'utilisateur n'en soit réellement conscient. Cela correspond aux interactions des utilisateurs, il s'agit par exemple des clics, de l'historique de recherche, de la durée passée sur un site, des achats réalisés, etc. (Zanker et al., 2019)

La seconde étape représente le réel système de recommandation. Il s'agit d'analyser et de transformer les données de manière à pouvoir les exploiter (Castagnos et al., 2014). Une fois les informations récoltées, nous disposons d'une base de données dans laquelle on retrouve généralement un ensemble de notes attribuées par des utilisateurs à des items. Ce que nous appelons ici « items » correspond à des produits ou services que l'utilisateur va pouvoir évaluer tels qu'un film, une musique, un bien qu'il a pu acheter, etc. Cette base de données peut alors être divisée en deux sous-groupes : un test set et un training set. L'ensemble des éléments repris dans le training set va permettre d'établir le modèle et ceux appartenant au test set serviront à l'évaluer. Généralement, on retrouve au moins 50% des données dans le training set et le reste

dans le test set. (Lathia, 2010 ; Zanker et al., 2019). Les différents types d’algorithmes exploitant ces données seront décrits dans la section suivante.

Finalement, la dernière étape permet de fournir des recommandations aux utilisateurs ainsi que de faire des prédictions concernant leurs évaluations à propos de différents items (Castagnos et al., 2014). Le but étant de fournir aux utilisateurs l’expérience la plus intéressante (Zanker et al., 2019).

1.2. Types d’algorithmes

Comme nous l’avons expliqué dans la section précédente, il existe différentes manières d’exploiter les données afin de proposer des recommandations. Les principaux types, que nous détaillerons ci-dessous, sont le filtrage collaboratif et le filtrage basé sur le contenu. Bien sûr, d’autres approches peuvent également être utilisées telles que des méthodes hybrides, basées à la fois sur le contenu et le filtrage collaboratif. Les recommandations peuvent aussi être faites sur base de la popularité en suggérant uniquement les items les plus populaires auprès de l’ensemble des utilisateurs. (Castagnos et al., 2014)

Le premier type de système de recommandation que nous allons aborder est le filtrage basé sur le contenu. Celui-ci se réfère aux caractéristiques des items consultés et/ou appréciés précédemment par l’utilisateur. Le modèle peut ensuite recommander à l’utilisateur d’autres items en fonction de leurs caractéristiques (Castagnos et al., 2014). Prenons l’exemple où les items représentent des films dont les caractéristiques correspondent au genre, à la durée, etc. Si un utilisateur apprécie les comédies romantiques de moins de 2h, le système de recommandation basé sur le contenu lui recommandera alors d’autres comédies romantiques dont la durée est de moins de 2h.

En ce qui concerne le filtrage collaboratif, de très nombreux algorithmes ont été développés dans ce domaine. On part du présupposé que des utilisateurs qui avaient des goûts similaires dans le passé auront des goûts similaires dans le futur (Lathia, 2010). Plusieurs algorithmes se basent sur la technique des plus proches voisins, ou *k-nearest neighbors* en anglais. Cela se fonde dans l’utilisation des relations entre les utilisateurs ou les items. En ce qui concerne l’approche orientée « items », il est nécessaire de calculer la similarité entre toutes paires d’items. Ensuite, l’algorithme pourra prédire l’évaluation d’un utilisateur pour un item en analysant les items voisins de celui considéré. Lorsqu’on parle d’éléments voisins, cela revient à dire que ceux-ci sont similaires, la similarité entre ces deux items est élevée. On suppose alors

que les scores d'items similaires tendent à être identiques lorsqu'ils sont évalués par le même utilisateur (Lathia, 2010). L'approche orientée « utilisateurs » nécessite, quant à elle, de calculer la similarité entre toutes paires d'utilisateurs. Ensuite, il est possible de prédire les préférences d'un utilisateur envers un item en regardant les notes attribuées à cet item par les voisins de l'utilisateur considéré. À nouveau, des utilisateurs voisins ont une similarité élevée. On suppose alors qu'ils tendent à évaluer un item dans la même direction (Lathia, 2010). Que ce soit pour les approches orientées items ou utilisateurs, il est nécessaire d'estimer la similarité entre toutes paires d'items/utilisateurs. Celle-ci peut être calculée grâce à différentes mesures : corrélation de Pearson, similarité cosinus, etc. (Koren, 2010 ; Lathia, 2010). Prenons à nouveau un exemple pour que cela soit plus clair. Imaginons cette fois que deux utilisateurs apprécient de la même manière les mêmes films. Si l'un de ces deux utilisateurs a vu un film et l'a aimé, on va recommander ce film à l'autre utilisateur étant donné qu'ils ont les mêmes goûts, en supposant que ce dernier ne l'a pas encore vu.

Toujours dans le filtrage collaboratif, une autre méthode que celle des plus proches voisins a été développée, il s'agit de la factorisation matricielle. Celle-ci permet d'estimer les notes attribuées par les utilisateurs en caractérisant à la fois les utilisateurs et les items. Pour cela, chaque utilisateur est représenté par une série de facteurs latents et chaque item est également représenté par une série d'autres facteurs latents. Ces facteurs sont obtenus grâce à un processus itératif qui apprend sur base des éléments connus, à savoir ceux du training set. Ensuite, il est possible d'estimer la préférence d'un utilisateur pour un certain item grâce aux facteurs latents caractérisant cet utilisateur et ceux caractérisant cet item. (Koren, 2010 ; Nguyen & Zhu, 2012 ; Paterek, 2007 ; Yu, Wang & Gao, 2014)

1.3. Intérêts pour les entreprises

Les systèmes de recommandation peuvent avoir un impact important sur le chiffre d'affaire de certaines entreprises. Différents avantages pour celles-ci ont déjà été démontrés tels que le fait de générer des ventes additionnelles ainsi que d'augmenter la loyauté des consommateurs envers l'entreprise en améliorant leur expérience (Postma & Brokke, 2002, cités dans Senecal & Nantel, 2004 ; Schafer, Konstan & Riedl, 1999, cités dans Fleder & Hosanagar, 2008 ; Srinivasan, Anderson & Ponnawolu, 2002 cités dans Senecal & Nantel, 2004). À travers les systèmes de recommandation, les utilisateurs développent une certaine confiance dans l'entreprise (Chen & Pu, 2008). Des études montrent que la confiance des utilisateurs est

positivement corrélée à leurs intentions d'achat ou de retourner sur le site internet de la société (Grabner-Kräuter & Kaluscha, 2003 cités dans Chen & Pu, 2008).

Pour mesurer l'impact réel des systèmes de recommandation, les entreprises peuvent utiliser différents taux comme par exemple le taux de clics. Ce dernier mesure la proportion d'éléments présentés et qui ont reçu un clic, il est régulièrement utilisé pour les publicités en ligne. L'inconvénient est qu'il se peut que les utilisateurs aient cliqué accidentellement sur l'élément présenté ou encore qu'ils ne soient pas satisfaits du lien sur lequel ils ont cliqué (Zanker et al., 2019). Les taux d'adoption et de conversion sont une autre manière de voir l'impact des recommandations pour l'entreprise. Ils dépendent du domaine dans lequel la société est active. L'adoption est considérée comme l'utilisation d'un service ou d'une fonctionnalité alors que la conversion a une signification commerciale, il s'agit par exemple d'ajouter un article au panier sur un site de commerce en ligne. (Zanker et al., 2019)

1.4. Intérêts pour les consommateurs

Bien sûr, les consommateurs perçoivent aussi les bénéfices des systèmes de recommandation. À notre époque, nous faisons face à une masse d'informations et de données beaucoup trop importante pour que les utilisateurs puissent tout gérer et analyser eux-mêmes. La filtration réalisée par les systèmes de recommandation permet donc aux consommateurs de faire un tri parmi cette multitude d'informations (Helberger et al., 2018 ; Pariser, 2011 ; Zanker et al., 2019). Les systèmes de recommandation minimisent alors le coût d'accès à l'information pour les utilisateurs en leur offrant un gain de temps tout en leur proposant l'information qu'ils cherchaient (Senecal & Nantel, 2004 ; Zanker et al., 2019). Pham et Healey (2005, cités dans Fleder & Hosanagar, 2008) ajoutent également que les systèmes de recommandation permettent de faire découvrir aux utilisateurs des éléments qu'ils ne connaissaient pas avant.

De plus, les utilisateurs perçoivent le sentiment d'être uniques à travers cette relation personnalisée (Wind & Rangaswamy, 2001, cités dans Senecal & Nantel, 2004). En outre, Nguyen, Hui, Harper, Terveen et Konstan (2014) ont effectué une expérience au terme de laquelle ils ont réalisé que les utilisateurs qui suivaient les recommandations proposées percevaient une meilleure expérience que ceux qui ne les suivaient pas. Cette étude avait été réalisée sur une base de données de films, il n'est donc pas garanti que les mêmes résultats s'appliqueraient pour tous les systèmes de recommandation existants.

2. Mesurer les performances des systèmes de recommandation

Il existe de nombreuses métriques qui permettent de mesurer les performances des systèmes de recommandation. Le terme « performance » varie forcément en fonction de l'objectif visé par l'algorithme : proposer des produits similaires à ceux appréciés par l'utilisateur précédemment, ou des éléments nouveaux afin de le surprendre, ou encore un contenu plus diversifié. Dans ces divers cas, des métriques différentes permettent de comparer les algorithmes et nous allons les détailler dans cette section.

2.1. Les métriques étudiant l'exactitude

Pendant de nombreuses années, ce fut les métriques d'exactitude qui étaient utilisées pour évaluer les performances des systèmes de recommandation. Nous avons choisi de traduire le terme anglais « accuracy », souvent employé dans la littérature, par le mot « exactitude » en français. Nous allons tout d'abord présenter différentes métriques permettant d'évaluer l'exactitude des prédictions. Ensuite, nous exposerons d'autres métriques ayant pour objectif de mesurer l'exactitude des recommandations.

Mais avant cela, rappelons les notions de training set et test set utiles pour ces métriques. Nous avons décrit précédemment que les algorithmes utilisés dans les systèmes de recommandation apprennent sur base du training set. Grâce à cela, ils sont alors capables de fournir des recommandations et/ou des prédictions. Ensuite, il est nécessaire d'évaluer les résultats fournis par les différents modèles afin d'apprécier la qualité du système de recommandation. Pour cela, des métriques sont utilisées pour comparer les résultats obtenus avec les éléments du test set et déterminer quel modèle est le plus adéquat.

Au niveau des prédictions

Les métriques évaluant l'exactitude des prédictions permettent de mesurer le taux d'erreurs à minimiser. Pour rappel, une prédiction correspond au score, estimé par l'algorithme, qu'un utilisateur attribuerait à un certain item. Afin de déterminer le taux d'erreur, il est nécessaire de comparer les prédictions, obtenues sur base des données du training set, avec les notes réellement rencontrées dans le test set. Plus la différence entre les valeurs prédites et les valeurs réellement rencontrées est faible, plus cela signifie que l'exactitude des prédictions est élevée, le but étant d'avoir des algorithmes les plus précis possibles. (Lathia, 2010)

Les principales métriques qui permettent de mesurer l'exactitude des prédictions établies par un algorithme sont le MAE (Mean Absolute Error) et le RMSE (Root Mean Squared Error). Grâce au MAE, il est possible de connaître l'erreur moyenne réalisée en considérant qu'une erreur correspond à la valeur absolue de la différence entre le score prédit et le score réellement rencontré. Ci-dessous se trouve la formule pour obtenir le MAE (1) (Herlocker, Konstan, Terveen & Riedl, 2004 ; Ziegler, Mcnee, Konstan & Lausen, 2005)

$$MAE = \frac{\sum_{p_{u,i} \in R} |p_{u,i} - r_{u,i}|}{|R|} \quad (1)$$

Où $p_{u,i}$ représente le score prédit pour l'utilisateur u à l'item i ;

$r_{u,i}$ correspond au score que l'utilisateur u a réellement attribué à l'item i ;

R désigne l'ensemble des scores prédits ;

$|R|$ représente le nombre d'éléments composant R .

La formule du RMSE (2) est assez similaire hormis qu'un poids plus important est accordé aux erreurs plus grandes car la différence entre le score prédit et le score réel est mise au carré et non plus en valeur absolue. Il faut ensuite prendre la racine carrée du tout pour obtenir le RMSE. (Koren, 2010 ; Lathia, 2010 ; Szlavik, Kowalczyk & Schut, 2011)

$$RMSE = \sqrt{\frac{\sum_{p_{u,i} \in R} (p_{u,i} - r_{u,i})^2}{|R|}} \quad (2)$$

Où $p_{u,i}$ représente le score prédit pour l'utilisateur u à l'item i ;

$r_{u,i}$ correspond au score que l'utilisateur u a réellement attribué à l'item i ;

R désigne l'ensemble des scores prédits ;

$|R|$ représente le nombre d'éléments composant R .

Bien sûr, il existe d'autres métriques permettant de mesurer l'exactitude des prédictions mais le MAE et le RMSE sont les deux plus répandues. En 2006, Netflix lança un défi qui consistait à trouver un système de recommandation plus performant en termes d'exactitude des résultats. Il fallait alors élaborer un algorithme qui permettait de réduire au maximum le RMSE, l'objectif étant d'atteindre une réduction de 10% du RMSE de l'algorithme utilisé à ce moment-là par Netflix. S'en est alors suivi un concours à travers le monde entier au terme duquel le grand

gagnant s'est vu offrir la somme d'un million de dollars pour avoir trouvé un algorithme offrant un RMSE de 0,8567. (Hallinan & Striplas, 2016 ; Netflix Technology Blog, 2012)

Au niveau des listes de recommandations

Il existe également différentes métriques permettant d'étudier l'exactitude des recommandations proposées par un algorithme. Tout comme pour les prédictions, les recommandations sont établies par l'algorithme sur base des données du training set. Ensuite, il est possible d'évaluer l'exactitude de ces recommandations grâce aux données du test set. Le but étant que ces recommandations correspondent au mieux à ce que l'utilisateur apprécie réellement.

Les deux métriques les plus utilisées dans la littérature sont celles de « Precision » et de « Recall » (Han & Yamana, 2017 ; Herlocker et al., 2004 ; Sarwar, Karypis, Konstan & Riedl, 2000). Elles permettent de déterminer si les recommandations suggérées à l'utilisateur lui conviennent ou pas. Afin de calculer ces métriques, il est nécessaire de connaître les éléments de la Figure 1.1. Un item est considéré comme pertinent pour un utilisateur lorsqu'il a été jugé positivement par cet utilisateur dans le test set. Le fait qu'un item soit jugé positivement varie forcément d'une base de données à l'autre ainsi que des hypothèses fixées par les chercheurs. Par exemple, s'il s'agit d'une base de données contenant les notes (variant de 1 à 5) attribuées par des utilisateurs à des films, on pourrait éventuellement considérer qu'un item est pertinent pour un utilisateur lorsqu'une note de 4 ou 5 lui a été attribuée (Herlocker et al., 2004).

	Recommandé	Non-recommandé	Total
Pertinent	N_{pr}	N_{pn}	N_p
Impertinent	N_{ir}	N_{in}	N_i
Total	N_r	N_n	N

Figure 1.1. Nombre d'items pertinents/ impertinents, recommandés/ non-recommandés

Afin d'obtenir la métrique de Precision, la formule suivante (3) peut être réalisée. Cette métrique permet de représenter la probabilité qu'un item recommandé à l'utilisateur soit pertinent pour lui. (Herlocker et al., 2004 ; Sarwar et al., 2000)

$$Precision = \frac{N_{pr}}{N_r} \quad (3)$$

Où N_{pr} correspond au nombre d'items pertinents pour l'utilisateur et qui lui étaient recommandés par l'algorithme ;

N_r représente le nombre d'items qui étaient recommandés à l'utilisateur par l'algorithme.

En ce qui concerne la métrique de Recall, elle s'obtient par la formule suivante (4). Cette métrique représente la probabilité qu'un item pertinent pour l'utilisateur lui ait été recommandé. (Herlocker et al., 2004 ; Sarwar et al., 2000)

$$Recall = \frac{N_{pr}}{N_p} \quad (4)$$

Où N_{pr} correspond au nombre d'items pertinents pour l'utilisateur et qui lui étaient recommandés par l'algorithme ;

N_p représente le nombre d'items pertinents pour l'utilisateur.

Ces métriques de Precision et de Recall peuvent parfois être conflictuelles. En effet, augmenter le nombre d'items recommandés (N_r) aurait tendance à accroître le Recall alors que cela aurait l'effet inverse sur la métrique de Precision.

Dans le calcul de ces deux métriques, l'ordre dans lequel les recommandations sont proposées à l'utilisateur n'a pas d'importance. On considère que l'utilisateur consulte l'ensemble des recommandations qui lui sont suggérées (Han & Yamana, 2017). D'autres métriques existantes accordent une valeur plus importante aux items pertinents placés dans le haut de la liste de recommandations. Dans ce domaine, les métriques les plus connues sont « Mean Reciprocal Rank (MRR) », « Mean Average Precision (MAP) » et « Normalized Discounted Cumulative Gain (NDCG) ». (Goel, Kumar, Kumar & Chopra, 2019 ; Han & Yamana, 2017). À titre d'exemple, nous allons présenter le MRR dont la formule se trouve ci-dessous (5) (Goel et al., 2019).

$$MRR = \frac{1}{|U|} \sum_{u \in U} \frac{1}{k_u} \quad (5)$$

Où U correspond à l'ensemble des utilisateurs ;

$|U|$ représente donc le nombre d'utilisateurs ;

k_u correspond au rang de la première recommandation pertinente pour l'utilisateur u .

Prenons un exemple pour clarifier cette dernière notion. Imaginons que les items correspondent à des films et qu'une liste de 10 films est recommandée à un utilisateur. Suite aux recommandations établies pour l'utilisateur, supposons que ce dernier n'ait pas évalué le premier film de la liste mais qu'il ait attribué une note de 4/5 au second film de la liste des films qui lui étaient recommandés. Ce second film est alors considéré comme le premier item pertinent pour l'utilisateur. Dès lors, k_u , le rang de la première recommandation pertinente pour cet utilisateur u , est 2. Pour connaître le MRR, il faut calculer le rang pour chaque utilisateur et appliquer ensuite la formule présentée (5).

2.2. Au-delà de l'exactitude

Les métriques permettant de maximiser l'exactitude des résultats ont pour objectif d'avoir un système de recommandation plus optimal. Mais, les facteurs humains sont alors mis de côté (Castagnos et al., 2014). De plus, il se peut que certains algorithmes, se concentrant uniquement sur l'exactitude, enferment petit à petit les consommateurs dans leurs préférences dominantes. Ceci fera partie de notre recherche. Dès lors, il devient intéressant d'étudier d'autres facteurs qui permettent de mesurer la performance des systèmes de recommandation. Nous allons principalement nous concentrer sur la diversité et la nouveauté.

Smyth et McClave (cités dans Castagnos et al., 2014, p4) ont défini la diversité « comme étant opposée à la similarité dans les systèmes de recommandation ». Castagnos et al. (2014, p4) sont du même avis car ils indiquent qu'il s'agit d'une « mesure quantifiant la dissimilarité dans un ensemble d'items ».

En ce qui concerne la nouveauté, nous pouvons considérer qu'un item est nouveau pour un utilisateur lorsque ce dernier n'avait pas connaissance de cet item avant qu'il ne lui soit recommandé (Herlocker et al., 2004 ; Kaminskas & Bridge, 2016).

Différents avantages de la diversité et de la nouveauté dans les recommandations faites aux utilisateurs ont pu être identifiés. Selon Helberger et al. (2018), promouvoir la diversité dans

les systèmes de recommandation permettrait de contrer les bulles filtrantes. De plus, la diversité et la nouveauté seraient positivement corrélées à la satisfaction des utilisateurs (Castagnos et al., 2014 ; Han & Yamana, 2017) pour autant qu'elles soient utilisées avec parcimonie afin de ne pas provoquer l'incompréhension des utilisateurs. Castagnos et al. (2014) ajoutent également que la diversité favorise l'atteinte d'un bon compromis concernant les recommandations faites aux utilisateurs car les items recommandés correspondent à leurs préférences tout en intégrant des nouveautés et en évitant que les items soient trop proches les uns des autres (Castagnos et al., 2014, p4). En outre, Helberger et al. (2018) suggèrent que proposer un contenu diversifié favorise le développement des goûts des utilisateurs ainsi que leur épanouissement personnel. D'autres auteurs (Nguyen et al., 2014) partagent également ce point de vue. Selon eux, il est intéressant de proposer des items diversifiés aux utilisateurs afin de leur faire découvrir de nouveaux éléments dont ils ne sont pas conscients, liant ainsi la diversité et la nouveauté. Ziegler et al. (2005) identifient un avantage supplémentaire en indiquant qu'un utilisateur peut avoir plusieurs intérêts dont seulement certains sont ciblés par les recommandations établies. Dès lors, intégrer la diversité ouvrirait des portes vers d'autres éléments qui peuvent en réalité être également des points d'intérêts de l'utilisateur.

Malheureusement, plusieurs auteurs ont observé que les algorithmes favorisant la diversité allaient généralement à l'encontre de l'exactitude et inversement. Plus la diversité exigée est élevée, plus l'exactitude diminue. (Zhou et al., 2010 ; Ziegler et al., 2005)

Dans ce travail nous nous concentrerons majoritairement sur la diversité et la nouveauté comme façons supplémentaires d'évaluer des systèmes de recommandation. Mais bien sûr, il existe de nombreuses autres métriques que nous n'aborderons que brièvement. Par exemple, Nguyen et al. (2014) ont comparé les résultats d'un système de recommandation sur deux groupes à travers l'expérience utilisateur. Celle-ci était obtenue grâce aux notes attribuées par les utilisateurs aux différents items. La sérendipité est également une métrique qui peut être utilisée. Il est difficile de lui attribuer une définition exacte. Ziegler et al. (2005) précisent que la sérendipité est le fait de recommander quelque chose de non évident à l'utilisateur, auquel il ne s'attendait pas. D'autres auteurs (Bethke, 2016 ; Han & Yamana, 2017) ajoutent qu'en plus du degré de surprise de l'utilisateur, la sérendipité inclut également une certaine pertinence pour l'utilisateur. Il ne s'agit pas de lui proposer des items auxquels il ne s'attend pas mais qui ne l'intéressent pas non plus. La sérendipité est donc liée à la diversité et à la nouveauté aussi.

3. Conséquences des bulles filtrantes

Pariser (2011) est le premier à parler de bulle filtrante dans laquelle les utilisateurs se retrouvent enfermés, ils sont alors cloisonnés dans cet environnement introduit par leurs préférences dominantes. Mais finalement, on peut se demander quelles sont les conséquences de ces bulles filtrantes pour les utilisateurs et la société de manière plus générale. Différents auteurs ont des avis bien spécifiques sur cette question, nous aborderons ceci dans cette section. Avant cela, nous nous demanderons si l'apparition de cette bulle est due uniquement aux systèmes de recommandation ou si d'autres éléments entrent en jeu.

3.1. D'où proviennent ces bulles filtrantes ?

Selon certains auteurs, l'apparition des bulles filtrantes n'est pas seulement liée aux technologies et aux systèmes de recommandation. Geschke et al. (2019) identifient 3 niveaux permettant l'apparition des bulles filtrantes : individuel, social et technologique. Le niveau individuel consiste à reproduire par habitude nos attitudes préexistantes, il s'agit du biais de confirmation (Jonas, Schulz-Hardt, Frey, & Thelen, 2001, cités dans Geschke et al., 2019 ; Knobloch-Westerwick, Mothes, & Polavin, 2017, cités dans Geschke et al., 2019). Le second niveau de filtre concerne l'aspect social. Nous avons tendance à devenir amis et à développer un réseau social avec des personnes qui partagent les mêmes caractéristiques sociodémographiques et comportements que nous (McPherson, Smith-Lovin & Cook, 2001, cités dans Geschke et al., 2019). Troisièmement, intervient le niveau technologique avec les algorithmes de systèmes de recommandation. Selon Geschke et al. (2019), c'est ce troisième filtre qui divise encore plus la société en chambres d'écho distinctes et encore moins connectées les unes aux autres. Finalement, le filtrage des informations sur ces différents points peut mener à diminuer la variété des informations proposées à l'utilisateur et créer des bulles filtrantes (Geschke et al., 2019).

D'autres auteurs (Nikolov et al., 2015) partagent également le même point de vue quant au fait que les systèmes de filtres sont une combinaison à la fois des algorithmes de recommandations, mais également de l'aspect social. Selon eux, les individus choisissent eux-mêmes de s'entourer d'autres personnes partageant les mêmes points de vue qu'eux. Par exemple sur Facebook, les utilisateurs ont tendance à choisir en amis des personnes qui leur sont plutôt similaires (Nikolov et al., 2015). Selon ces différents chercheurs, une composante algorithmique ainsi qu'une composante humaine entreraient alors en jeu dans la création de bulles filtrantes.

3.2. Fragmentation et/ou homogénéisation des individus ?

De nombreux auteurs débattent quant à savoir si les systèmes de recommandation et le fait que les utilisateurs soient enfermés dans leurs préférences dominantes provoquent finalement une fragmentation des individus ou, au contraire, leur homogénéisation. Certains d'entre eux s'accordent sur le fait que ces deux phénomènes, fragmentation et homogénéisation des individus, ont lieu simultanément. Est-ce que les systèmes de recommandation divisent ou rassemblent les individus ? Nous allons analyser les points de vue de différents chercheurs dans cette section.

Tout d'abord commençons par définir la fragmentation et l'homogénéisation. Selon Hosanagar et al. (2014), le phénomène de fragmentation apparaît lorsque les utilisateurs partagent de moins en moins d'éléments en commun les uns avec les autres. Alors qu'à l'inverse, les recommandations ont des effets homogènes lorsque les utilisateurs partagent des informations avec d'autres personnes avec qui ils n'auraient pas communiqué sans ces recommandations.

Selon Pariser (2011), les systèmes de recommandation provoquent une fragmentation des individus qui affecte leur capacité de choix. Les utilisateurs n'ont même pas conscience de toutes les possibilités qui les entourent en étant confinés dans leurs préférences dominantes. Il pense que la fragmentation et l'enfermement dans une bulle filtrante peuvent avoir des conséquences négatives importantes notamment concernant la démocratie dont nous reparlerons plus tard dans ce travail. Toutefois, cette fragmentation est bien utile dans le marketing étant donné qu'en fonction du produit présenté, le public visé ne sera pas le même (Pariser, 2011). Sunstein (2007) pense également que les systèmes de recommandation provoquent de la fragmentation dans le domaine du filtrage des informations. Il met en évidence le fait que ces systèmes de filtrage limitent l'exposition des médias aux utilisateurs uniquement en leur présentant des éléments qui suscitent leur intérêt.

D'autres auteurs (Hosanagar et al., 2014) ont étudié ce phénomène de fragmentation à travers des systèmes de recommandation de musiques. Ils ont pu observer, contrairement à ce que suggère Pariser, que les recommandations ne provoquaient pas de la fragmentation mais une hausse des points communs (traduits de l'anglais : « commonality ») entre les utilisateurs. En effet, ils démontrent que le système de recommandation mis en place aide les utilisateurs à explorer de nouveaux produits et élargir leurs préférences. Évidemment, les chercheurs précisent que ces résultats ne sont pas forcément valables pour tous les systèmes de recommandation. Mais ils prouvent que la fragmentation n'est pas universellement vraie.

Plusieurs auteurs (Geschke et al., 2019 ; Negroponte, 1995, cité dans Hosanagar et al, 2014 ; Van Alstyne & Brynjolfsson, 2005, cités dans Hosanagar et al., 2014) ont démontré qu'il était possible que les systèmes de recommandation provoquent à la fois de la fragmentation et de l'homogénéisation des individus. Selon Geschke et al. (2019), l'apparition d'un de ces phénomènes dépend du but du système de recommandation mis en place. Parfois, le système de filtrage peut mener à diminuer la variété des informations proposées à l'utilisateur et créer des bulles filtrantes. À l'inverse, l'objectif d'un site internet peut être de maximiser le taux de clics. Dans ce cas, il se peut que le système de filtrage augmente l'exposition à différents points de vue afin que l'utilisateur soit curieux de découvrir ce qui est dit.

3.3. Les dangers des bulles filtrantes

Plusieurs problèmes ont pu être identifiés par différents auteurs comme étant des conséquences des systèmes de recommandation et de l'apparition des bulles filtrantes. Un débat a notamment lieu de nos jours concernant l'accès biaisé aux informations ainsi que l'impact de celui-ci sur la démocratie. Des discriminations raciales et sexistes ont même pu être constatées à travers certains systèmes de recommandation.

Un accès biaisé aux informations

Comme mentionné précédemment, Pariser (2011) démontre que les systèmes de recommandation, en plaçant les individus dans une bulle filtrante, les empêchent d'avoir accès à des informations qui ne leur plaisent pas forcément et sont tout de même importantes. Dès lors, les individus prennent des choix sans connaître l'ensemble des possibilités qui leur sont offertes. Selon Pariser (2011), il faudrait avoir une vue d'ensemble pour pouvoir prendre des décisions. Or, il estime que ce n'est pas le cas avec les systèmes de filtrage. Bozdag et Van den Hoven (2015) ainsi que Helberger et al. (2018) sont du même avis : les bulles filtrantes réduisent la diversité des informations auxquelles les utilisateurs ont accès. Selon Pariser (2011), un effet de distorsion de la réalité est alors provoqué car les bulles filtrantes agissent comme un intermédiaire entre le processus mental des individus et leur environnement extérieur.

Pariser (2011, pp 84-88) explique que notre cerveau fonctionne comme une balance cognitive qui permet à la fois de renforcer nos idées existantes et d'en acquérir de nouvelles. Malheureusement, les bulles filtrantes peuvent mettre en péril cette balance car les informations

qui nous sont apportées tendent à amplifier la confirmation de nos idées plutôt que de nous confronter à d'autres points de vue. Sur ce point, Sumpter (2018) partage le même avis que Pariser. Il expose même que les bulles filtrantes dans les systèmes d'informations peuvent conduire à des théories conspirationnistes. En effet, les individus qui cherchent à valider leur théorie vont se voir proposer des informations qui la confirment. Ils sont alors placés dans une bulle conspirationniste et ne voient pas les avis extérieurs qui pourraient les faire revenir à la raison (Sumpter, 2018, pp 142-144).

De plus, un autre problème lié à la diffusion des informations dans notre société est identifié : parmi toute cette masse de données disponibles sur internet, certaines informations sont fausses (Nunamaker, Burgoon & Giboney, 2016 ; Sumpter, 2018). En effet, les algorithmes utilisés n'analysent pas la signification de ce qui est partagé et n'en vérifient pas la véracité. Pourtant, les utilisateurs ont accès à ces informations et pourraient les croire vraies.

Impact sur la démocratie

Différents auteurs (Allcott & Gentzkow, 2017 ; Bozdag & Van den Hoven, 2015 ; Geschke et al., 2019 ; Pariser, 2011 ; Sumpter, 2018 ; Sunstein, 2018, cité dans Geschke et al., 2019) ont démontré que les bulles filtrantes pouvaient également avoir un impact sur la démocratie.

Comme nous l'avons expliqué précédemment, l'accès filtré aux informations implique la prise de décision dans un environnement où les utilisateurs n'ont pas conscience de tous les choix qui s'offrent à eux. Certaines informations sont rendues invisibles à leurs yeux (Pariser, 2011). Selon Bozdag et Van den Hoven (2015), il s'agit d'une atteinte à la démocratie car les systèmes de recommandation limitent la liberté de choix des utilisateurs ainsi que leur liberté de penser car les pensées peuvent être contraintes par les filtres. En étant enfermé dans une bulle filtrante, l'utilisateur reçoit majoritairement des informations qui confirment son point de vue et ses croyances. Il ne peut donc pas discuter avec des personnes qui ont un avis opposé au sien car celles-ci sont hors de sa bulle (Bozdag & Van den Hoven, 2015). Les utilisateurs n'ont alors qu'une vision partielle de la réalité. Or, pour faire des choix, notamment en politique, Pariser (2011) estime qu'il est nécessaire d'avoir une vision globale de ce qui nous entoure. Selon lui, la démocratie ne peut fonctionner que si les individus sont capables de penser au-delà de leurs propres intérêts. Mais pour cela, une vision globale est nécessaire ainsi que la prise de conscience des besoins des autres citoyens afin de prendre des décisions ensemble, ce qui va dans la direction opposée des bulles filtrantes (Pariser, 2011, p164).

Les choix des individus sont donc biaisés non seulement par l'accès filtré aux informations mais également par la présence d'informations faussées. Ces dernières ont aussi un impact négatif sur la démocratie car elles peuvent influencer les individus dans leurs choix de candidat aux élections (Allcott & Gentzkow, 2017). Pariser (2011, p 141) exprime même qu'il est possible pour les gouvernements de manipuler la vérité. Lors des élections présidentielles américaines de 2016, une multitude de fausses informations ont été diffusées sur les deux candidats : Trump et Clinton (Sumpter, 2018). Malheureusement, toutes ces fausses informations influencent l'opinion public. Allcott et Gentzkow (2017) ont voulu mesurer l'effet de ces fausses informations sur les élections présidentielles mais n'ont pas réussi à estimer à quel point une information ou publicité influençait le choix de vote des électeurs. Une autre campagne de désinformation a eu lieu lors des élections françaises en 2017. Celle-ci a été lancée deux jours avant les élections avec la dénomination *#MacronLeaks*. De fausses informations sur Macron se mélangeaient aux vraies. (Sumpter, 2018)

Selon Geschke et al. (2019) et Sunstein (2018, cité dans Geschke et al., 2019), une fragmentation dans la sphère politique apparaît actuellement dans de nombreux pays du monde. Geschke et al. (2019) parlent notamment de la montée des partis populistes en Europe et des protestations contre l'immigration mais également de la campagne de Donald Trump et du Brexit. Ils pensent que ces différents événements, qui se sont développés rapidement, ont potentiellement été impactés par les technologies.

D'autres inconvénients

En plus de l'accès biaisé aux informations et de l'impact sur la démocratie qui en découle, d'autres conséquences des bulles filtrantes ont pu être identifiées telles qu'une diminution de la créativité et de l'innovation (Pariser, 2011). En effet, Pariser (2011) mentionne qu'être dans une bulle, qui filtre les informations à la place des individus, les encourage à adopter une approche plus passive. Cela peut s'expliquer par le fait que les utilisateurs n'ont pas la nécessité d'effectuer des explorations par eux-mêmes étant donné que le système de recommandation leur fournit directement l'information recherchée. De plus, l'environnement dans lequel les bulles filtrantes enferment les utilisateurs n'est pas un contexte propice à la créativité (Pariser, 2011).

Sumpter (2018) mentionne que les systèmes de recommandation peuvent mener à de la discrimination raciale et du sexisme. Dans son livre, il reprend l'exemple de Facebook, réseau

social sur lequel il est possible de publier des annonces ciblées. Par exemple, il est envisageable d'exclure des individus selon leur affinité ethnique (« African American », « Asian American », « Hispanic », ...). Selon Sumpter (2018), il s'agit clairement d'une preuve de discrimination proposée par Facebook. Datta, Tschantz et Datta (2014) ont également pu identifier une discrimination entre les genres mise en place par un système de recommandation. Effectivement, ils ont réalisé une expérience où des hommes et des femmes se rendaient sur des sites internet pour rechercher un emploi. Par la suite, les hommes ont reçu des publicités pour des emplois mieux rémunérés que ceux proposés aux femmes alors que les deux groupes possédaient des caractéristiques identiques hormis le genre.

4. Solutions pour contrer les bulles filtrantes

Après avoir observé toutes ces conséquences qui peuvent survenir avec les bulles filtrantes, nous pouvons nous demander comment il est possible d'éviter leur apparition. Certains auteurs ont identifié différentes solutions que nous allons aborder ici. La première consiste à introduire la diversité dans les algorithmes des systèmes de recommandation. La seconde solution réside dans l'implication des utilisateurs. Celle-ci se divise en plusieurs suggestions qui proposent d'accroître la conscience des utilisateurs, leur laisser plus de contrôle voire même de les inciter à manipuler les systèmes de recommandation.

4.1. Introduire la diversité dans les algorithmes

Dans la section précédente, nous avons identifié un problème majeur lié aux bulles filtrantes : l'accès biaisé aux informations. Nous mentionnions que les utilisateurs faisaient majoritairement face à des informations qui confirmaient leurs croyances et que celles-ci n'étaient alors pas remises en cause. Helberger et al. (2018) suggèrent qu'introduire la diversité dans les systèmes de filtrage de l'information permettrait de contrer les bulles filtrantes et forcément les inconvénients qui y sont liés. Selon eux, cela favoriserait l'échange d'idées et le dialogue entre différents points de vue et permettrait également la formation de l'opinion public raisonnée. Pariser (2011) suggère également aux entreprises d'exposer les individus à des informations en dehors de leurs expériences habituelles. De nombreux auteurs ont alors proposé des algorithmes qui introduisent la diversité et la nouveauté dans les recommandations. Nous allons aborder les suggestions d'algorithmes de Castagnos et al. (2014), de Zhou et al. (2010) ainsi que de Han et Yamana (2017).

Castagnos et al. (2014) ont comparé cinq systèmes de recommandation dont deux favorisant la diversité dans les recommandations, appelés FCDR (Filtrage Collaboratif avec Diversité Relative) et FCDF (Filtrage Collaboratif avec Diversité Fixe). Le premier consiste tout d'abord à obtenir une liste des 50 meilleures recommandations (appelée top-50) pour un individu selon un algorithme de filtrage collaboratif classique. Cette liste ne correspond pas à la liste des recommandations finale mais elle est utilisée pour l'obtenir. Le premier item recommandé correspond au premier élément du top-50. Ensuite, les éléments sont ajoutés un à un à la liste de recommandations finale en fonction de l'item du top-50 qui propose la plus grande diversité avec les éléments déjà présents dans la liste de recommandations finale. Cette dernière opération est effectuée jusqu'à ce que l'on obtienne le nombre de recommandations souhaité.

Le second algorithme introduisant la diversité proposé par Castagnos et al. (2014) est assez similaire au premier. La différence réside dans le fait que $x\%$ des éléments du top-50 sont directement ajoutés à la liste de recommandations finale et non pas uniquement le premier élément du top-50. Dans leur étude, ils ont fixé ce seuil à 60%. Ces deux algorithmes fournissaient alors des listes de recommandations plus diversifiées qu'un filtrage collaboratif classique.

D'autres auteurs (Zhou et al., 2010) ont proposé de créer un algorithme hybride qui permettrait à la fois de gagner en exactitude et en diversité. Ils proposent de combiner deux méthodes de recommandations : une qui donne de très bons résultats au niveau de l'exactitude des résultats et une autre qui permet d'avoir des recommandations nouvelles. Ils suggèrent alors d'introduire un paramètre d'hybridation qui permet d'attribuer un poids plus ou moins important à chacune des méthodes. Ensuite, il faut tester plusieurs valeurs de ce paramètre d'hybridation afin de voir quelle valeur permet d'obtenir les meilleurs résultats d'un point de vue de la nouveauté et de l'exactitude. Bien sûr, la valeur de ce paramètre est choisie en fonction de l'objectif visé au niveau de ces deux éléments.

Des algorithmes utilisant une méthode de clustering peuvent également être repris pour ajouter de la diversité au sein des recommandations établies pour un utilisateur. Han et Yamana (2017) expliquent qu'un utilisateur peut être représenté par plusieurs clusters. Chacun de ces clusters constitue une sous-caractéristique de l'utilisateur et reprend un ensemble de sous-recommandations. Afin d'obtenir l'ensemble de recommandations finales pour l'utilisateur, les différentes sous-recommandations, provenant des différents clusters, sont agrégées. Grâce à cela, les recommandations reprennent plusieurs sous-préférences de l'utilisateur.

4.2. Impliquer les utilisateurs

Une autre solution afin d'éviter l'apparition des bulles filtrantes consiste à impliquer plus les utilisateurs. Cette implication passe par différents niveaux.

Tout d'abord, plusieurs auteurs (Bozdag & Van den Hoven, 2015 ; Helberger et al., 2018 ; Pariser, 2011 ; Zhou et al., 2010) proposent que les utilisateurs choisissent eux-mêmes le degré de personnalisation des recommandations qui leur sont proposées. En mettant en évidence un algorithme hybride entre l'exactitude et la diversité, Zhou et al. (2010) suggèrent que les utilisateurs soient autorisés à tuner eux-mêmes le paramètre d'hybridation. Ils choisiraient alors la valeur de ce paramètre en fonction de s'ils souhaitent avoir des recommandations qui

s'adaptent plus à leurs préférences ou des recommandations plus diverses et nouvelles. De plus, il a été prouvé que les utilisateurs aiment contrôler le contenu de ce qu'ils consomment et préfèrent clairement les systèmes qui leur offrent cette possibilité (Akstrand, Kluver, Harper & Konstan, 2015, cités dans Helberger et al., 2018 ; Harper et al., 2015, cités dans Helberger et al., 2018 ; Komiak, Wang & Benbasat, 2005, cités dans Helberger et al., 2018). Pariser (2011) conseille aux utilisateurs d'utiliser des sites web qui permettent de choisir le degré de personnalisation et l'utilisation de leurs informations. Il explique notamment que Twitter laisse plus d'opportunités de contrôle de leur bulle filtrante aux utilisateurs que Facebook (Pariser, 2011, pp 225-227).

De plus, Bozdag et Van den Hoven (2015) suggèrent de renforcer la connaissance qu'a l'utilisateur de son propre biais. Ils ont alors créé un outil qui permet de montrer aux utilisateurs leurs habitudes de consommation en matière d'informations. Ceux-ci peuvent voir sur quels sites ils se rendent le plus souvent pour consulter des informations, ils prennent ainsi conscience de leur propre biais. À travers cet exemple, il s'agit plutôt d'une prise de conscience des sources d'informations plutôt que des systèmes de recommandation. Pariser (2011), quant à lui, estime que les entreprises devraient rendre leurs systèmes de filtration plus transparents envers le public et dire aux utilisateurs quelles informations elles possèdent sur eux et ce qu'elles en font.

Pariser (2011) va même jusqu'à inciter les utilisateurs à contourner les systèmes de recommandation. Il les invite notamment à supprimer leur historique web ainsi que les cookies de leur serveur internet. En outre, il les pousse à élargir leurs intérêts dans d'autres directions pour donner aux algorithmes de personnalisation une vue plus étendue concernant leurs préférences. Toutefois, Bozdag et Van den Hoven (2015) pensent qu'il ne faut pas arriver au point de tromper les systèmes de personnalisation car cela aurait un impact négatif pour les utilisateurs aussi.

5. Incorporation de la notion de temps

Le sujet de ce mémoire incorpore une notion de temps car, pour rappel, nous souhaitons étudier si les systèmes de recommandation enferment les utilisateurs, au cours du temps, dans leurs préférences dominantes. Dès lors, il devient intéressant de rechercher les études qui ont déjà été réalisées dans ce domaine. Ce qui nous préoccupe est d'évaluer l'évolution de la diversité et de la nouveauté au sein des recommandations à travers le temps. En effet, plus les valeurs obtenues sur ces métriques sont faibles, plus les items proposés à l'utilisateur se ressemblent et on pourrait alors supposer que celui-ci se retrouve de plus en plus enfermé dans ses préférences dominantes. Nous allons détailler ci-dessous différentes études qui pourraient nous aider dans nos recherches. La première étude concerne uniquement l'incorporation de la notion de temps en comparant différents algorithmes alors que les cinq études suivantes analysent la diversité à travers le temps.

5.1. Cremonesi et Turrin (2010)

L'étude réalisée par Cremonesi et Turrin (2010) analyse les performances de plusieurs algorithmes de filtrage collaboratif à travers le temps. Le terme performance est ici étudié en fonction de l'exactitude (« accuracy » en anglais) à travers la métrique de *Recall*. Comme expliqué précédemment, cette métrique exprime le pourcentage de films pertinents recommandés à l'utilisateur par rapport au nombre de films pertinents (recommandés ou pas). Pour la réalisation de cette étude, les chercheurs ont repris deux bases de données provenant toutes deux du domaine des télévisions interactives (ITV). Les notes attribuées par les utilisateurs aux items, étant ici des films, sont binaires, c'est-à-dire que l'utilisateur a vu le film (1) ou non (0). Deux algorithmes de filtrage collaboratifs sont étudiés. Le premier est un système de filtrage basé sur les k plus proches voisins en item-based où la similarité cosinus est calculée entre toutes paires d'items. Le second algorithme utilise la factorisation matricielle. Pour rappel, ce type d'algorithme établit un ensemble de facteurs latents pour chaque utilisateur ainsi que pour chaque item afin de les caractériser au mieux et de pouvoir estimer la note attribuée par un utilisateur à un certain item, c'est-à-dire dans notre cas, estimer si un utilisateur a vu un certain film ou non. Pour chaque période de temps t , il se peut que de nouveaux utilisateurs ou de nouveaux films apparaissent. Dès lors, $I(t)$ est défini comme étant l'ensemble des items actifs, c'est-à-dire ceux qui ont été vus par au moins un utilisateur en période t ou

avant. De la même manière, $U(t)$ représente l'ensemble des utilisateurs actifs, c'est-à-dire ceux qui ont vu au moins un film en période t ou avant.

Suite à cela, les chercheurs ont pu établir différents constats. Premièrement, ils ont constaté que la mesure du *Recall* décroît dans le temps avant d'atteindre une valeur stable. Ils expriment que cela peut être lié au fait que le nombre de films pris en considération augmente dans le temps suite à l'apparition de nouveaux films. Plus le nombre d'items est important, plus il sera difficile pour l'algorithme de sélectionner des items que les utilisateurs ont vu dans le test set. En comparant les deux algorithmes, Cremonesi et Turrin ont découvert que l'algorithme des k plus proches voisins offrait généralement une valeur du *Recall* plus importante que celui basé sur la factorisation matricielle et ce, pour les deux bases de données. En se concentrant cette fois uniquement sur l'algorithme de factorisation matricielle, les chercheurs ont remarqué que l'exactitude de ce système de filtrage décroissait au cours du temps lorsque peu de facteurs latents étaient utilisés. Par contre, lorsqu'un grand nombre de facteurs latents était utilisé, l'exactitude se voyait augmentée au cours du temps jusqu'à dépasser celle de l'algorithme des k plus proches voisins. De plus, le nombre de facteurs latents dépend également de l'objectif à atteindre. Si nous souhaitons mettre en avant des films moins connus, il est plus intéressant d'utiliser un grand nombre de facteurs latents dans la factorisation matricielle. En effet, les premiers facteurs captent les caractéristiques des items les plus populaires, il faut ensuite plus de facteurs pour capter le « bruit » autour formé par les films moins populaires.

En conclusion de cette étude, des résultats intéressants ont pu être observés dans la comparaison d'algorithmes de filtrage collaboratif à travers le temps. Mais, la notion de diversité n'a pas encore été introduite. C'est pourquoi nous allons maintenant nous attarder sur d'autres études qui ont analysé l'évolution de la diversité au cours des recommandations.

5.2. Nguyen et al. (2014)

Nguyen et al. (2014) ont étudié l'impact d'un seul système de recommandation, il s'agit d'un système de filtrage collaboratif basé sur les items. Ils ont alors analysé l'évolution de l'expérience utilisateur et de la diversité, variable qui nous intéresse ici. Ces métriques ont été calculées à deux moments différents, une fois au début de la période et une fois à la fin de celle-ci afin de pouvoir évaluer l'évolution des métriques dans le temps. Cette étude a eu lieu sur deux groupes d'utilisateurs. Le premier groupe représente l'ensemble des utilisateurs qui suivent régulièrement les recommandations qui leur sont faites alors que l'autre groupe

comprend les utilisateurs qui ne suivent jamais les recommandations. La base de données correspond à un ensemble de notes attribuées par les utilisateurs à des films.

Au terme de cette étude, ils ont pu constater que la diversité diminuait entre le début et la fin de la période et ce, dans les deux groupes d'utilisateurs. Lorsqu'on parle de diversité ici, il s'agit de la diversité au sein de la liste des films recommandés. Celle-ci a été calculée à partir de la distance moyenne entre chaque paire de films de la liste. Les auteurs ont pu conclure de cette étude que la diversité diminue au cours du temps et que les utilisateurs se voient proposer des films de plus en plus similaires. Toutefois, ces résultats ne sont pas à généraliser. En effet, seulement un algorithme a été étudié ici, il se pourrait que des résultats différents soient fournis par d'autres algorithmes. De plus, seulement deux moments dans le temps ont été observés. Il aurait pu être intéressant d'analyser plus de périodes pour observer les fluctuations. Dans les sections suivantes ainsi que dans nos propres recherches, nous essayerons de contrer ces deux inconvénients qui ne nous permettent pas de fournir des résultats généralisables pour l'instant.

5.3. Lathia (2010)

Lathia (2010) a également incorporé la notion de temps dans les systèmes de recommandation et plus précisément dans les algorithmes de filtrage collaboratif. À des intervalles de temps réguliers, le RMSE a été calculé afin de connaître l'évolution de l'exactitude dans le temps. Une deuxième métrique a été étudiée, il s'agit de la diversité. Mais celle-ci est différente de celle analysée par Nguyen et al. en 2014. Pour rappel, ces derniers avaient calculé la diversité au sein d'une liste de recommandations sur base de la similarité entre toutes paires d'items composant cette liste ; plus les items étaient similaires, plus la diversité était faible. Ici, la diversité est calculée entre deux listes de recommandations en regardant la proportion d'éléments communs aux deux listes par rapport au nombre total d'éléments des deux listes combinées. Plus le nombre d'éléments communs aux deux listes est élevé plus la diversité est faible. Un nombre élevé d'items identiques présents dans les deux listes signifie que ces listes de recommandations se ressemblent fortement et que les recommandations faites à l'utilisateur ne varient pas beaucoup dans le temps. Finalement, il ressort de cette étude que les systèmes de recommandation sont de plus en plus précis dans le temps mais fournissent une diversité de plus en plus faible.

5.4. Castagnos et al. (2014)

Nous allons maintenant aborder une troisième étude qui introduit cette notion de temps dans les systèmes de recommandation. Celle-ci a été réalisée par Castagnos et al. en 2014. Contrairement aux deux études précédentes, celle-ci ne se concentre pas uniquement sur les algorithmes de filtrage collaboratif. En effet, 5 algorithmes différents ont été étudiés : un basé sur la popularité recommandant uniquement les items les plus populaires, un autre utilisant le filtrage basé sur le contenu et finalement 3 algorithmes de filtrage collaboratif. Parmi les 3 algorithmes de filtrage collaboratif, 2 d'entre eux visent à proposer un contenu plus diversifié. Nous avons déjà décrit le fonctionnement de ces deux algorithmes dans la section précédente de ce travail. Ici aussi, les auteurs ont étudié la diversité à plusieurs intervalles de temps. Ceci a été analysé à travers la similarité intra-liste, tout comme l'étude de Nguyen et al. (2014). Pour chaque liste de recommandations établie, la similarité des items composant cette liste a été calculée. Plus la similarité intra-liste est élevée, plus cela signifie que les éléments composant cette liste se ressemblent et que la diversité est faible.

Au terme de cette étude, les auteurs ont pu établir différents constats. Premièrement, la similarité intra-liste est bien plus élevée pour l'algorithme dont le filtrage est basé sur le contenu. Cela signifie que les items proposés se ressemblent plus, la diversité est plus faible. Pour rappel, les algorithmes de filtrage basés sur le contenu cherchent à recommander des items dont les caractéristiques sont identiques aux items évalués positivement par l'utilisateur. Prenons l'exemple où les items correspondent à des films, si l'utilisateur apprécie les thrillers, on va lui recommander d'autres thrillers. Dès lors, il devient assez évident que les films recommandés se ressemblent étant donné que le but de l'algorithme est de proposer des items avec les mêmes caractéristiques que celles appréciées par l'utilisateur.

Deuxièmement, ils ont constaté que la similarité intra-liste restait plutôt stable au cours du temps pour tous les algorithmes sauf celui dont le filtrage est basé sur le contenu. Cela signifie que la diversité au sein des items recommandés semble rester stable dans le temps. Ce résultat est assez surprenant car il s'oppose à ceux obtenus par les deux études dont nous avons parlé précédemment. En ce qui concerne l'algorithme dont le filtrage est basé sur le contenu, la similarité intra-liste diminue au cours du temps, cela revient à dire que les items proposés se différencient plus et que la diversité augmente. Mais il ne faut pas oublier que cette similarité intra-liste reste malgré tout bien plus élevée que celles fournies par les autres systèmes de recommandation. Toutefois, ces résultats sont à considérer de manière précautionneuse.

Effectivement, les auteurs eux-mêmes reconnaissent que la base de données sur laquelle ils fondent leurs recherches est assez faible car elle ne contient qu'un peu plus de 500 films. Dès lors, il se pourrait que la similarité intra-liste fournie par le filtrage basé sur le contenu diminue au cours du temps car le nombre de films possédant le plus de caractéristiques en commun avec ceux appréciés par l'utilisateur diminue. Donc, des films possédant moins de caractéristiques appréciées seraient proposés et cela pourrait expliquer que les films recommandés soient de moins en moins similaires au cours du temps.

5.5. Nikolov et al. (2015)

Une autre étude, réalisée par Nikolov et al. (2015), a, cette fois-ci, analysé la diversité à travers les réseaux sociaux et les moteurs de recherches. Ces auteurs ont utilisé une base de données composée de trois caractéristiques : la page cible sur laquelle un utilisateur arrive en cliquant sur un lien, la page source sur laquelle l'utilisateur était lorsqu'il a cliqué sur le lien et le moment (« timestamp » en anglais) du clic. Malheureusement, pour des questions de protection des données privées, il n'était pas possible d'identifier les utilisateurs en particulier. Leur analyse s'est donc faite de manière collective afin d'étudier la diversité collective.

Pour mesurer la diversité, ils ont utilisé l'entropie de Shannon. Celle-ci permet d'évaluer la source d'information : plus cette source émet des informations différentes, plus l'entropie sera grande. À l'inverse, si cette source émet toujours la même information, son entropie sera minimale. (Wikipédia, 2019). Dans le cas de cette étude, visiter par exemple 10 pages depuis 10 sites différents donnera une diversité plus grande que de visiter 10 pages depuis la même page source (Nikolov et al., 2015).

Premièrement, ils ont pu identifier que la diversité obtenue sur les réseaux sociaux est bien plus faible que celle obtenue à travers les moteurs de recherche. Les auteurs ont alors supposé que les réseaux sociaux exposent les individus à un plus petit ensemble de sources d'informations que les moteurs de recherche. Une deuxième constatation fut qu'ils ne remarquèrent pas de réelle évolution de la diversité au cours du temps que ce soit sur les réseaux sociaux ou les moteurs de recherche.

5.6. Szlavik et al. (2011)

Szlavik et al. (2011) ont également étudié la diversité dans les systèmes de recommandation à travers le temps. Pour cela, ils se sont basés sur une base de données réelles fournie par Netflix. Chaque ligne de la base de données était composée des informations suivantes : `user_id` (l'identifiant de l'utilisateur), `movie_id` (l'identifiant du film), `date` (le moment où l'utilisateur a attribué une note à ce film) et `rating` (la note entre 1 et 5 attribuée par l'utilisateur à ce film). Ils ont choisi d'étudier un seul algorithme qui est celui de Funk. Il s'agit d'un algorithme utilisant la factorisation matricielle. Afin d'incorporer la notion de temps dans leur travail, ils ont établi différents « round » représentant chacun une période d'un mois. À chaque « round », des recommandations sont faites à l'utilisateur et celui-ci adopte un certain comportement, c'est-à-dire qu'il évalue plusieurs films qui lui étaient recommandés ou non, le modèle de facteurs latents est alors mis à jour en fonction de la réponse de l'utilisateur. En établissant les recommandations, les chercheurs ont également veillé à ne pas recommander des films qui n'étaient pas encore sortis à cette période de temps.

Les chercheurs ont choisi de tester plusieurs comportements de réponse des utilisateurs. Le premier type de comportement est appelé « Yes-men » et consiste à ce que l'utilisateur note uniquement des films qui lui ont été recommandés. Le nombre de films évalués correspond au nombre de films repris dans la base de données pour le round suivant pour cet utilisateur. Le second comportement de réponse (« Uniform Yes-men ») est assez similaire au premier car l'utilisateur n'attribue des notes qu'à des films qui lui étaient recommandés mais cette fois, le nombre de films évalués suit une loi uniforme, c'est-à-dire que tous les utilisateurs évaluent le même nombre de films. Le troisième comportement est appelé « Trend-followers ». L'utilisateur évalue uniquement des films qui ont les meilleures notes en moyenne. Le nombre de films évalués correspond au nombre de films repris dans la base de données pour le round suivant pour cet utilisateur. Le quatrième comportement, « Uniform Trend-followers », est identique au troisième sauf que le nombre de films évalués par l'utilisateur suit une loi uniforme. Dans le cinquième comportement, « Randomisers », l'utilisateur évalue des films de manière aléatoire parmi l'ensemble des films disponibles. Tout comme pour les premier et troisième comportements, le nombre de films évalués correspond au nombre de films repris dans la base de données pour le round suivant pour cet utilisateur. Pour le sixième et dernier comportement évalué, « 25% Yes-Randomisers », la probabilité que l'utilisateur note un film qui lui était recommandé est de 25%. Dans le reste des cas, l'utilisateur évalue un film choisi aléatoirement parmi l'ensemble des films disponibles, autres que ceux présents dans la liste des

films recommandés. Le nombre de films évalués correspond au nombre de films repris dans la base de données pour le round suivant pour cet utilisateur.

À chaque round, la diversité a été étudiée à travers cinq mesures différentes. Tout d'abord, les chercheurs ont regardé le nombre de films évalués dans le round considéré. Il s'agit d'une mesure globale et non pas par utilisateur, elle permet de connaître l'étendue des films évalués. La seconde mesure étudie la variance globale des notes attribuées dans le round considéré. Il s'agit également d'une mesure globale qui permet, cette fois, d'étudier l'étendue des valeurs des notes de ce round. Troisièmement, pour chaque utilisateur, la variance des notes qu'il a attribuées lors de ce round est étudiée. Afin d'avoir une mesure globale, la moyenne des variances pour tous les utilisateurs est calculée. La quatrième mesure est basée sur l'entropie de Shannon dont nous avons parlé dans l'étude de cas précédente. La dernière mesure de diversité calcule la diversité entre toutes paires d'utilisateurs en se basant sur la similarité cosinus. Ces mesures de diversité sont donc toutes différentes des mesures de diversité utilisées dans les études précédemment présentées.

Suite à cela, il est possible d'établir plusieurs constats. Tout d'abord, la moyenne des notes attribuées par les utilisateurs augmente au fur et à mesure des rounds. Deuxièmement, les modèles uniformes ont été comparés à leur version non uniforme. Les chercheurs ont alors remarqué qu'introduire la notion d'uniformité dans le comportement des utilisateurs réduit de manière importante le nombre de films évalués. En utilisant les versions uniformisées des modèles, ils ont également constaté que la variance de l'ensemble des films évalués avait tendance à diminuer alors que la variance des notes attribuées par chaque utilisateur avait tendance à augmenter. Troisièmement, ils ont relevé que la sélection aléatoire de films mène à des notes moyennes plus faibles et donc une satisfaction plus faible des utilisateurs. Concernant l'entropie de Shannon ainsi que la diversité cosinus entre les utilisateurs, c'est le modèle aléatoire qui donne les meilleurs résultats. En conclusion de cette étude, les chercheurs ont mentionné qu'offrir des films aléatoirement permettait d'avoir une diversité plus élevée mais des scores plus faible. Alors qu'offrir des films plus populaires menait à une diversité plus faible mais des scores plus élevés. De plus, ils n'ont pas relevé de variation de la diversité au cours du temps.

5.7. Conclusions de ces études

Après l'analyse de ces différentes études, il est nécessaire de faire un résumé des résultats qui ont un impact sur notre domaine de recherche. Certains auteurs (Lathia, 2010 ; Nguyen et al., 2014) s'accordent à dire que la diversité diminue au cours du temps dans les listes de recommandations établies par les algorithmes de filtrage collaboratif. Cependant, Castagnos et al. (2014), Nikolov et al. (2015) ainsi que Szlavik et al. (2011) n'ont pas remarqué de variation de la diversité dans le temps. Castagnos et al. (2014) ont toutefois constaté que la similarité intra-liste était plus élevée dans les recommandations établies par les algorithmes dont le filtrage était basé sur le contenu plutôt que dans ceux utilisant le filtrage collaboratif. Quant à Nikolov et al. (2015), ils ont pu identifier que les réseaux sociaux proposaient un contenu moins diversifié par rapport aux moteurs de recherche.

Conclusion

À travers cette première partie, nous avons pu établir une revue de la littérature en analysant les études et points de vue des chercheurs dans le domaine. Nous avons alors découvert le fonctionnement ainsi que les différents types de systèmes de recommandation existants. Nous avons également appris que les systèmes de recommandation pouvaient augmenter le chiffre d'affaire des entreprises mais aussi la confiance que les utilisateurs y placent tout en leur offrant un gain de temps.

Ensuite, nous avons présenté les principales métriques afin d'étudier les performances des systèmes de recommandation. Celles qui nous préoccuperont pour la suite du travail sont les mesures de diversité et de nouveauté dans l'objectif d'étudier l'enfermement des utilisateurs.

À la suite de cela, nous avons pu découvrir les conséquences principales d'un enfermement des utilisateurs dans leurs préférences dominantes telles qu'un accès biaisé aux informations ou des impacts sur la démocratie.

Afin de surpasser ces différentes conséquences, nous avons étudié les solutions qui permettent de contrer les bulles filtrantes. Dans cette optique, plusieurs chercheurs ont proposé des algorithmes introduisant plus de diversité ou de nouveauté au sein des recommandations. Ceux-ci pourront nous être utiles pour la suite de nos recherches.

Finalement, nous avons mis en évidence plusieurs études incorporant la notion de temps. Dans la suite de ce travail, nous allons donc tenter d'étudier, nous aussi, la variation de la diversité et de la nouveauté dans le temps à travers plusieurs algorithmes afin de confronter les points de vue des différents chercheurs.

Chapitre 2 : méthodologie

Introduction

Dans ce chapitre, nous allons présenter la méthodologie que nous comptons appliquer afin de réaliser notre étude. Notre objectif est alors de tester plusieurs algorithmes de systèmes de recommandation et d'analyser l'évolution de la diversité et de la nouveauté, au sein des recommandations établies, au cours du temps.

Pour ce faire, le langage de programmation Python va être nécessaire, ceci sera expliqué plus en détail dans la section concernant les outils utilisés.

À la suite de cela, nous aborderons la base de données choisie pour réaliser notre étude. Il s'agit d'une base de données MovieLens reprenant des notes attribuées par des utilisateurs à des films. Nous détaillerons également les différentes manipulations prévues sur cette base de données avant que celle-ci ne soit exploitable pour la suite de nos recherches.

Ensuite, sept algorithmes différents seront présentés. Cinq d'entre eux sont des algorithmes de filtrage collaboratif alors que les deux restants consistent respectivement à recommander des films de manière aléatoire et des films populaires.

Après cela, nous détaillerons deux métriques qui permettent de mesurer la diversité ainsi que la nouveauté au sein des recommandations établies pour les utilisateurs.

Enfin, nous présenterons cinq scénarios à tester qui feront évoluer la base de données, chacun de manière différente. Certains de ces scénarios incorporeront à la fois une dimension humaine et une dimension algorithmique alors que d'autres n'en incorporeront qu'une des deux. Pour rappel, différents chercheurs (Geschke et al., 2019 ; Nikolov et al., 2015) estiment que l'apparition des bulles filtrantes est principalement due à deux composantes : algorithmique et humaine. La composante algorithmique est liée aux systèmes de recommandation qui enfermeraient potentiellement les utilisateurs dans une bulle alors que la composante humaine se base sur le fait que les individus s'enfermeraient eux-mêmes dans une bulle filtrante du fait de leurs choix et préférences personnelles. L'objectif de ceci est de pouvoir étudier l'impact des algorithmes sur l'enfermement potentiel des utilisateurs. Parmi les différents scénarios, trois comportements de réponse des utilisateurs face aux recommandations seront également testés.

1. Outils utilisés

Pour réaliser nos recherches, nous avons choisi d'utiliser le langage de programmation Python 3 et, pour ce faire, de passer par l'environnement Jupyter Notebook. Cet environnement nous permet de combiner à la fois des parties de code ainsi que du texte structuré. Afin d'avoir recours à cet environnement, nous avons téléchargé Anaconda Navigator, qui est une plateforme gratuite permettant de gérer l'environnement Jupyter Notebook et d'avoir accès à toute une série de bibliothèques disponibles pour le langage Python. Les différentes parties de la méthodologie seront donc réalisées à l'aide de ces outils.

2. Choix et analyse de la base de données

2.1. Présentation et division de la base de données

Nous avons choisi d'utiliser une base de données MovieLens reprenant les notes, variant sur une échelle de 1 à 5, attribuées par des utilisateurs à des films. Cette base de données comprend, plus exactement, 20 000 263 notes attribuées par 138 493 utilisateurs sur 27 278 films. Nous savons également que chaque utilisateur a évalué au moins 20 films. Chaque ligne de la base de données comprend donc 4 éléments : « user_id » correspondant à l'identifiant attribué à un utilisateur, « item_id » correspondant à l'identifiant attribué à un film, « rating » représentant la note attribuée par cet utilisateur à ce film et « timestamp » signifiant le moment auquel la note a été attribuée, ou plus précisément, le nombre de secondes qui se sont écoulées au moment où la note a été attribuée depuis le 1^{er} janvier 1970. Par exemple, une des lignes de la base de données pourrait être la suivante : « 23, 65, 4, 1111622106 ». Cela signifierait que l'utilisateur avec l'identifiant 23 aurait attribué une note de 4/5 au film dont l'identifiant est le 65, et ce, le 23 mars 2005 à 23h55 (ce qui correspond à 1111622106 secondes après le 1^{er} janvier 1970).

Notre base de données est stockée dans un fichier CSV. Il s'agit de l'abréviation de « Comma-separated Values » (Wikipédia, 2020). Ce type de fichier contient des données tabulaires, c'est-à-dire des données qui pourraient être stockées dans des tableaux comme c'est le cas des fichiers Excel (.xls et .xlsx) par exemple. Dans un fichier CSV, les données sont séparées par des virgules. Donc, nos données seront disponibles sous la forme présentée à la Figure 2.1.

Dans le but d'incorporer la notion de temps dans notre analyse, la base de données sera ordonnée en fonction de la caractéristique « timestamp » de manière à ce que les éléments plus anciens se situent sur le haut de celle-ci et les plus récents dans le bas. Suite à cela, la base de données sera divisée en x parties. Une fois les divisions réalisées, les utilisateurs qui ne seraient pas présents dans chacune de ces divisions seront alors éliminés de la base de données. En effet, nous souhaitons étudier l'évolution au cours du temps, il faut donc que tous les utilisateurs soient présents du début à la fin de notre étude. Plusieurs valeurs de x , représentant le nombre de divisions, seront testées : 5, 10 et 20 afin de trouver la valeur de x qui nous permet d'avoir un nombre d'utilisateurs suffisant pour pouvoir réaliser notre étude.

La base de données que nous avons choisie contient des notes qui ont été attribuées entre le 9 janvier 1995 et le 31 mars 2015. Imaginons que nous divisons la base de données en 5 parties. Nous aurions alors par exemple la première partie qui contiendrait les notes attribuées entre le

9 janvier 1995 et le 25 janvier 1999. Ensuite, la seconde partie contiendrait les notes attribuées entre le 26 janvier 1999 et le 16 février 2003 et ainsi de suite. Nous conserverions alors dans chacune des parties uniquement les notes qui ont été attribuées par des utilisateurs ayant évalué des films dans chacune des parties temporelles.

La première division de la base de données correspondra à la matrice utilisée dans la première période de temps par les algorithmes afin de fournir des recommandations. Cette matrice évoluera au cours de l'analyse en fonction des périodes de temps et des comportements des utilisateurs. Cela sera abordé plus en détail dans la partie concernant la simulation du comportement des utilisateurs (cf. point 0 de ce chapitre).

2.2. Conversion des données en une matrice

Pour chaque division temporelle, nous souhaitons convertir les données qui y sont reprises en une matrice. Les données présentes dans une de nos divisions temporelles pourraient, par exemple, être celles indiquées dans la Figure 2.1.

```
1, 1, 2, 1108592542
2, 3, 2, 1111789342
3, 2, 2, 1113217871
2, 1, 5, 1115089383
3, 4, 4, 1122333112
1, 2, 3, 1124118315
2, 4, 4, 1126527132
3, 1, 3, 1128986721
1, 5, 4, 1135555199
```

Figure 2.1. Exemple de données

Notre but est alors de convertir ces données en une matrice ayant la forme suivante : les lignes correspondent aux utilisateurs et les colonnes aux films. En ce qui concerne la manière de remplir ces matrices, nous nous retrouvons face à deux choix principaux :

1. Indiquer dans la matrice les différents scores attribués aux films par les utilisateurs (cf. Figure 2.2(a)).
2. Constituer une matrice binaire contenant uniquement des 0 et des 1 (cf. Figure 2.2(b) et (c)). Dans ce cas, nous avons à nouveau deux possibilités pour choisir comment attribuer les valeurs de 1 ou de 0 :

- i. La première consiste à mettre la valeur de 1 dans la matrice lorsque l'utilisateur a vu le film, c'est-à-dire qu'il lui a attribué une note variant de 1 à 5, et la valeur de 0 lorsqu'il n'a pas vu le film (cf. Figure 2.2(b)).
- ii. Notre deuxième possibilité est d'indiquer la valeur de 1 dans la matrice uniquement si l'utilisateur a apprécié le film, c'est-à-dire qu'il lui a attribué une note supérieure à 3, et de mettre toutes les autres valeurs à 0, c'est-à-dire pour tous les films que l'utilisateur n'a pas vus ou pour les films qu'il a vus mais n'a pas appréciés (cf. Figure 2.2(c)).

Nous disposons des trois choix décrits ci-dessus pour convertir les données présentées à la Figure 2.1 en une matrice où les utilisateurs représentent les lignes et où les colonnes correspondent aux films (cf. Figure 2.2).

$$\begin{pmatrix} 2 & 3 & 0 & 0 & 4 \\ 5 & 0 & 2 & 4 & 0 \\ 3 & 2 & 0 & 4 & 0 \end{pmatrix}$$

a. Scores exacts

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

b. Matrice binaire
(1 = film vu)

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

c. Matrice binaire
(1 = film apprécié)

Figure 2.2. Représentations possibles des données dans une matrice

Nous avons choisi d'éliminer la première option (cf. Figure 2.2(a)) et de nous concentrer sur une matrice binaire car notre étude concerne uniquement l'analyse des recommandations et pas des prédictions. En effet, les termes de *recommandations* et *prédictions* sont bien différents. Proposer des recommandations aux utilisateurs consiste à leur suggérer des films qu'ils pourraient potentiellement apprécier. Alors que réaliser des prédictions réside dans le fait d'estimer les scores que les utilisateurs pourraient attribuer à ces films. Étant donné que nous n'étudierons que les recommandations et non pas les prédictions, une matrice binaire peut très bien convenir pour notre étude. Aggarwal (2016) nous apprend dans son livre qu'utiliser une matrice binaire plutôt qu'une matrice contenant des scores exacts présente différents avantages dont le principal concerne l'incorporation de 0 dans les matrices pour remplacer les données manquantes. Selon lui, un biais plus important est incorporé lorsque l'on inscrit une valeur de 0 pour les données manquantes dans une matrice avec les scores exacts plutôt que dans une matrice binaire.

Dès lors, il nous reste à choisir entre les deux possibilités de matrices binaires. Nous avons identifié un inconvénient majeur à l'option 2.ii. (cf. Figure 2.2(c)) : étant donné que nous ne

distinguons pas les films que l'utilisateur n'a pas vus de ceux qu'il n'a pas appréciés, il serait possible de lui recommander un film qu'il a déjà vu et n'a pas apprécié. Cela étant assez dommageable pour l'expérience utilisateur, nous avons finalement opté pour l'option 2.i. (cf. Figure 2.2(b)).

2.3. Résumé du travail sur les données

En résumé, nous allons ordonner notre base de données initiale en fonction de la date à laquelle chacune des notes a été attribuée. Ensuite, nous la diviserons en x parties. Chacune des parties contiendra donc les données, issues de la base de données initiale, appartenant à une certaine période temporelle. Nous éliminerons également toutes les lignes de données référant aux utilisateurs qui ne seraient pas présents dans toutes les parties. Après cela, pour chacune des parties de la base de données, nous créerons une matrice où les lignes correspondent aux utilisateurs et les colonnes aux films et où une valeur de 1 est attribuée lorsque l'utilisateur a vu le film et une valeur de 0 lorsqu'il ne l'a pas vu.

3. Étude de plusieurs algorithmes

Afin d'établir des recommandations, les algorithmes se baseront sur une matrice binaire du même type que ce que nous venons de présenter, à savoir avec une valeur de 1 lorsque l'utilisateur a vu le film et de 0 lorsqu'il ne l'a pas vu. Dans cette matrice, les lignes représentent les utilisateurs et les colonnes correspondent aux films. En voici un exemple à la Figure 2.3.

	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	1	0	0	1	0	0	1	0	1
Utilisateur 2	0	0	1	1	0	0	1	0	0	1
Utilisateur 3	1	1	0	1	0	0	1	0	1	0
Utilisateur 4	0	0	1	0	1	1	0	1	0	0
Utilisateur 5	0	1	0	0	1	1	0	0	1	1

L'utilisateur 1 a vu le film 2

Figure 2.3. Exemple de matrice de données sur laquelle nos algorithmes se basent pour établir des recommandations.

À la première période de temps, nous nous baserons sur la première matrice pour proposer des recommandations aux utilisateurs. Pour chaque période de temps suivante étudiée, cette matrice initiale évoluera en fonction du comportement des utilisateurs face aux recommandations. Ceci sera détaillé plus précisément au point 5 de ce chapitre.

Sept algorithmes différents seront testés. Pour l'ensemble de ceux-ci, un certain nombre de recommandations à fournir à chaque utilisateur sera fixé. Plusieurs valeurs seront testées : 10, 20 et 100. Nous avons choisi de tester les valeurs de 10 et 20 car nous supposons qu'au-delà de 20 recommandations, l'utilisateur ne les consulterait probablement pas toutes en raison d'un nombre trop important. En effet, à travers plusieurs études reprises dans la revue de littérature, nous avons pu identifier que le nombre de recommandations varie généralement entre 10 et 20 (Castagnos et al., 2014 ; Nguyen et al., 2014). Toutefois, certaines études (Castagnos et al., 2014 ; Cremonesi & Turrin, 2010) se sont également attelées à proposer un nombre plus important de recommandations. C'est pourquoi nous avons choisi de tester la valeur de 100 également. En sélectionnant ces différentes valeurs de recommandations, nous pourrons

comparer les résultats en ce qui concerne la nouveauté et la diversité présentes au sein des recommandations établies par chacun des algorithmes. Cela nous permettra alors de voir si le nombre de recommandations impacte l'enfermement potentiel des utilisateurs dans une bulle filtrante.

Pour chacun des 7 algorithmes que nous allons présenter, nous prendrons un exemple sur base des données reprises dans la Figure 2.3 afin d'expliquer le fonctionnement de l'algorithme concerné et de montrer comment les recommandations sont établies pour l'utilisateur 1.

3.1. Algorithme de filtrage collaboratif item-based

Le premier algorithme est un filtrage collaboratif orienté item. Par la suite, nous abrègerons son nom par *algorithme item-based*. Pour rappel, la similarité entre toutes paires d'items, c'est-à-dire entre chaque paire de films dans notre cas, est calculée. Pour cela, nous avons choisi d'utiliser la mesure de similarité cosinus dont la formule se trouve ci-dessous (6) (Breese, Heckerman, & Kadie, 2013).

$$\text{Cosine sim}(i, j) = \frac{\sum_{u \in U_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in U_i} r_{ui}^2} \sqrt{\sum_{u \in U_j} r_{uj}^2}} \quad (6)$$

Où $\text{Cosine sim}(i, j)$ est la similarité cosinus calculée entre l'item i et l'item j ;

r_{ui} vaut 1 lorsque l'utilisateur u a vu le film i et vaut 0 lorsqu'il ne l'a pas vu ;

U_{ij} correspond à l'ensemble des utilisateurs qui ont vu le film i et le film j ;

U_i représente l'ensemble des utilisateurs qui ont vu le film i ;

U_j correspond à l'ensemble des utilisateurs qui ont vu le film j .

Imaginons que la matrice sur laquelle l'algorithme se base pour établir des recommandations soit celle présentée à la Figure 2.3, nous obtiendrions alors la matrice de similarité entre les items présentée à la Figure 2.4.

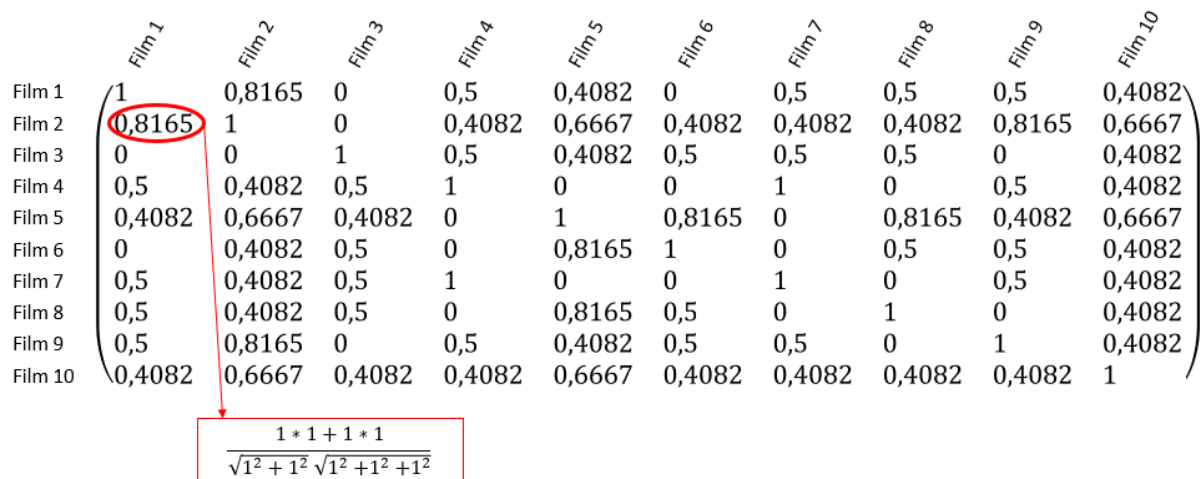


Figure 2.4. Matrice de similarité cosinus entre les items

À travers la matrice de similarités disponible à la Figure 2.4, nous observons que tous les films ont une similarité avec eux-mêmes d'une valeur de 1. Cette valeur ne sera évidemment pas prise en considération lors de la détermination des voisins de tous les films car un film ne peut pas être son propre voisin.

L'équation (6), calculant les similarités, permet donc de déterminer, pour chaque film i , ses k plus proches voisins (i.e., les k films ayant la similarité la plus élevée avec le film i). Si nous reprenons notre exemple et que nous décidons d'identifier les $k=3$ plus proches voisins pour chaque film, nous obtenons les listes de voisins reprises à la Figure 2.5.

	Liste des $k=3$ plus proches voisins
Film 1	Films 2, 4 et 7
Film 2	Films 1, 9 et 5
Film 3	Films 4, 6 et 7
Film 4	Films 7, 1 et 3
Film 5	Films 6, 8 et 2
Film 6	Films 5, 3 et 8
Film 7	Films 4, 1 et 3
Film 8	Films 5, 1 et 3
Film 9	Films 2, 1 et 4
Film 10	Films 2, 5 et 1

Figure 2.5. Liste des $k=3$ plus proches voisins pour chaque film

La liste de n recommandations proposées à un utilisateur u sera alors constituée des n films (non déjà vus par l'utilisateur u) les plus proches de l'ensemble constitué des films déjà vus par

l'utilisateur u . Pour ce faire, nous ajouterons tout d'abord à la liste de recommandations le premier voisin de chacun des films vus par l'utilisateur. Ensuite, nous ajouterons le second voisin de chacun des films vus par l'utilisateur et ainsi de suite jusqu'à ce que le nombre de recommandations demandées ait été atteint.

Reprenons notre exemple pour clarifier cela. Étant donné que notre base de données d'exemple est petite, nous allons supposer que nous devons recommander seulement 2 films à chaque utilisateur. Déterminons, sur base de l'algorithme présenté, la liste de recommandations à proposer au premier utilisateur. Pour ce faire, nous avons repris à la Figure 2.6 la méthodologie appliquée pour déterminer la liste de recommandations pour cet utilisateur. Tout d'abord nous parcourons un à un le premier voisin de chaque film vu par l'utilisateur et recommandons uniquement ceux qui n'ont pas encore été vus par celui-ci. Dans notre cas, seul le film 6 peut être recommandé. Ensuite, nous passons au parcours des seconds voisins de chaque film vu par l'utilisateur et nous pouvons alors recommander le film 4. La liste de recommandations ayant atteint le nombre demandé, à savoir deux films, l'algorithme s'arrête. Le même procédé sera alors appliqué pour établir les listes de recommandations de tous les utilisateurs.

Films vus par l'utilisateur 1	Voisins de ces films
Film 1	Films 2 , 4 et 7
Film 2	Films 1 , 9 et 5
Film 5	Films 6, 8 et 2
Film 8	Films 5 , 1 et 3
Film 10	Films 2 , 5 et 1

Figure 2.6. Établir les recommandations pour l'utilisateur 1 selon l'algorithme item-based

3.2. Algorithme de filtrage collaboratif user-based

Deuxièmement, un algorithme de filtrage collaboratif orienté utilisateurs sera étudié. Pour la suite du travail, nous abrègerons son nom par *algorithme user-based*. Nous utiliserons également la mesure de similarité cosinus afin de calculer les similarités entre toutes paires d'utilisateurs. Pour ce faire, la formule ci-dessous (7) sera appliquée.

$$\text{Cosine sim}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2} \sqrt{\sum_{i \in I_v} r_{vi}^2}} \quad (7)$$

Où $\text{Cosine sim}(u, v)$ correspond à la similarité cosinus calculée entre l'utilisateur u et l'utilisateur v ;

r_{ui} vaut 1 lorsque l'utilisateur u a vu le film i et vaut 0 lorsqu'il ne l'a pas vu ;

I_{uv} comprend l'ensemble des films qui ont été vus à la fois par l'utilisateur u et par l'utilisateur v ;

I_u représente l'ensemble des films qui ont été vus par l'utilisateur u ;

I_v correspond à l'ensemble des films vus par l'utilisateur v .

En reprenant l'exemple de matrice de données mentionné à la Figure 2.3, il nous est possible de déterminer les similarités cosinus entre toutes paires d'utilisateurs. Ces similarités sont reprises dans la Figure 2.7 ci-dessous.

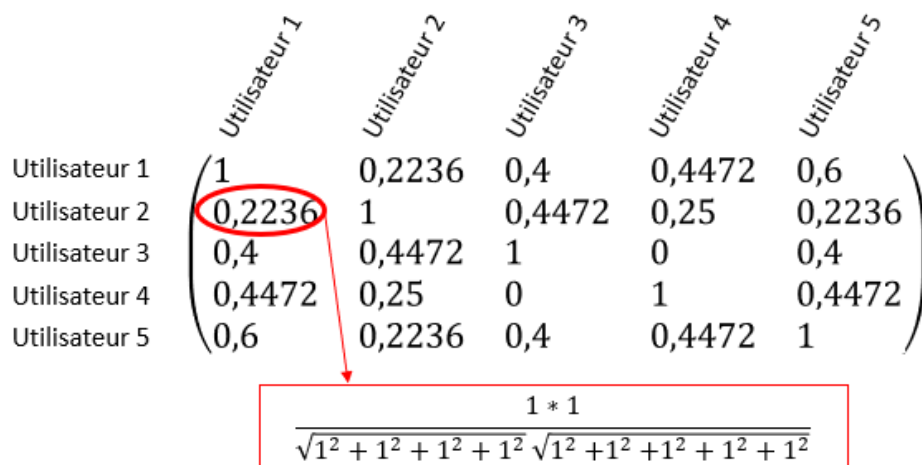


Figure 2.7. Matrice de similarité cosinus entre les utilisateurs

De la même manière que pour le premier algorithme, nous pourrions identifier les k plus proches voisins de l'utilisateur à qui nous souhaitons recommander des films. Si nous reprenons notre exemple et que nous décidons d'identifier les $k=3$ plus proches voisins pour chaque utilisateur, nous obtenons les listes de voisins décrites dans la Figure 2.8. L'ordre dans lequel les voisins sont repris dans chacune des listes a une importance. En effet, nous regarderons d'abord le premier voisin de l'utilisateur pour lui recommander des films.

	Liste des $k=3$ plus proches voisins
Utilisateur 1	Utilisateurs 5, 4 et 3
Utilisateur 2	Utilisateurs 3, 4 et 1
Utilisateur 3	Utilisateurs 2, 1 et 5
Utilisateur 4	Utilisateurs 1, 5 et 2
Utilisateur 5	Utilisateurs 1, 4 et 3

Figure 2.8. Liste des $k=3$ plus proches voisins pour chaque utilisateur

Des films que ses k plus proches voisins ont vus seront ensuite recommandés à l'utilisateur. Tout comme pour l'algorithme précédent, nous ajouterons des films, non vus par l'utilisateur, un à un à la liste de recommandations, jusqu'à ce que celle-ci ait atteint le nombre de recommandations demandées. Tout d'abord, nous ajouterons un à un les films vus par le plus proche voisin de l'utilisateur. Ensuite, nous ajouterons les films vus par le second voisin le plus proche de l'utilisateur. Le même procédé sera appliqué sur les voisins suivants et ce, jusqu'à ce que la liste soit totalement complétée.

Reprenons notre exemple pour constituer la liste de recommandations des utilisateurs selon cet algorithme. Tout comme pour l'algorithme précédent, nous allons imaginer que nous devons fournir une liste de deux recommandations à chaque utilisateur et allons prendre l'exemple de l'utilisateur 1. Nous avons identifié à la Figure 2.9 la méthode à appliquer pour établir la liste de recommandations pour cet utilisateur. Tout d'abord, nous identifions ses voisins et les films vus par ceux-ci. Ensuite, l'algorithme parcourt, un à un, l'ensemble des films vus par le premier voisin (i.e. l'utilisateur 5 dans notre cas) et les ajoute à la liste de recommandations de l'utilisateur 1 uniquement si ce dernier ne les a pas encore vus. Dès lors, nous pouvons remarquer que la liste de recommandations pour l'utilisateur 1 est constituée des films 6 et 9 étant donné que les films 2 et 5 avaient déjà été vus par celui-ci. S'il avait fallu recommander plus de films, l'algorithme aurait regardé s'il pouvait recommander le film 10 et ensuite il serait

passé aux films vus par le second voisin de l'utilisateur 1 (i.e. l'utilisateur 4) et ensuite ceux vus par le troisième voisin de l'utilisateur 1 (i.e. l'utilisateur 3) jusqu'à ce que la liste de recommandations ait atteint le nombre de recommandations demandées. Le même processus sera appliqué pour établir les recommandations pour les autres utilisateurs.

Voisins de l'utilisateur 1	Films vus par ses voisins
Utilisateur 5	Films 2 , 5 , 6 , 9 , 10
Utilisateur 4	Films 3, 5, 6, 8
Utilisateur 3	Films 1, 2, 4, 7, 9

Figure 2.9. Établir les recommandations pour l'utilisateur 1 selon l'algorithme user-based

3.3. Algorithme de filtrage collaboratif intégrant une plus grande diversité au sein des recommandations

Le troisième algorithme que nous étudierons se base également sur le principe du filtrage collaboratif mais il permet d'intégrer une plus grande diversité au sein des recommandations établies. Nous l'abrégerons par *algorithme introduisant plus de diversité*. Nous avons choisi de reprendre un algorithme utilisé dans l'étude de Castagnos et al. (2014). Celui-ci consiste à choisir, parmi un ensemble d'items présélectionnés, un sous-ensemble d'items offrant la plus grande diversité possible. C'est ce sous-ensemble qui constituera la liste de recommandations pour l'utilisateur.

Tout d'abord, il est nécessaire de constituer, pour chaque utilisateur, une liste composée d'un nombre de recommandations plus important que le nombre de recommandations à fournir finalement auprès de l'utilisateur. Cette première liste de films sera constituée sur base d'un algorithme de filtrage collaboratif classique. Nous avons déterminé le nombre d'éléments qui la composeront : il s'agit du quintuple du nombre de recommandations à fournir au final pour l'utilisateur. Cette décision a été prise sur base de l'étude réalisée par Castagnos et al. (2014) où 10 films étaient recommandés parmi une présélection d'un top-50 de films. Cette liste initiale de recommandations sera obtenue par l'algorithme user-based. Cette liste de base ne représente donc pas la liste de recommandations finales qui sera fournie à l'utilisateur.

Pour clarifier cela, reprenons notre exemple de données présenté à la Figure 2.3 et imaginons que nous devons recommander deux films à chaque utilisateur. Le principe de cet algorithme consiste, tout d'abord, à élaborer un ensemble d'items contenant le quintuple du nombre de recommandations demandées. Toutefois, notre base de données utilisée pour l'exemple étant très petite, nous allons nous contenter de constituer un ensemble initial contenant uniquement le double, non pas le quintuple, du nombre de recommandations demandées. Cela revient à dire que nous allons constituer un ensemble initial de 4 items. Cet ensemble initial va être élaboré en faisant appel à l'algorithme user-based décrit au point 3.2. Pour l'utilisateur 1, cet algorithme nous fournira alors l'ensemble d'items suivants : les films 6, 9, 3 et 4.

Le premier film qui appartiendra à la liste de recommandations finales correspond au premier élément de la liste de base constituée précédemment. Dans notre exemple, le premier film qui sera ajouté à la liste de recommandations pour l'utilisateur 1 correspond donc au film 6.

Ensuite, les films seront ajoutés un à un à la liste finale en fonction du film de la liste de base qui propose la plus grande diversité avec les éléments déjà présent dans la liste de recommandations. Des films sont alors ajoutés de cette manière à la liste des films recommandés jusqu'à atteindre le nombre de recommandations requises pour cet utilisateur.

La formule permettant d'obtenir la diversité entre plusieurs éléments est la suivante (8) (Castagnos et al., 2014). Comme cela est inscrit dans la formule, nous avons besoin de connaître la similarité entre toutes paires d'items, nous utiliserons alors les similarités cosinus.

$$Diversity(i_1, \dots, i_n) = \frac{\sum_{j=1..n-1} \sum_{k=j+1..n} (1 - sim(i_j, i_k))}{\frac{n}{2} * (n - 1)} \quad (8)$$

Où $Diversity(i_1, \dots, i_n)$ signifie la diversité de l'ensemble contenant les items i_1 à i_n ;

n correspond au nombre d'items composant l'ensemble dont nous souhaitons calculer la diversité ;

$sim(i_j, i_k)$ représente la similarité entre l'item i_j et l'items i_k .

Poursuivons notre exemple pour plus de clarté. Nous souhaitons donc constituer une liste de deux recommandations pour l'utilisateur 1. Jusqu'à présent, la liste de base, parmi laquelle nous allons choisir les films à recommander pour l'utilisateur 1, contient les films 6, 9, 3 et 4. Nous avons déjà ajouté à la liste de recommandations finales le film 6. L'algorithme va alors sélectionner parmi les films restants dans la liste de base, c'est-à-dire les films 9, 3 et 4, celui

qui permet d'offrir la plus grande diversité avec celui déjà présent dans la liste de recommandations, c'est-à-dire le film 6.

Pour ce faire, l'algorithme calcule, sur base de la formule (8), la diversité entre le film déjà présent dans la liste de recommandations finales et chacun des 3 autres films présents dans la liste de recommandations potentielles. Nous obtenons alors les mesures de diversité présentées dans la Figure 2.10 ci-dessous.

Films repris dans l'ensemble	Diversité de l'ensemble
Films 6 et 9	$Div(i_6, i_9) = \frac{1 - sim(i_6, i_9)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0,5}{1} = 0,5$
Films 6 et 3	$Div(i_6, i_3) = \frac{1 - sim(i_6, i_3)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0,5}{1} = 0,5$
Films 6 et 4	$Div(i_6, i_4) = \frac{1 - sim(i_6, i_4)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0}{1} = 1$

Le film 4 offre la plus grande diversité

Figure 2.10. Mesures de diversité calculées pour différents ensembles

Une fois ces mesures de diversité calculées, il suffit de sélectionner l'item qui permet d'offrir la plus grande diversité avec le(s) film(s) déjà présent(s) dans la liste des recommandations. Dans notre cas, la liste de recommandations n'était composée que d'un seul élément, le film 6 et à travers la Figure 2.10, nous nous apercevons que c'est le film 4 qui permet d'offrir la plus grande diversité avec ce film 6. Dès lors, nous ajoutons le film 4 à la liste des recommandations finales pour l'utilisateur 1.

Étant donné que notre liste de recommandations ne devait contenir que deux éléments, celle-ci est déjà remplie et est constituée des films 6 et 4. Si nous avions dû ajouter d'autres éléments à la liste de recommandations finales, il aurait été nécessaire de calculer la diversité présente entre les films déjà repris dans la liste finale avec chacun des films restants dans la liste potentielle. Nous aurions alors calculé, selon la formule (8), la diversité présente entre les films 6, 4 et 9 ainsi que celle présente entre les films 6, 4 et 3 et nous aurions sélectionné le film 9 ou 3 en fonction de celui qui offrait la plus grande mesure de diversité.

3.4. Algorithme de filtrage collaboratif intégrant une plus grande nouveauté au sein des recommandations

De la même manière que pour l'algorithme précédent, nous avons choisi de réaliser un algorithme qui permet d'intégrer, cette fois-ci non pas une plus grande diversité mais, une plus grande nouveauté au sein des recommandations. Nous l'abrégerons alors par *algorithme introduisant plus de nouveauté*. Dans la littérature, nous avons défini la notion de nouveauté de la manière suivante : un item est nouveau pour un utilisateur lorsque ce dernier n'avait pas connaissance de cet item avant qu'il ne lui soit recommandé (Herlocker et al., 2004 ; Kaminskas & Bridge, 2016). Dès lors, nous allons nous baser sur la popularité des items pour déterminer le degré de nouveauté d'un ensemble de recommandations. Plus un item est populaire, c'est-à-dire plus le nombre d'utilisateurs qui ont vu cet item est important, moins il sera considéré comme nouveau (Herlocker et al., 2004 ; Vargas & Castells, 2011 ; Zhou et al., 2010).

Nous avons choisi d'appliquer le même principe que pour l'algorithme précédent, c'est-à-dire que nous ferons appel, dans un premier temps, à l'algorithme user-based pour nous fournir une liste de recommandations de base, cinq fois plus grande que celle à fournir réellement à l'utilisateur. En reprenant le même exemple que pour l'algorithme introduisant plus de diversité, nous partons du principe que notre ensemble initial pour l'utilisateur 1 est composé des films 6, 9, 3 et 4.

Ensuite, au sein de cette liste, nous allons choisir de recommander à l'utilisateur les films ayant le degré de nouveauté le plus important. Le degré de nouveauté d'un item peut se calculer par la formule suivante (9).

$$Novelty(i) = 1 - pop(i) \quad (9)$$

Où $Novelty(i)$ correspond au degré de nouveauté de l'item i ;

$pop(i)$ représente le degré de popularité de l'item i et est compris entre 0 et 1.

Pour permettre l'application de cette formule, il sera nécessaire de calculer le degré de popularité de l'item étudié. La popularité d'un film peut se mesurer par le nombre d'utilisateurs ayant vu ce film. Mais, pour l'application de la formule présentée ci-dessus (9), il est nécessaire que $pop(i)$ soit compris entre 0 et 1. Dès lors, nous avons choisi de calculer une proportion en rapportant le nombre d'utilisateurs ayant vu un film par rapport au nombre d'utilisateurs compris dans la base de données afin de connaître le degré de popularité de ce film. De cette

manière, lorsqu'un film a été vu par tous les utilisateurs, il aura un degré de popularité maximal, c'est-à-dire de 1. À l'inverse, si un film n'a été vu par aucun utilisateur, son degré de popularité sera nul. (Kaminskas & Bridge, 2016).

Donc, nous calculerons le degré de nouveauté de tous les films et ajouterons à la liste des recommandations finales pour l'utilisateur ceux qui ont le plus grand degré de nouveauté parmi l'ensemble des films issus de la liste de recommandations de base.

Reprenons à présent notre exemple. Notre liste d'items initiale, fournie par l'algorithme user-based, est constituée des films suivants : 6, 9, 3 et 4. Nous devons uniquement fournir deux recommandations à l'utilisateur 1 parmi ces films. Nous allons alors choisir les deux films ayant le plus grand degré de nouveauté. Pour ce faire, il est nécessaire de connaître le degré de popularité de chacun des items. Celui-ci peut être calculé en divisant le nombre d'utilisateurs ayant vu le film par le nombre d'utilisateurs repris dans la base de données. Ces informations sont accessibles dans la Figure 2.3. L'algorithme va alors calculer le degré de popularité ainsi que le degré de nouveauté de chacun des films de la liste potentielle (cf. Figure 2.11)

	Degré de popularité	Degré de nouveauté
Film 6	$\frac{2}{5} = 0,4$	$1 - 0,4 = 0,6$
Film 9	$\frac{2}{5} = 0,4$	$1 - 0,4 = 0,6$
Film 3	$\frac{2}{5} = 0,4$	$1 - 0,4 = 0,6$
Film 4	$\frac{2}{5} = 0,4$	$1 - 0,4 = 0,6$

Figure 2.11. Degrés de popularité et de nouveauté des items de la liste initiale

Cet exemple peut paraître moins parlant car tous les films repris dans la liste initiale ont été chacun vus par deux utilisateurs. Dès lors, les degrés de popularité sont identiques pour chacun de ces films ainsi que, forcément, les degrés de nouveauté. L'algorithme doit normalement choisir les deux films, parmi ces quatre items potentiels, qui offrent les plus grands degrés de nouveauté. Dans ce cas, il choisira simplement les deux premiers, à savoir les films 6 et 9, qui constitueront la liste finale de recommandations pour l'utilisateur 1.

3.5. Algorithme établissant des recommandations aléatoires

Le cinquième algorithme permettra de recommander des films de manière aléatoire aux utilisateurs. Nous le nommerons, par la suite, *algorithme aléatoire*. Pour chaque utilisateur, des films seront alors choisis aléatoirement parmi l'ensemble des films que l'utilisateur n'a pas encore vu et ces films lui seront recommandés.

Par exemple pour l'utilisateur 1, dont les données sont reprises dans la Figure 2.3, les films qu'il n'a pas encore vus sont les films 3, 4, 6, 7 et 9. Dès lors, notre algorithme sélectionnera au hasard deux films parmi ceux-ci afin de les recommander à l'utilisateur 1.

3.6. Algorithme établissant des recommandations populaires

En ce qui concerne le sixième algorithme, il recommandera une liste des films les plus populaires à la période de temps étudiée. Nous le nommerons alors *algorithme populaire*. Ce sont les films qui ont été vus par le plus d'utilisateurs. Évidemment, il faudra également veiller à ne pas recommander à l'utilisateur des films qu'il a déjà vus.

Reprenons à nouveau notre exemple pour une meilleure compréhension. Cet algorithme va tout d'abord calculer le nombre d'utilisateurs ayant vu chacun des films. Sur base des données reprises dans la Figure 2.3, nous obtenons le tableau présenté à la Figure 2.12 regroupant le nombre d'utilisateurs ayant vu chacun des films.

	Nombre d'utilisateurs ayant vu le film
Film 1	2
Film 2	3
Film 3	2
Film 4	2
Film 5	3
Film 6	2
Film 7	2
Film 8	2
Film 9	2
Film 10	2

Figure 2.12. Nombre d'utilisateurs ayant vu chacun des films

Ensuite, l'algorithme va trier les films par ordre de popularité croissante. Dès lors, les films seront triés de la façon suivante : films 2, 5, 1, 3, 4, 6, 7, 8, 9 et 10. Imaginons que nous

souhaitions constituer une liste de deux recommandations pour l'utilisateur 1, nous allons voir si nous pouvons lui recommander le premier film populaire, il s'agit du film 2. Malheureusement, l'utilisateur 1 a déjà vu ce film, nous ne pouvons donc pas le lui recommander. Nous regardons ensuite si nous pouvons lui recommander le second film le plus populaire, il s'agit du film 5 que l'utilisateur a également déjà vu. L'algorithme regarde alors le film populaire suivant et ainsi de suite jusqu'à ce que la liste de recommandations ait atteint le nombre de recommandations demandées, à savoir 2. La liste de recommandations pour l'utilisateur 1 sera alors composée des films 3 et 4.

3.7. Algorithme de filtrage collaboratif avec factorisation matricielle

Le dernier algorithme que nous souhaitons étudier se base sur le filtrage collaboratif, tout comme les quatre premiers algorithmes présentés, mais en utilisant, cette fois-ci, la factorisation matricielle. Nous y ferons référence en le nommant *algorithme de factorisation matricielle*. Ce type d'algorithme avait été brièvement présenté dans la revue de littérature mais nous allons dès à présent l'aborder plus spécifiquement. Le principe de la factorisation matricielle consiste à représenter chaque utilisateur et chaque item par une série de facteurs latents qui les caractérisent. Ces facteurs latents sont obtenus sur base d'un processus itératif.

Plusieurs types de factorisation matricielle existent, mais nous avons choisi de reprendre la forme la plus classique, celle du « Regularized Singular Value Decomposition (RSVD) ». Il s'agit d'un algorithme qui permet d'offrir une qualité importante dans les recommandations fournies (Yu et al., 2014). L'objectif de celui-ci est de pouvoir estimer si un utilisateur verra un certain film ou non. Pour cela, la formule suivante (10) peut être reprise (Koren, 2010 ; Nguyen & Zhu, 2012 ; Paterek, 2007 ; Yu et al., 2014).

$$\hat{r}_{ui} = p_u^T q_i \quad (10)$$

Où \hat{r}_{ui} correspond à l'estimation du fait que l'utilisateur verra un certain film ou non ;

p_u^T représente la transposée du vecteur reprenant l'ensemble des facteurs latents pour l'utilisateur u ;

q_i représente le vecteur reprenant l'ensemble des facteurs latents pour l'item i ;

$p_u^T q_i$ correspond donc au produit interne des deux vecteurs.

La valeur de \hat{r}_{ui} pourra être comprise entre 0 et 1. Plus elle est proche de 1, plus nous pourrions supposer que l'utilisateur regardera le film. À l'inverse, plus la valeur est proche de 0, plus on supposera que l'utilisateur ne sera pas intéressé de regarder le film.

Comme nous venons de le mentionner, un processus itératif est nécessaire pour la réalisation de cet algorithme. À chaque itération, nous calculerons \hat{r}_{ui} selon la formule présentée ci-dessus (10). Ceci ne sera pas calculé pour toutes les paires utilisateur-film de la base de données mais seulement lorsqu'une valeur de 1 est inscrite dans notre matrice (cf. exemple de matrice à la Figure 2.3).

Ensuite, il nous sera possible de calculer l'erreur rencontrée selon la formule suivante (11) (Paterek, 2007) en comparant la valeur réellement rencontrée dans la matrice à celle estimée par notre algorithme. Le but est bien sûr de minimiser cette erreur.

$$err_{ui} = r_{ui} - \hat{r}_{ui} \quad (11)$$

Où err_{ui} correspond à l'erreur calculée pour l'utilisateur u et le film i ;

r_{ui} représente la valeur réellement rencontrée dans notre matrice pour l'utilisateur u et le film i ;

\hat{r}_{ui} représente l'estimation de la valeur pour l'utilisateur u et le film i .

En reprenant notre exemple présenté à la Figure 2.3, nous calculerions alors à chaque itération la valeur de \hat{r}_{ui} pour toutes les paires utilisateur-film de la base de données où l'utilisateur a vu le film considéré. Nous calculerions alors les \hat{r}_{ui} suivants : $\hat{r}_{1,1}$, $\hat{r}_{1,2}$, $\hat{r}_{1,5}$, $\hat{r}_{1,8}$, $\hat{r}_{1,10}$, $\hat{r}_{2,3}$, $\hat{r}_{2,4}$, $\hat{r}_{2,7}$, ..., $\hat{r}_{5,10}$ où le premier indice correspond à l'utilisateur et le second au film. Ces valeurs estimées seraient calculées en appliquant la formule (10). Une fois ces estimations obtenues, elles seraient comparées aux valeurs réelles afin de calculer l'erreur. Imaginons que nous ayons une valeur estimée de $\hat{r}_{1,1}$ qui vaille 0,6. Sachant que la valeur réelle de $r_{1,1}$ vaut 1 (cf. Figure 2.3), l'erreur rencontrée pour la paire utilisateur₁ – film₁ serait $err_{1,1} = 1 - 0,6 = 0,4$. Le même principe de calcul serait appliqué à toutes les paires utilisateur-film estimées.

Après cela, toujours lors de chaque itération, nous calculerons de nouvelles valeurs pour les facteurs latents de chaque utilisateur et de chaque film à travers les formules suivantes (12 et 13) (Paterek, 2007 ; Yu et al., 2014).

$$p_u = p_u + \eta * (err_{ui} q_i - \lambda p_u) \quad (12)$$

$$q_i = q_i + \eta * (err_{ui} p_u - \lambda q_i) \quad (13)$$

Où p_u représente le vecteur reprenant l'ensemble des facteurs latents pour l'utilisateur u ;

q_i représente le vecteur reprenant l'ensemble des facteurs latents pour l'item i ;

η correspond au taux d'apprentissage dont la valeur est à déterminer ;

err_{ui} correspond à l'erreur calculée pour l'utilisateur u et le film i ;

λ est un paramètre dont la valeur est à déterminer.

Afin de pouvoir réaliser ces différentes opérations, il nous est nécessaire de déterminer certains paramètres : le nombre de facteurs latents caractérisant chaque utilisateur et chaque film, le nombre d'itérations que nous réaliserons ainsi que les valeurs de η et λ . Ceci n'étant pas un point d'attention particulier de notre étude, nous avons choisi de reprendre les valeurs utilisées dans l'étude de Yu et al. (2014). Dès lors, nous fixons le nombre de facteurs latents à 15, le nombre d'itérations à 200, le taux d'apprentissage (η) à 0,005 et λ à 0,1.

Au terme des 200 itérations, nous posséderons alors les valeurs définitives des facteurs latents pour chaque utilisateur ainsi que pour chaque film. Afin de constituer les listes de recommandations, nous calculerons, pour chaque utilisateur, les valeurs \hat{r}_{ui} estimées pour chacun des films de la base de données. Ensuite, nous sélectionnerons les films qui ont la plus grande valeur de \hat{r}_{ui} (tout en veillant à ce qu'ils n'aient pas encore été vus par l'utilisateur) afin de constituer la liste de recommandations pour l'utilisateur u . Nous compléterons la liste jusqu'à ce que celle-ci ait atteint le nombre de recommandations demandées.

Reprenons notre exemple et imaginons que les valeurs \hat{r}_{ui} estimées pour l'utilisateur 1 sont celles mentionnées dans la Figure 2.13 ci-dessous.

	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	(0,6	0,7	0,5	0,3	0,8	0,6	0,2	0,7	0,6	0,9)

Figure 2.13. Exemple de valeurs \hat{r}_{ui} estimées pour l'utilisateur 1

Dès lors, nous pouvons ordonner les films par ordre décroissant de ces \hat{r}_{ui} estimés, ce qui nous donne l'ordre suivant : films 10, 5, 2, 8, 1, 6, 9, 3, 4 et 7. L'algorithme regarde s'il peut

recommander le premier film de cette liste, le film 10, mais l'utilisateur 1 l'a déjà vu, nous ne pouvons donc pas le lui recommander. Ensuite, l'algorithme passe au film suivant, le film 5 mais l'utilisateur 1 l'a déjà vu également. L'algorithme regarde donc, dans l'ordre, tous les films de cette liste pour les ajouter à la liste de recommandations pour l'utilisateur 1 jusqu'à ce que l'on ait atteint le nombre de recommandations demandées, à savoir 2. La liste de recommandations pour l'utilisateur 1 sera alors constituée des films 6 et 9.

4. Métriques pour évaluer les recommandations

Afin d'évaluer les recommandations suggérées aux utilisateurs, nous allons nous concentrer sur deux métriques : la diversité et la nouveauté. Ce sont ces métriques qui nous permettront d'étudier l'enfermement potentiel des utilisateurs dans leurs préférences dominantes à travers les systèmes de recommandation. À chaque période de temps et pour chaque algorithme étudié, nous calculerons la diversité ainsi que la nouveauté présentes au sein des recommandations établies par l'algorithme.

De plus, des valeurs de diversité et de nouveauté globales seront calculées à chaque période de temps t . Pour ce faire, nous calculerons la diversité et la nouveauté présentes parmi l'ensemble des films vus au temps t . Grâce à cela, il nous sera possible de comparer les mesures de diversité et de nouveauté fournies par les différents algorithmes avec les mesures de diversité et de nouveauté globales.

Tout comme pour la partie précédente, nous allons baser les explications des deux métriques sur un exemple principal. Imaginons que nous n'ayons que deux périodes de temps à étudier, les films vus par les différents utilisateurs à chacune de ces deux périodes sont repris dans la Figure 2.14 pour la première période de temps et dans la Figure 2.15 pour la seconde période de temps. La matrice regroupant les données des deux périodes se trouve à la Figure 2.16.

Période 1	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	0	1	0	0	1	0	0	0	0	1
Utilisateur 2	0	0	0	0	0	0	1	0	0	1
Utilisateur 3	1	0	0	1	0	0	0	0	0	0
Utilisateur 4	0	0	1	0	1	0	0	0	0	0
Utilisateur 5	0	1	0	0	1	0	0	0	0	1

Figure 2.14. Matrice de données de la première période temporelle

Période 2	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	0	0	0	0	0	0	1	0	0
Utilisateur 2	0	0	1	1	0	0	0	0	0	0
Utilisateur 3	0	1	0	0	0	0	1	0	1	0
Utilisateur 4	0	0	0	0	0	1	0	1	0	0
Utilisateur 5	0	0	0	0	0	1	0	0	1	0

Figure 2.15. Matrice de données de la seconde période temporelle

Périodes 1+2	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	1	0	0	1	0	0	1	0	1
Utilisateur 2	0	0	1	1	0	0	1	0	0	1
Utilisateur 3	1	1	0	1	0	0	1	0	1	0
Utilisateur 4	0	0	1	0	1	1	0	1	0	0
Utilisateur 5	0	1	0	0	1	1	0	0	1	1

Figure 2.16. Matrice de données regroupant les 2 périodes temporelles

4.1. La diversité

La mesure de la diversité a déjà été mentionnée précédemment lorsque nous avons détaillé l'algorithme introduisant plus de diversité (cf. point 3.3 de ce chapitre). C'est cette même métrique que nous allons reprendre afin de mesurer la diversité présente au sein d'une liste de recommandations. Nous utiliserons donc la formule (8), indiquée au point 3.3 de ce chapitre.

Afin de calculer cette métrique, il est nécessaire de connaître la similarité entre toutes paires d'items comme nous pouvons le voir dans la formule à travers $sim(i_j, i_k)$. Dans le but de pouvoir réaliser des comparaisons temporelles entre les résultats obtenus sur cette métrique, il nous semblait primordial de conserver la même matrice de similarité tout au long de notre étude. Dès lors, nous avons choisi que les similarités entre les items correspondraient aux similarités cosinus et seraient obtenues à partir de la matrice représentant notre base de données initiale, toutes divisions temporelles confondues. En appliquant ce concept à notre exemple, nous pourrions calculer les similarités cosinus entre toutes paires d'items sur base de la matrice de données reprise à la Figure 2.16 (i.e. celle regroupant les données de toutes les divisions temporelles). Nous obtiendrions alors les mesures de similarités entre les items présentées dans

la Figure 2.17 ci-dessous. Ce sont ces valeurs de similarités qui seront utilisées pour calculer les différentes mesures de diversité.

	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Film 1	1	0,8165	0	0,5	0,4082	0	0,5	0,5	0,5	0,4082
Film 2	0,8165	1	0	0,4082	0,6667	0,4082	0,4082	0,4082	0,8165	0,6667
Film 3	0	0	1	0,5	0,4082	0,5	0,5	0,5	0	0,4082
Film 4	0,5	0,4082	0,5	1	0	0	1	0	0,5	0,4082
Film 5	0,4082	0,6667	0,4082	0	1	0,8165	0	0,8165	0,4082	0,6667
Film 6	0	0,4082	0,5	0	0,8165	1	0	0,5	0,5	0,4082
Film 7	0,5	0,4082	0,5	1	0	0	1	0	0,5	0,4082
Film 8	0,5	0,4082	0,5	0	0,8165	0,5	0	1	0	0,4082
Film 9	0,5	0,8165	0	0,5	0,4082	0,5	0,5	0	1	0,4082
Film 10	0,4082	0,6667	0,4082	0,4082	0,6667	0,4082	0,4082	0,4082	0,4082	1

Figure 2.17. Similarités cosinus entre les items

En poursuivant notre exemple, nous pourrions imaginer qu'un de nos algorithmes ait établi des recommandations pour tous les utilisateurs, lors de la première période de temps, c'est-à-dire en se basant sur la matrice de données reprise à la Figure 2.14. Nous pouvons alors calculer la diversité présente au sein des listes de recommandations suggérées aux utilisateurs. Dans la Figure 2.18, nous calculons la diversité présente au sein de chaque liste de recommandations établie.

Films recommandés	Diversité des listes de recommandations
Utilisateur 1: Films 1 et 4	$Div(i_1, i_4) = \frac{1 - sim(i_1, i_4)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0,5}{1} = 0,5$
Utilisateur 2: Films 3 et 9	$Div(i_3, i_9) = \frac{1 - sim(i_3, i_9)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0}{1} = 1$
Utilisateur 3: Films 8 et 5	$Div(i_8, i_5) = \frac{1 - sim(i_8, i_5)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0,8165}{1} = 0,1835$
Utilisateur 4: Films 2 et 6	$Div(i_2, i_6) = \frac{1 - sim(i_2, i_6)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0,4082}{1} = 0,5918$
Utilisateur 5: Films 1 et 3	$Div(i_1, i_3) = \frac{1 - sim(i_1, i_3)}{\frac{2}{2} * (2 - 1)} = \frac{1 - 0}{1} = 1$

Figure 2.18. Diversité présente dans chacune des listes de recommandations

Pour donner suite à cela, nous pouvons calculer la diversité moyenne proposée par l'algorithme qui a établi ces recommandations simplement en calculant la moyenne des diversités obtenues dans la Figure 2.18 ci-dessus. Cette diversité moyenne est de 0,6551.

Finalement, il nous reste à comparer cette diversité moyenne par rapport à la diversité globale pour la période de temps étudiée. Pour obtenir cette dernière, il nous faut calculer la diversité présente au sein de l'ensemble des films évalués à la période de temps considérée. Étant donné que nous étudions la première période de temps, nous pouvons identifier dans la Figure 2.14 les films qui ont été vus à cette période, il s'agit des films 1, 2, 3, 4, 5, 7 et 10. En appliquant la formule (8), nous obtenons une diversité globale de 0,5675.

Lors de nos analyses, afin d'étudier l'enfermement potentiel des utilisateurs, nous comparerons l'évolution de la diversité obtenue par un algorithme étudié avec l'évolution de la diversité globale, à travers les différentes périodes de temps. Cela nous permettra alors de distinguer l'impact de l'algorithme étudié sur la diversité, par rapport à l'évolution naturelle de la diversité (i.e. celle issue de la diversité globale) au sein de la base de données.

4.2. La nouveauté

La seconde métrique étudiée sera la nouveauté. La définition de celle-ci a déjà été présentée lors de l'explication de l'algorithme introduisant plus de nouveauté (cf. point 3.4 de ce chapitre). La formule suivante (14) sera appliquée pour déterminer la nouveauté d'un ensemble d'items (Vargas & Castells, 2011).

$$Novelty(i_1, \dots, i_n) = \frac{\sum_{j=1..n} (1 - pop(i_j))}{n} \quad (14)$$

Où $Novelty(i_1, \dots, i_n)$ signifie la nouveauté de l'ensemble contenant les items i_1 à i_n ;

n correspond au nombre d'items composant l'ensemble dont nous souhaitons calculer la nouveauté ;

$pop(i_j)$ représente le degré de popularité de l'item i_j et est compris entre 0 et 1.

Pour rappel, $pop(i_j)$, le degré de popularité d'un item j , correspond au nombre d'utilisateurs ayant vu le film j divisé par le nombre d'utilisateurs repris dans la base de données. Pour ce faire, nous devons choisir quelle base de données, matrice, utiliser à chaque période de temps afin de calculer les degrés de popularité de chaque film. Nous nous retrouvons alors face à deux choix :

1. Le premier consiste à reprendre, tout comme pour le calcul de la métrique de diversité, une seule base de données identique tout au long de l'étude. Celle-ci correspondrait à la base de données initiale toutes divisions temporelles confondues (cf. Figure 2.16) , et permettrait d'avoir des degrés de popularité, pour chaque film, constants tout au long de notre étude. Toutefois, cette première option présente un inconvénient majeur car elle ne considère pas l'évolution de la popularité de certains films au cours du temps. Effectivement, un film populaire à une certaine période ne le sera pas forcément à la période suivante.
2. Notre deuxième option consiste, quant à elle, à reprendre des matrices qui seraient différentes en fonction de la période temporelle étudiée afin de calculer les degrés de popularité de chaque film. Alors, nous reprendrions, à chaque période de temps, une matrice de données équivalente à la division temporelle correspondante de la base de données initiale. Pour notre exemple, cela consisterait à calculer la popularité des films à la première période de temps sur base uniquement des films qui ont été vus lors de cette première période de temps, à savoir les données reprises dans la Figure 2.14. Pour la seconde période de temps nous nous baserions alors uniquement sur les films vus lors de cette seconde période, à savoir les données reprises dans la matrice à la Figure 2.15. Dans ce cas, nous pourrions nous interroger quant à la comparabilité des résultats obtenus sur cette métrique de nouveauté aux différentes périodes temporelles étant donné que la base de données sur laquelle nos résultats sont établis varie au cours du temps.

Enfin, nous avons décidé d'appliquer notre deuxième option concernant le choix de la base de données à utiliser pour calculer cette métrique car elle nous semblait plus proche de la réalité. En effet, la popularité des films évolue au cours du temps et il nous semblait primordial d'intégrer ceci afin d'avoir des résultats réalistes.

Dans le cadre de notre exemple, si nous souhaitions étudier la nouveauté des recommandations fournies par un algorithme lors de la première période de temps, il serait tout d'abord nécessaire de connaître la popularité de tous les films à cette période de temps. Nous obtenons alors les degrés de popularité suivants (cf. Figure 2.19).

	Degré de popularité
Film 1	1/5 = 0,2
Film 2	2/5 = 0,4
Film 3	1/5 = 0,2
Film 4	1/5 = 0,2
Film 5	3/5 = 0,6
Film 6	0/5 = 0
Film 7	1/5 = 0,2
Film 8	0/5 = 0
Film 9	0/5 = 0
Film 10	3/5 = 0,6

Figure 2.19. Degrés de popularité des différents films à la première période de temps.

En appliquant la formule (14), nous calculerons la nouveauté de chacune des listes de recommandations établies auprès des utilisateurs pour ensuite déterminer la nouveauté moyenne présente au sein des listes de recommandations fournies par un certain algorithme. Ensuite, nous pourrions calculer la nouveauté globale présente à la période de temps étudiée pour comparer ces deux résultats de nouveauté.

Appliquons ceci à notre exemple en calculant la nouveauté présente dans chacune des listes de recommandations proposées par un de nos algorithmes pour la première période de temps (cf. Figure 2.20). Cela nous permet alors de calculer la nouveauté moyenne qui est de 0,8.

Films recommandés	Nouveauté des listes de recommandations
Utilisateur 1: Films 1 et 4	$Nov(i_1, i_4) = \frac{(1 - pop(i_1)) + (1 - pop(i_4))}{2} = \frac{(1 - 0,2) + (1 - 0,2)}{2} = 0,8$
Utilisateur 2: Films 3 et 9	$Nov(i_3, i_9) = \frac{(1 - pop(i_3)) + (1 - pop(i_9))}{2} = \frac{(1 - 0,2) + (1 - 0)}{2} = 0,9$
Utilisateur 3: Films 8 et 5	$Nov(i_8, i_5) = \frac{(1 - pop(i_8)) + (1 - pop(i_5))}{2} = \frac{(1 - 0) + (1 - 0,6)}{2} = 0,7$
Utilisateur 4: Films 2 et 6	$Nov(i_2, i_6) = \frac{(1 - pop(i_2)) + (1 - pop(i_6))}{2} = \frac{(1 - 0,4) + (1 - 0)}{2} = 0,8$
Utilisateur 5: Films 1 et 3	$Nov(i_1, i_3) = \frac{(1 - pop(i_1)) + (1 - pop(i_3))}{2} = \frac{(1 - 0,2) + (1 - 0,2)}{2} = 0,8$

Figure 2.20. Nouveauté présente dans chacune des listes de recommandations

Ensuite, nous pouvons comparer cette nouveauté moyenne avec la nouveauté globale présente dans la base de données à la première période de temps. Pour cela, nous calculons, grâce à la formule (14), la nouveauté présente au sein de la liste des films vus à la première période de temps, c'est-à-dire les films 1, 2, 3, 4, 5, 7 et 10. Nous obtenons une mesure de nouveauté globale de 0,6571.

Cette même opération devra alors être répétée à chaque période de temps. Tout comme pour la diversité, il sera nécessaire de comparer l'évolution de la nouveauté moyenne fournie par l'algorithme au cours du temps avec l'évolution naturelle de la nouveauté (i.e. celle obtenue par la nouveauté globale) au sein de la base de données afin d'étudier l'enfermement potentiel des utilisateurs.

5. Simulation du comportement des utilisateurs

Avant d'aborder plus en détail la simulation du comportement des utilisateurs, nous tenions à rappeler que plusieurs périodes de temps seront étudiées au travers de notre analyse. Lors de chaque passage à la période suivante, il sera nécessaire de faire évoluer la base de données sur laquelle nous établissons les recommandations pour les utilisateurs. En effet, pour la première période de temps, nous allons fournir des recommandations aux utilisateurs, en fonction des différents algorithmes, sur base de la matrice correspondant à la première division temporelle de notre base de données initiale. Lors du passage à la période de temps suivante, nous allons ajouter des données dans la matrice permettant aux algorithmes de fournir des recommandations à la période de temps suivante. Ces données ajoutées traduiront alors un certain comportement des utilisateurs.

Dans cette optique, cinq scénarios différents seront étudiés et comparés. Pour rappel, nous avons appris dans la revue de la littérature que la création de bulles filtrantes peut se faire à travers deux composantes : algorithmique et humaine. La composante algorithmique suggère que ce sont les algorithmes de recommandations qui créent les bulles filtrantes dans lesquelles les utilisateurs seraient enfermés. Alors que la composante humaine intègre le fait que les utilisateurs, à travers leurs propres choix, s'enfermeraient eux-mêmes dans une bulle filtrante.

Dès lors, les deux premiers scénarios présenteront ces deux composantes de manière simultanée alors que le troisième ne considèrera que la composante humaine et que les deux derniers analyseront uniquement la composante algorithmique afin de pouvoir étudier l'impact des systèmes de recommandation sur les utilisateurs.

Dans chacun des trois premiers scénarios, un comportement de réponse différent sera étudié : le premier se basera sur le fait que les utilisateurs acceptent toutes les recommandations, le second établira le postulat que les utilisateurs acceptent seulement une partie des recommandations et le dernier considèrera que les utilisateurs n'acceptent aucune recommandation. Ce troisième scénario permet alors d'exclure la composante algorithmique. Les quatrième et cinquième scénarios intègreront uniquement la composante algorithmique et simuleront respectivement le fait que les utilisateurs acceptent toutes les recommandations ou seulement une partie de celles-ci.

Nous allons à présent décrire plus précisément chacun de ces cinq scénarios. Pour ce faire, nous nous baserons sur un exemple commun. Supposons que la base de données initiales ait été

divisée en deux périodes de temps dont les données sont respectivement reprises dans la Figure 2.21 et la Figure 2.22 ci-dessous.

Période 1	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	0	1	0	0	1	0	0	0	0	1
Utilisateur 2	0	0	0	0	0	0	1	0	0	1
Utilisateur 3	1	0	0	1	0	0	0	0	0	0
Utilisateur 4	0	0	1	0	1	0	0	0	0	0
Utilisateur 5	0	1	0	0	1	0	0	0	0	1

Figure 2.21. Matrice de données de la première période de temps

Période 2	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	0	0	0	0	0	0	1	0	0
Utilisateur 2	0	0	1	1	0	0	0	0	0	0
Utilisateur 3	0	1	0	0	0	0	1	0	1	0
Utilisateur 4	0	0	0	0	0	1	0	1	0	0
Utilisateur 5	0	0	0	0	0	1	0	0	1	0

Figure 2.22. Matrice de données de la seconde période de temps

Imaginons qu'un de nos algorithmes ait fourni les recommandations suivantes (cf. Figure 2.23) à chacun des utilisateurs au terme de la première période de temps. Nous allons expliquer dans chacun de nos scénarios comment la matrice de données, sur laquelle l'algorithme se base pour établir ces recommandations, évoluera en fonction du scénario considéré.

	Films recommandés
Utilisateur 1	Films 1 et 4
Utilisateur 2	Films 3 et 9
Utilisateur 3	Films 8 et 5
Utilisateur 4	Films 2 et 6
Utilisateur 5	Films 1 et 3

Figure 2.23. Listes de recommandations fournies par notre algorithme à la première période de temps.

5.1. Composantes humaine et algorithmique & les utilisateurs acceptent toutes les recommandations

À travers cette première simulation, nous considérerons à la fois les composantes humaine et algorithmique de l'apparition potentielle des bulles filtrantes. De plus, nous supposons que les utilisateurs acceptent toutes les recommandations, c'est-à-dire qu'ils vont regarder tous les films qui leur ont été recommandés.

Afin de considérer la composante humaine dans ce premier scénario, à chaque période de temps, nous intégrerons à notre matrice, sur laquelle les algorithmes établissent les recommandations, l'ensemble des données de la période temporelle suivante reprises dans notre base de données initiale. En effet, les informations reprises dans notre base de données initiale représentent les films que les utilisateurs ont vus sans avoir été influencés par un quelconque système de recommandation. Il s'agit donc clairement d'une composante humaine.

Par exemple, lors de la première période de temps, notre matrice correspond à celle reprise à la Figure 2.21. Dans le but d'y intégrer la composante humaine pour la période de temps suivante, nous allons y ajouter l'ensemble des données concernant la seconde division temporelle de la base de données initiale, à savoir les informations reprises dans la Figure 2.22.

Afin d'incorporer également la composante algorithmique, nous allons ajouter d'autres données pour traduire le comportement de l'utilisateur. Plus précisément, nous inscrirons la valeur de 1, dans la matrice, pour tous les films qui avaient été recommandés à un utilisateur et ce, pour chaque utilisateur. De cette manière, nous considérerons que l'utilisateur a vu les films qui lui ont été recommandés.

Pour ce scénario, la matrice sur laquelle se basera l'algorithme pour établir des recommandations à la période de temps suivante sera celle représentée à la Figure 2.24. Elle intègre à la fois les données présentes précédemment dans la matrice (représentées en noir dans la matrice), les données correspondant à la période de temps suivante (i.e. intégration de la composante humaine) (représentées en bleu dans la matrice) ainsi que les données traduisant l'acceptation de toutes les recommandations par les utilisateurs (représentées en rouge dans la matrice).

Scénario 1	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	1	0	1	1	0	0	1	0	1
Utilisateur 2	0	0	1	1	0	0	1	0	1	1
Utilisateur 3	1	1	0	1	1	0	1	1	1	0
Utilisateur 4	0	1	1	0	1	1	0	1	0	0
Utilisateur 5	1	1	1	0	1	1	0	0	1	1

Figure 2.24. Matrice de données intégrant les caractéristiques du scénario 1

5.2. Composantes humaine et algorithmique & les utilisateurs acceptent les recommandations avec une certaine probabilité

Dans ce deuxième cas de figure, nous considérons à nouveau les composantes humaine et algorithmique. Dès lors, nous ajouterons à la matrice l'ensemble des données composant la division temporelle suivante contenue dans la base de données initiale afin d'incorporer la composante humaine.

En ce qui concerne la composante algorithmique, nous nous basons sur le fait que les utilisateurs acceptent les recommandations avec une certaine probabilité p . C'est certainement le cas le plus proche de la réalité. Chaque film recommandé auprès d'un utilisateur a donc une probabilité p d'être vu par celui-ci. Afin d'intégrer cela dans notre matrice, nous y inscrirons une valeur de 1, avec une certaine probabilité p , pour chaque film recommandé auprès de chaque utilisateur. À l'inverse, la valeur attribuée pour ce film et cet utilisateur restera à 0 dans la matrice avec une probabilité $(1-p)$.

Plusieurs valeurs de ce paramètre p pourraient être testées. Mais, une étude a déjà été réalisée dans ce domaine par Szlavik et al. (2011). Après avoir étudié la question, ces chercheurs ont choisi de fixer le pourcentage d'acceptation des recommandations à 25%. Étant donné qu'il ne s'agit pas d'un point d'attention particulier de notre travail, nous avons décidé de reprendre cette même valeur.

Nous avons appliqué ce scénario à notre exemple (cf. Figure 2.25). Les données issues de la première période de temps sont représentées en noir dans la matrice, celles correspondant à la période de temps suivante (i.e. intégration de la composante humaine) sont représentées en bleu et celles traduisant l'acceptation des recommandations par les utilisateurs avec une probabilité

de 25% sont représentées en rouge. Bien sûr, l'acceptation des recommandations se faisant avec une certaine probabilité, il aurait été possible d'obtenir d'autres résultats concernant l'acceptation des recommandations.

Scénario 2	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	1	0	1	1	0	0	1	0	1
Utilisateur 2	0	0	1	1	0	0	1	0	1	1
Utilisateur 3	1	1	0	1	0	0	1	1	1	0
Utilisateur 4	0	0	1	0	1	1	0	1	0	0
Utilisateur 5	0	1	0	0	1	1	0	0	1	1

Figure 2.25. Matrice de données intégrant les caractéristiques du scénario 2

5.3. Composante humaine & les utilisateurs n'acceptent aucune recommandation

Pour ce troisième comportement simulé, nous incorporerons également à notre matrice l'ensemble des données de la division temporelle suivante provenant de la base de données initiale car cela permet d'intégrer la composante humaine.

Nous étudierons alors le fait que les utilisateurs n'acceptent aucune recommandation, c'est-à-dire qu'ils ne tiennent pas compte des recommandations qui leur ont été faites. Dès lors, nous n'ajouterons pas d'autres données supplémentaires à la matrice étant donné que seule la composante humaine est considérée.

L'évolution de notre matrice d'exemple est celle représentée dans la Figure 2.26. Effectivement, elle ne contient que les données associées à la première période de temps (en noir) et les données correspondant à la période de temps suivante (i.e. composante humaine) (en bleu). Elle ne contient donc pas de données supplémentaires qui traduiraient une composante algorithmique.

Scénario 3	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	1	0	0	1	0	0	1	0	1
Utilisateur 2	0	0	1	1	0	0	1	0	0	1
Utilisateur 3	1	1	0	1	0	0	1	0	1	0
Utilisateur 4	0	0	1	0	1	1	0	1	0	0
Utilisateur 5	0	1	0	0	1	1	0	0	1	1

Figure 2.26. Matrice de données intégrant les caractéristiques du scénario 3

5.4. Composante algorithmique & les utilisateurs acceptent toutes les recommandations

Le comportement de réponse des utilisateurs, simulé dans ce scénario, est similaire à celui présenté dans le premier scénario, à savoir que les utilisateurs acceptent toutes les recommandations. Pour cela, nous passerons dans la matrice une valeur de 1 pour chaque film recommandé à un utilisateur et ce, pour tous les utilisateurs.

Toutefois, ce scénario se distingue du premier par le fait qu'il n'intègre que la composante algorithmique. Afin de traduire ceci, nous n'ajouterons pas à la matrice la division temporelle suivante provenant de la base de données initiale étant donné que cela traduirait la composante humaine.

En reprenant notre exemple, nous obtenons la matrice située à la Figure 2.27. Celle-ci contient les données provenant de la première période de temps (en noir) ainsi que les données traduisant l'acceptation des recommandations (en rouge). Par rapport à nos trois premiers scénarios, la composante humaine qui était représentée par les données en bleu n'est pas intégrée.

Scénario 4	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	1	1	0	1	1	0	0	0	0	1
Utilisateur 2	0	0	1	0	0	0	1	0	1	1
Utilisateur 3	1	0	0	1	1	0	0	1	0	0
Utilisateur 4	0	1	1	0	1	1	0	0	0	0
Utilisateur 5	1	1	1	0	1	0	0	0	0	1

Figure 2.27. Matrice de données intégrant les caractéristiques du scénario 4

5.5. Composante algorithmique & les utilisateurs acceptent les recommandations avec une certaine probabilité

Dans ce dernier scénario, nous considérons également uniquement la composante algorithmique. Le comportement simulé des utilisateurs réside, ici, dans le fait que ceux-ci n'acceptent les recommandations qu'avec une certaine probabilité p , comme cela a été présenté dans le deuxième scénario.

En appliquant ce scénario à notre exemple, nous obtenons la matrice reprise la Figure 2.28. Celle-ci contient les données provenant de la première période temporelle (en noir) ainsi que les données traduisant l'acceptation des recommandations avec une probabilité de 25% (en rouge). Vu que ces dernières données ajoutées le sont à travers une décision probabiliste, plusieurs résultats différents auraient pu être possibles également concernant l'acceptation des recommandations.

Scénario 5	Film 1	Film 2	Film 3	Film 4	Film 5	Film 6	Film 7	Film 8	Film 9	Film 10
Utilisateur 1	0	1	0	1	1	0	0	0	0	1
Utilisateur 2	0	0	0	0	0	0	1	0	1	1
Utilisateur 3	1	0	0	1	0	0	0	1	0	0
Utilisateur 4	0	0	1	0	1	0	0	0	0	0
Utilisateur 5	0	1	0	0	1	0	0	0	0	1

Figure 2.28. Matrice de données intégrant les caractéristiques du scénario 5

Conclusion

Pour conclure ce chapitre, revenons brièvement sur les étapes principales qui le composent. Tout d'abord, nous avons choisi d'utiliser, pour notre étude, une base de données MovieLens reprenant des notes attribuées par des utilisateurs à des films. Afin de réaliser une étude temporelle, nous la diviserons en plusieurs parties en fonction du moment où les notes ont été attribuées et conserverons uniquement les utilisateurs présents dans toutes les parties. Chaque partie sera alors stockée sous forme d'une matrice binaire où les lignes correspondent aux utilisateurs et les colonnes aux films. Une valeur de 1 sera indiquée lorsque l'utilisateur a vu le film et une valeur de 0 lorsqu'il ne l'a pas vu.

Alors, nous réaliserons 7 algorithmes capables de fournir des recommandations à tous les utilisateurs sur base d'une matrice de données telle que décrite précédemment. Les algorithmes étudiés seront les suivants : algorithme item-based, algorithme user-based, algorithme introduisant plus de diversité, algorithme introduisant plus de nouveauté, algorithme aléatoire, algorithme populaire et algorithme avec factorisation matricielle.

Au départ, la matrice de données sur laquelle les algorithmes se baseront pour établir des recommandations correspondra à celle de la première division temporelle de la base de données initiale. Ensuite, cette matrice évoluera, à chaque période de temps, en fonction du scénario choisi. Les algorithmes se baseront alors sur cette nouvelle matrice de données pour établir de nouvelles recommandations et ainsi de suite jusqu'à ce qu'on ait parcouru toutes les périodes de temps.

À chaque fois que des recommandations seront établies, nous calculerons la diversité et la nouveauté présentes au sein de celles-ci. Cela nous permettra, par la suite, d'étudier l'évolution de ces mesures sur chacun des algorithmes et de les comparer avec l'évolution naturelle de la base de données dans le but d'étudier l'enfermement potentiel des utilisateurs.

Chapitre 3 : Analyse des résultats

Introduction

Dans ce chapitre, nous allons nous concentrer sur l'analyse des résultats obtenus dans l'objectif de répondre aux différentes questions que nous nous sommes posées tout au long de notre étude. Rappelons tout d'abord la problématique initiale. L'objectif premier de ce mémoire consiste à savoir si les systèmes de recommandation enferment les utilisateurs, au cours du temps, dans leurs préférences dominantes. Dès lors, nous souhaiterions répondre à plusieurs questions qui sont les suivantes :

Est-ce que le nombre de recommandations proposées aux utilisateurs a une influence sur leur enfermement potentiel ?

Comment évoluent la diversité et la nouveauté au cours du temps ?

Le comportement de réponse des utilisateurs a-t-il une influence sur la diversité et la nouveauté présentes au sein des recommandations fournies par l'algorithme ?

En possédant ces éléments de réponse, nous pourrions alors tenter de répondre à la question de recherche au cœur de ce mémoire, à savoir :

Les systèmes de recommandation enferment-ils les utilisateurs, au cours du temps, dans leurs préférences dominantes ?

Dans cet objectif, nous allons tout d'abord aborder brièvement comment nous avons appliqué notre méthodologie à nos données ainsi que les différentes décisions que nous avons dû prendre pour obtenir nos résultats.

Ensuite, nous passerons à la présentation succincte des résultats obtenus pour finalement nous concentrer sur l'analyse de ceux-ci et tenter de répondre à nos différentes interrogations.

1. Expérimentations et prise de décisions

Afin d'appliquer notre méthodologie, nous avons eu recours au langage de programmation Python. Grâce à celui-ci, nous avons codé tous les éléments nécessaires pour la mise en place de notre méthodologie, nous les avons testés et nous avons été amenés à prendre certaines décisions pour, finalement, obtenir nos résultats. Le détail de cette phase d'expérimentations étant assez long, nous l'avons repris en **annexe 1** et nous allons, ici, uniquement détailler les éléments importants qui en sont ressortis. L'entièreté de notre code Python est, quant à lui, disponible en **annexe 2**.

Avant de passer aux décisions que nous avons prises, abordons tout d'abord le code élaboré. Nous avons réalisé différentes méthodes qui ont permis de préparer notre base de données pour être exploitable par la suite. Nous avons également codé chacun des algorithmes présentés précédemment, ceux-ci permettant de fournir une liste de recommandations à chaque utilisateur sur base d'une matrice de données. Ensuite, les différentes méthodes permettant de calculer la diversité et la nouveauté présentes au sein d'une liste de films ont été créées. Après cela, nous avons réalisé le code permettant de simuler chacun des scénarios de réponse des utilisateurs et donc, de modifier la matrice de données (i.e. celle sur laquelle les algorithmes se basent pour fournir des recommandations) en fonction du scénario appliqué. Finalement, nous avons créé une dernière méthode permettant de faire tourner notre code et d'obtenir ainsi les résultats de diversité et de nouveauté présents au sein des recommandations établies par un algorithme et ce, pour chaque paire *période de temps – scénario* appliquée. Ceci a été réalisé trois fois pour chacun de nos algorithmes, à savoir en recommandant respectivement 10, 20 et 100 films aux utilisateurs.

Revenons maintenant sur certaines décisions importantes qui ont été prises lors de cette phase d'expérimentations. La première concerne le nombre de divisions temporelles appliquées. Dans la méthodologie, nous expliquions que la base de données initiale allait être divisée en x parties selon le moment auquel les notes ont été attribuées. Lors de notre phase d'expérimentations, nous avons testé les valeurs de 5, 10 et 20 et nous nous sommes aperçus que seule une division en 5 parties permettait de recenser plusieurs mêmes utilisateurs présents dans toutes les parties. Nous avons alors pris la décision de diviser notre base de données en 5 parties temporelles reprenant, de cette manière, 335 mêmes utilisateurs dans chacune d'elles.

Notre deuxième décision principale se rapporte à l'algorithme introduisant plus de diversité. Pour rappel, nous avons déterminé dans la méthodologie que cet algorithme recommanderait les films apportant le plus de diversité, en les sélectionnant à partir d'un ensemble d'items potentiels composé du quintuple du nombre de recommandations à fournir. Toutefois, nous avons remarqué que le temps d'exécution nécessaire croissait de manière exponentielle avec le nombre d'items potentiels, jusqu'à atteindre plus de 5h d'exécution lorsque 100 recommandations devaient être fournies (i.e. à partir d'un ensemble de 500 items potentiels). Dès lors, nous avons pris la décision de plafonner le nombre d'éléments composant la liste d'items potentiels à 200, nous permettant ainsi de ne pas dépasser 1h30 d'exécution lorsque 100 recommandations doivent être proposées à chaque utilisateur.

La dernière décision importante, que nous avons prise lors de cette phase d'expérimentations, concerne, cette fois, l'algorithme avec factorisation matricielle. Lors de l'application de tests sur cet algorithme, nous avons réalisé que les vecteurs reprenant les facteurs latents, que ce soit ceux caractérisant les utilisateurs ou les films, étaient tous pratiquement identiques. Cela provoquait alors le fait que les mêmes films étaient recommandés à tous les utilisateurs. Après une analyse de la littérature à ce propos, nous avons constaté que cet algorithme était toujours utilisé sur une matrice contenant des données variant sur une certaine échelle, par exemple de 0 à 5, et non pas sur une matrice uniquement binaire, comme c'est notre cas (Koren, 2010 ; Nguyen & Zhu, 2012 ; Paterek, 2007 ; Yuan, Luo & Shang, 2018 ; Yu et al., 2014). Dès lors, nous avons choisi de ne pas étudier cet algorithme dans la suite de nos recherches.

2. Présentation des résultats obtenus

Pour chacun de nos 6 algorithmes et pour chaque nombre de recommandations appliqué (i.e. 10, 20 et 100), nous avons obtenu les résultats de diversité et de nouveauté pour chaque paire *période de temps – scénario* appliquée. Nous avons alors représenté l'ensemble des 36 tableaux de résultats obtenus (i.e. 6 algorithmes, 3 nombres de recommandations et 2 métriques) sous forme de graphes disponibles en **annexe 3 point 1**, en voici un exemple pour l'algorithme item-based à la Figure 3.1.

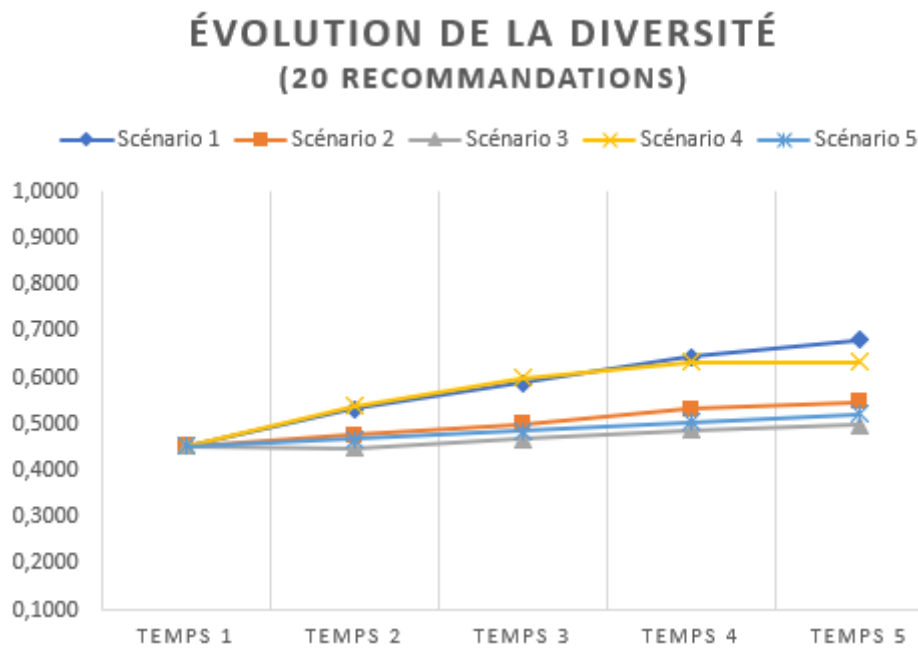


Figure 3.1. Évolution de la diversité, pour chaque scénario, obtenue par l'algorithme item-based lorsque 20 recommandations sont établies

Afin de réaliser de plus amples analyses, nous avons également calculé, pour chaque tableau de résultats, différentes moyennes (cf. Figure 3.2). Les moyennes par période de temps sont reprises dans la colonne à l'extrême droite du tableau, celles par scénario sont indiquées dans la dernière ligne du tableau et la moyenne générale (i.e. toutes périodes de temps et scénarios confondus) est représentée dans la dernière case en bas à droite du tableau.

Ces tableaux de résultats, avec les différentes moyennes, sont repris en **annexe 3 point 2** pour chaque algorithme, chaque nombre de recommandations et chaque métrique.

Diversité	Scénario 1	Scénario 2	Scénario 3	Scénario 4	Scénario 5	Moyennes
Temps 1	0,4531	0,4531	0,4531	0,4531	0,4531	0,4531
Temps 2	0,5329	0,4773	0,4475	0,5393	0,4681	0,4930
Temps 3	0,5883	0,5006	0,4681	0,5990	0,4848	0,5281
Temps 4	0,6453	0,5326	0,4872	0,6329	0,5037	0,5603
Temps 5	0,6811	0,5480	0,4986	0,6341	0,5219	0,5767
Moyennes	0,5802	0,5023	0,4709	0,5717	0,4863	0,5223

Figure 3.2. Résultats de diversité et moyennes de ceux-ci pour l'algorithme item-based lorsque 20 recommandations sont établies

3. Influence du nombre de recommandations

Dans cette section, nous allons nous concentrer sur la première question présentée dans l'introduction, à savoir :

« Est-ce que le nombre de recommandations proposées aux utilisateurs a une influence sur leur enfermement potentiel ? ».

Dans cet objectif, nous avons repris, pour chaque algorithme, chacune des trois moyennes générales (cf. Figure 3.3) de diversité obtenues (i.e. lorsque 10, 20 et 100 recommandations sont réalisées) et les avons représentées dans un graphe (cf. Figure 3.4). Nous reprenons les moyennes générales car l'impact plus spécifique des scénarios et périodes de temps ne nous intéresse pas pour l'instant étant donné qu'il sera analysé plus tard.

Diversité	Scénario 1	Scénario 2	Scénario 3	Scénario 4	Scénario 5	Moyennes
Temps 1	0,4531	0,4531	0,4531	0,4531	0,4531	0,4531
Temps 2	0,5329	0,4773	0,4475	0,5393	0,4681	0,4930
Temps 3	0,5883	0,5006	0,4681	0,5990	0,4848	0,5281
Temps 4	0,6453	0,5326	0,4872	0,6329	0,5037	0,5603
Temps 5	0,6811	0,5480	0,4986	0,6341	0,5219	0,5767
Moyennes	0,5802	0,5023	0,4709	0,5717	0,4863	0,5223

Figure 3.3. Moyenne générale des résultats de diversité obtenus par l'algorithme item-based lorsque 20 recommandations sont établies

Nous appliquons la même procédure pour la nouveauté (cf. Figure 3.5). Ces graphiques nous permettent alors d'avoir une représentation visuelle de l'impact du nombre de recommandations sur les mesures de diversité et de nouveauté.

En plus d'une représentation visuelle, nous souhaitons réaliser une analyse plus précise. Nous avons alors élaboré des tests de Student afin de comparer les résultats de diversité et de nouveauté obtenus lorsque nous avons fixé le nombre de recommandations à 10, 20 et 100 éléments (cf. **annexe 3 point 3**). Cela nous permettra alors de confirmer ou non une potentielle évolution identifiée visuellement à travers les graphiques.

Diversités moyennes générales

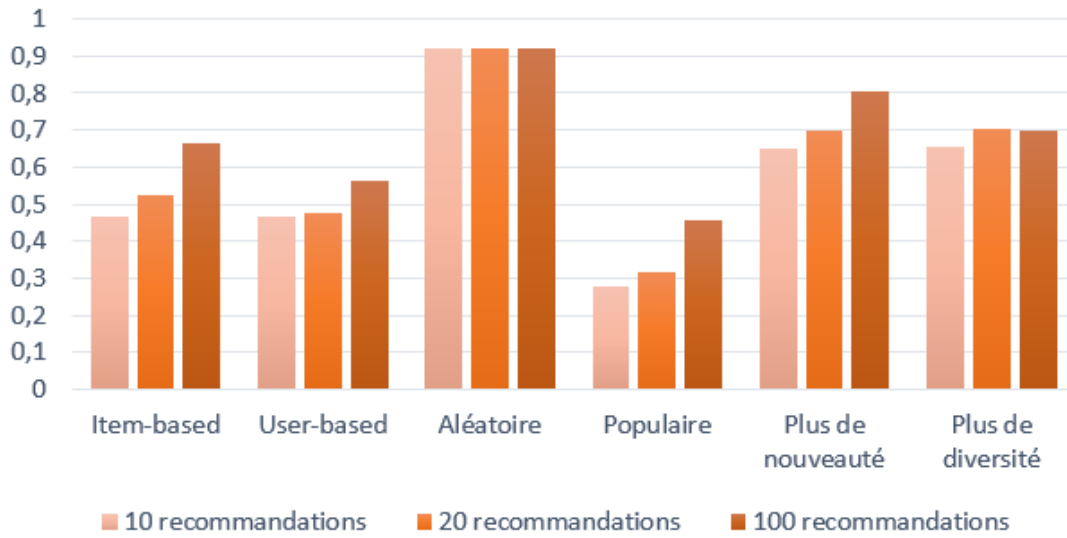


Figure 3.4. Graphique reprenant les diversités moyennes générales obtenues pour chaque algorithme en ayant établi 10, 20 et 100 recommandations

Nouveautés moyennes générales

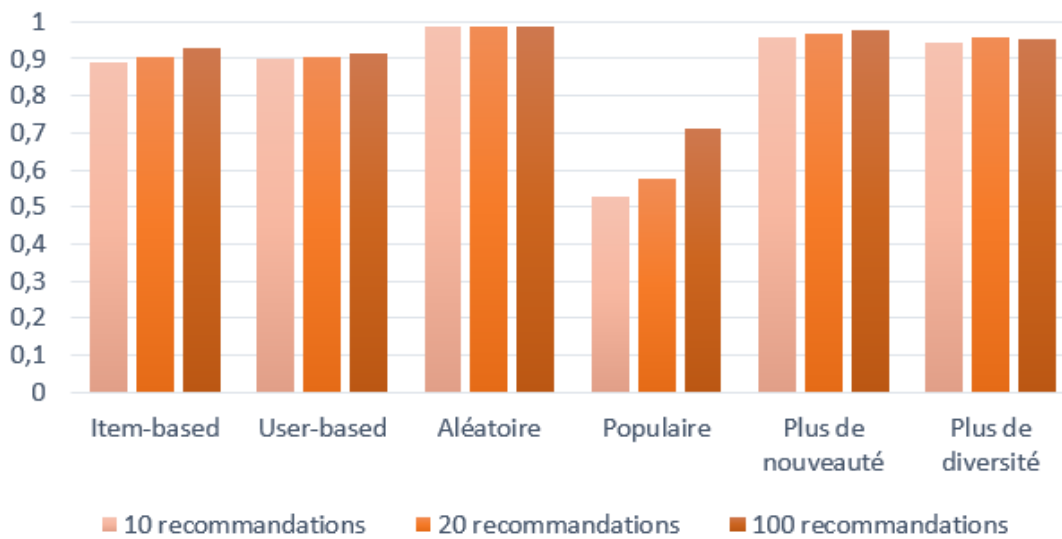


Figure 3.5. Graphique reprenant les nouveautés moyennes générales obtenues pour chaque algorithme en ayant établi 10, 20 et 100 recommandations

À travers ces différents éléments, nous avons pu établir plusieurs constats. Tout d'abord, nous avons remarqué que la diversité augmente généralement avec le nombre de recommandations. Nous pouvons supposer que cette augmentation provient du fait que les films les plus similaires ont été choisis premièrement par les algorithmes. Cela reviendrait à dire que plus le nombre de

recommandations à fournir augmente, plus les algorithmes doivent s'éloigner de ce qu'ils ont tendance à recommander. Toutefois, nous avons pu observer trois exceptions :

- L'algorithme aléatoire présente des niveaux de diversité constants peu importe le nombre de recommandations appliqué. Cela paraît assez évident étant donné qu'il sélectionne de manière aléatoire les recommandations, il n'attribue donc pas plus de pertinence aux premières recommandations par rapport aux suivantes, contrairement aux autres algorithmes.
- L'algorithme introduisant plus de diversité ne présente pas d'hausse de la diversité en passant de 20 à 100 recommandations. Cela peut s'expliquer grâce au fonctionnement de l'algorithme. Celui-ci sélectionne le nombre de recommandations demandées parmi un ensemble d'items en contenant le quintuple, cet ensemble étant toutefois plafonné à 200 éléments. L'algorithme dispose alors d'une marge de manœuvre plus faible pour sélectionner un panel d'items plus diversifiés lorsque 100 recommandations doivent être établies étant donné que l'ensemble d'items potentiels en contient seulement le double et non pas le quintuple. Pour cet algorithme, il ne serait donc pas utile de recommander un maximum d'éléments dans le but d'augmenter la diversité.
- L'algorithme user-based présente une hausse de la diversité entre 10 et 20 recommandations uniquement lorsque les scénarios 1 et 4 sont appliqués, c'est-à-dire lorsque les utilisateurs acceptent toutes les recommandations. Dans ce contexte, il peut être intéressant d'étudier le comportement des utilisateurs afin de choisir, en fonction de celui-ci, le nombre de recommandations à établir.

En ce qui concerne la nouveauté, même si le graphe (cf. Figure 3.5) nous montre légèrement les mêmes tendances que pour la diversité, cela ne peut pas être confirmé par nos tests statistiques. Seul l'algorithme populaire voit la nouveauté de ses recommandations augmenter avec une hausse du nombre de recommandations.

Suite à ces différentes observations, nous conseillerions de recommander un maximum d'éléments aux utilisateurs (hormis dans les trois cas d'exceptions présentés ci-dessus) dans le but d'augmenter la diversité et ainsi le champ des possibles des utilisateurs. Toutefois, il est important de prendre en considération le fait que les utilisateurs ne consultent pas forcément toutes les recommandations qui leur sont proposées. En effet, nous avons appris dans la littérature que le nombre de recommandations excède rarement 20 éléments (Castagnos et al., 2014 ; Nguyen et al., 2014). C'est pourquoi, nous conseillons, finalement, de fixer le nombre de recommandations proposées à chaque utilisateur à 20 films.

4. Évolution temporelle

Nous allons à présent aborder la seconde question que nous nous sommes posée, c'est-à-dire :

« *Comment évoluent la diversité et la nouveauté au cours du temps ?* ».

Afin de réaliser une étude complète sur l'évolution temporelle de nos métriques, nous allons tout d'abord nous concentrer sur les tendances générales observées pour chacun des algorithmes. Ensuite, nous passerons à une analyse détaillée par scénario, ce qui nous permettra également de répondre à notre troisième question présentée dans l'introduction.

4.1. Analyse générale

Dans le but d'étudier l'évolution de nos métriques au cours du temps, nous aurons besoin, pour chaque algorithme, des valeurs de diversité/nouveauté obtenues en moyenne à chaque période de temps (cf. Figure 3.6). Pour l'instant, nous ne considérons pas les différents scénarios car une analyse plus précise par scénario sera réalisée par la suite.

Diversité	Scénario 1	Scénario 2	Scénario 3	Scénario 4	Scénario 5	Moyennes
Temps 1	0,4531	0,4531	0,4531	0,4531	0,4531	0,4531
Temps 2	0,5329	0,4773	0,4475	0,5393	0,4681	0,4930
Temps 3	0,5883	0,5006	0,4681	0,5990	0,4848	0,5281
Temps 4	0,6453	0,5326	0,4872	0,6329	0,5037	0,5603
Temps 5	0,6811	0,5480	0,4986	0,6341	0,5219	0,5767
Moyennes	0,5802	0,5023	0,4709	0,5717	0,4863	0,5223

Figure 3.6. Moyennes, pour chaque période de temps, des mesures de diversité obtenues par l'algorithme item-based lorsque 20 recommandations sont établies

Nous avons alors constitué un graphique par métrique (i.e. diversité et nouveauté) et par nombre de recommandations (i.e. 10, 20 et 100 recommandations) afin de représenter l'évolution de la métrique étudiée pour chacun des algorithmes (cf. **annexe 3 points 4 et 6**).

Suite à cela, il se peut que nous constations une évolution temporelle des métriques étudiées. Mais, il serait alors intéressant d'analyser l'impact réel de l'algorithme sur cette évolution. En effet, les métriques peuvent présenter une évolution naturelle au sein de la base de données. Dans ce cas, l'algorithme ne serait pas responsable de cette évolution. Nous pouvons connaître l'évolution naturelle des métriques au sein de la base de données grâce aux valeurs de diversité/nouveauté globales. Pour rappel, ces mesures globales correspondent aux valeurs de

diversité/nouveauté présentes au sein de tous les films repris dans la base de données à la période de temps considérée. Alors, nous comparerons l'évolution de la diversité/nouveauté obtenue par un certain algorithme par rapport à l'évolution de la diversité/nouveauté globale afin de distinguer l'impact réel de l'algorithme sur cette évolution.

Nous allons tout d'abord commencer par nous focaliser sur l'évolution de la diversité dans le temps pour ensuite analyser celle de la nouveauté.

Étude de l'évolution temporelle de la diversité

Le graphique à la Figure 3.7 provient de l'**annexe 3 point 4** et représente l'évolution de la diversité pour chaque algorithme, lorsque 20 recommandations sont établies. Nous y avons également repris les valeurs de diversité globale de chaque période de temps (cf. courbe rouge dans la Figure 3.7).

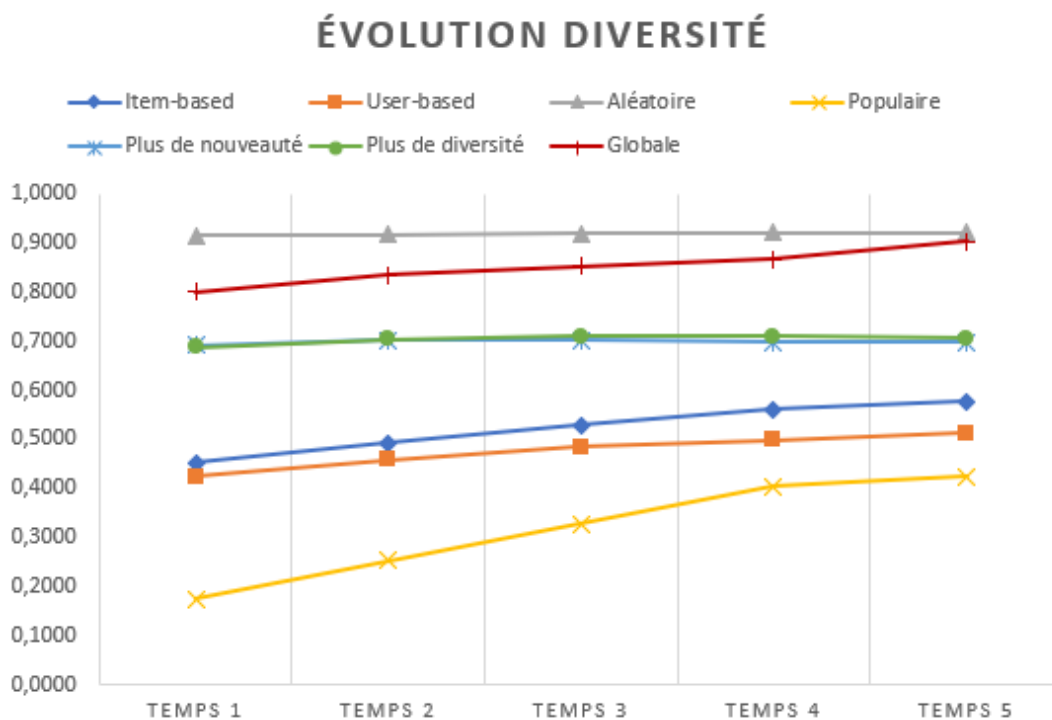


Figure 3.7. Évolution de la diversité dans le temps pour chaque algorithme lorsque 20 recommandations sont établies + Diversité globale

En plus de ce graphe nous permettant d'observer les tendances générales d'évolution pour chacun des algorithmes, nous avons souhaité réaliser une étude plus précise. Nous avons alors calculé les écarts qui peuvent être observés, entre chaque période de temps, sur les mesures de diversité obtenues par un certain algorithme. Ceci a été réalisé pour chaque algorithme et

chaque nombre de recommandations (cf. **annexe 3 point 5**). Par exemple, à la Figure 3.8, nous pouvons voir que la diversité augmente de 0,399 entre la première et la seconde période de temps.

20 recom	Moyennes	Ecart
Temps 1	0,4531	0,0399
Temps 2	0,4930	0,0351
Temps 3	0,5281	0,0322
Temps 4	0,5603	0,0164
Temps 5	0,5767	

$= 0,4930 - 0,4531$

Figure 3.8. Calcul de l'évolution de la diversité pour l'algorithme item-based lorsque 20 recommandations sont établies

Dans cet exemple (cf. Figure 3.8), nous remarquons que la diversité augmente au cours du temps. Toutefois, nous pouvons nous demander quel est l'impact réel de l'algorithme appliqué sur cette évolution de la diversité. Nous allons alors observer l'évolution naturelle de la diversité au sein de la base de données. Comme cela a été mentionné précédemment, l'évolution naturelle de la diversité peut être mesurée à travers l'évolution de la diversité globale, c'est-à-dire celle présente au sein de tous les films repris dans la base de données à chaque période de temps, celle-ci étant indiquée à la Figure 3.9. Pour chaque algorithme, nous allons alors comparer les écarts d'évolution de la diversité obtenue par l'algorithme considéré (cf. Figure 3.8) par rapport aux écarts de la diversité globale (Figure 3.9).

Globale	Diversité	Ecart
Temps 1	0,8007	0,0345
Temps 2	0,8352	0,0179
Temps 3	0,8532	0,0134
Temps 4	0,8666	0,0368
Temps 5	0,9034	

Figure 3.9. Évolution de la diversité globale

En analysant tout ceci, nous ressortons deux tendances principales. La première concerne les algorithmes item-based, user-based et populaire. Pour ces algorithmes, nous remarquons dans le graphique (cf. Figure 3.7) que la diversité, ayant un point de départ plus bas que les autres algorithmes, a tendance à augmenter au cours du temps. Toutefois, nous avons comparé ces

augmentations de la diversité (cf. **annexe 3 point 5**) avec l'évolution naturelle de la diversité au sein de la base de données (cf. Figure 3.9). Alors, nous remarquons que, durant les quatre premières périodes de temps, ces algorithmes augmentent la diversité de manière généralement plus importante que l'évolution naturelle de la diversité au cours du temps. Mais, nous constatons une régression lors du passage de la 4^{ème} période à la 5^{ème} où les algorithmes augmentent moins la diversité que l'augmentation naturelle. Dès lors, nous pouvons conclure que ces trois algorithmes, proposant une diversité plus faible au départ, l'augmentent au cours du temps, jusqu'à atteindre une certaine stagnation à partir de la quatrième période de temps.

Concentrons-nous maintenant sur les trois algorithmes restants : l'algorithme aléatoire, l'algorithme introduisant plus de diversité et celui introduisant plus de nouveauté. En observant à la fois le graphique (cf. Figure 3.7) et les écarts traduisant l'évolution de la diversité (cf. **annexe 3 point 5**), nous remarquons que ces algorithmes proposent, dès la première période de temps, des diversités plus élevées que les trois précédents. Par contre, la diversité n'augmente pas au cours du temps, elle a même parfois tendance à régresser. Nous ne constatons donc pas d'évolution temporelle, en ce qui concerne la diversité, pour ces trois algorithmes.

Étude de l'évolution temporelle de la nouveauté

Le graphique à la Figure 3.10, issu de l'**annexe 3 point 6**, représente l'évolution de la nouveauté pour chaque algorithme lorsque 20 recommandations sont établies. Nous y avons également indiqué les mesures de nouveauté globale (cf. courbe rouge dans la Figure 3.10).

De plus, nous avons calculé, pour chaque algorithme et chaque nombre de recommandations, les écarts de nouveauté présents entre chaque période de temps, traduisant ainsi l'évolution de la nouveauté pour l'algorithme étudié (cf. **annexe 3 point 7**). Nous avons également représenté l'évolution naturelle de la nouveauté sur base de la nouveauté globale présente au sein de la base de données à chaque période de temps (cf. Figure 3.11).

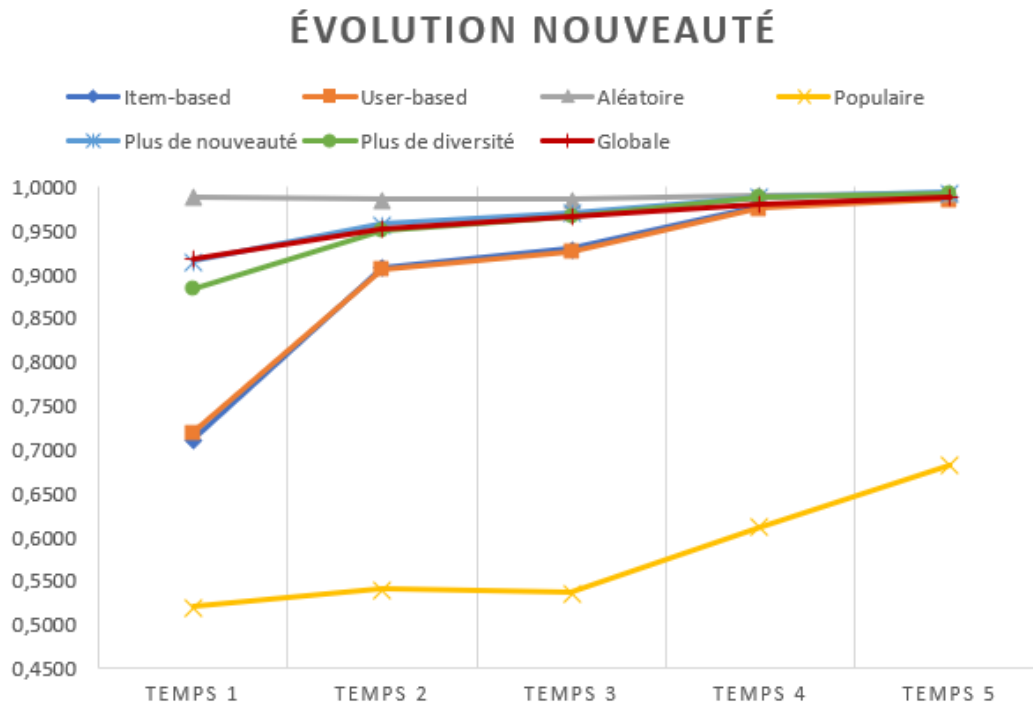


Figure 3.10. Évolution de la nouveauté dans le temps pour chaque algorithme lorsque 20 recommandations sont établies + Nouveauté globale

Globale	Nouveauté	Ecart
Temps 1	0,9186	0,0343
Temps 2	0,9529	0,0125
Temps 3	0,9654	0,0155
Temps 4	0,9809	0,0063
Temps 5	0,9872	

Figure 3.11. Évolution de la nouveauté globale

Pour analyser ces résultats, nous allons tout d'abord nous concentrer sur les quatre algorithmes de filtrage collaboratif (i.e. item-based, user-based, introduisant plus de nouveauté et introduisant plus de diversité). Ces algorithmes proposent une croissance de la nouveauté importante entre la première et la seconde période de temps, c'est-à-dire plus grande que l'évolution de la nouveauté globale aux mêmes périodes. Cela est d'autant plus vrai pour les algorithmes item-based et user-based étant donné qu'ils ont des niveaux de nouveauté plus bas que les deux autres au départ. Ensuite, pour les algorithmes item-based et user-based, les valeurs de nouveauté continuent de croître dans des proportions plus importantes que l'évolution naturelle de la nouveauté. Alors que, pour les deux autres algorithmes, nous pouvons également

observer une croissance de la nouveauté mais plutôt équivalente à l'évolution naturelle de cette métrique. Finalement, ces quatre algorithmes présentent, à la dernière période de temps, des valeurs de nouveauté plutôt équivalentes et légèrement inférieures ou supérieures à la nouveauté globale. En ce qui concerne ces algorithmes de filtrage collaboratif, nous pouvons conclure que la nouveauté évolue bien avec le temps, surtout pour les algorithmes de filtrage collaboratif classiques (i.e. item-based et user-based), mais que cette croissance ralentit également au fur et à mesure. Il est important de mettre en évidence un autre élément : les algorithmes introduisant plus de nouveauté et de diversité sont seulement plus intéressants, en termes de nouveauté, dans les premières périodes de temps. En effet, dans une optique long terme, les algorithmes item-based et user-based pourraient suffire afin d'atteindre un niveau de nouveauté presque équivalent.

Abordons brièvement maintenant l'algorithme aléatoire. Celui-ci propose, sans surprise, un niveau de nouveauté élevé (i.e. constamment supérieure à la nouveauté globale) et constant au cours du temps.

Enfin, nous pouvons identifier que l'algorithme populaire propose toujours une nouveauté bien inférieure aux niveaux de nouveauté obtenus par les autres algorithmes. De plus, son comportement d'évolution est assez différent. Effectivement, contrairement aux algorithmes de filtrage collaboratif dont la nouveauté montre une croissance plus importante au départ, ici, la nouveauté ne commence à croître réellement qu'à partir de la 3^{ème} période de temps. Donc, nous pouvons conclure que cet algorithme recommandant des items populaires nécessite plus de temps pour faire augmenter sa nouveauté et que celle-ci reste toujours bien en-deçà des nouveautés obtenues par les autres algorithmes.

4.2. Analyse par scénario

Après avoir réalisé une analyse temporelle pour chacun des algorithmes, nous allons à présent la préciser à travers une étude plus détaillée sur les différents scénarios. Cela nous permettra alors de répondre à notre troisième question de recherche, c'est-à-dire :

« Le comportement de réponse des utilisateurs a-t-il une influence sur la diversité et la nouveauté présentes au sein des recommandations fournies par l'algorithme ? ».

Au cours de notre étude nous avons appliqué 5 scénarios différents traduisant le comportement de réponse des utilisateurs. À présent, nous souhaitons analyser si ces scénarios ont eu un impact sur les résultats de diversité et de nouveauté obtenus avec chacun des algorithmes.

Rappelons tout d'abord comment nous avons construit nos différents scénarios. Les trois premiers intègrent la composante humaine, c'est-à-dire le fait que les utilisateurs choisissent de regarder des films en-dehors des recommandations qui leur ont été faites, alors que les deux derniers ne l'intègrent pas. Dans tous les scénarios, hormis le troisième, nous considérons le fait que les utilisateurs regardent des films qui leur ont été recommandés. Plus précisément, dans les 1^{er} et 4^{ème} scénarios, nous supposons que les utilisateurs regardent tous les films qui leur ont été recommandés ; alors que dans les 2^{ème} et 5^{ème} scénarios, nous considérons qu'ils les regardent avec 25% de chance.

À travers cette analyse, nous allons alors étudier si les composantes humaine et algorithmique ainsi que les comportements d'acceptation des utilisateurs par rapport aux recommandations peuvent avoir un impact sur les résultats en termes de diversité et de nouveauté.

Afin d'avoir une représentation visuelle nous aidant dans cette analyse, nous avons repris en **annexe 3 point 1** l'ensemble des graphes représentant, chacun, les diversités ou les nouveautés obtenues par un algorithme pour un certain nombre de recommandations et pour tous les scénarios à travers le temps. Nous avons donc créé un graphe par tableau de résultats repris en **annexe 3 point 2**.

Grâce à cela nous avons pu établir deux constats principaux que nous allons présenter. Le premier concerne le fait que lorsque les utilisateurs regardent uniquement les films qui leur étaient recommandés, cela peut réduire leur champ des possibles, en fonction de l'algorithme appliqué. À l'inverse, pour d'autres algorithmes, une acceptation totale des recommandations par les utilisateurs peut augmenter leur champ des possibles au cours du temps.

Regarder uniquement les films recommandés peut réduire le champ des possibles

En analysant les données et les graphes des algorithmes introduisant plus de nouveauté et de diversité ainsi que ceux de l'algorithme populaire (cf. **annexe 3 points 1 et 2**), nous avons constaté que les scénarios 4 et 5 présentaient pratiquement systématiquement des diversités inférieures à celles obtenues sur les trois autres scénarios. Nous avons repris un exemple à la Figure 3.12.

Dans certains cas, la diversité présente tout de même une croissance sur ces scénarios mais elle est alors inférieure à celle observée pour les autres scénarios. Dans d'autres cas, la diversité a même tendance à décroître au cours du temps pour ces scénarios 4 et 5.

Cette observation concerne la diversité pour les trois algorithmes abordés et également la nouveauté uniquement pour l’algorithme populaire.

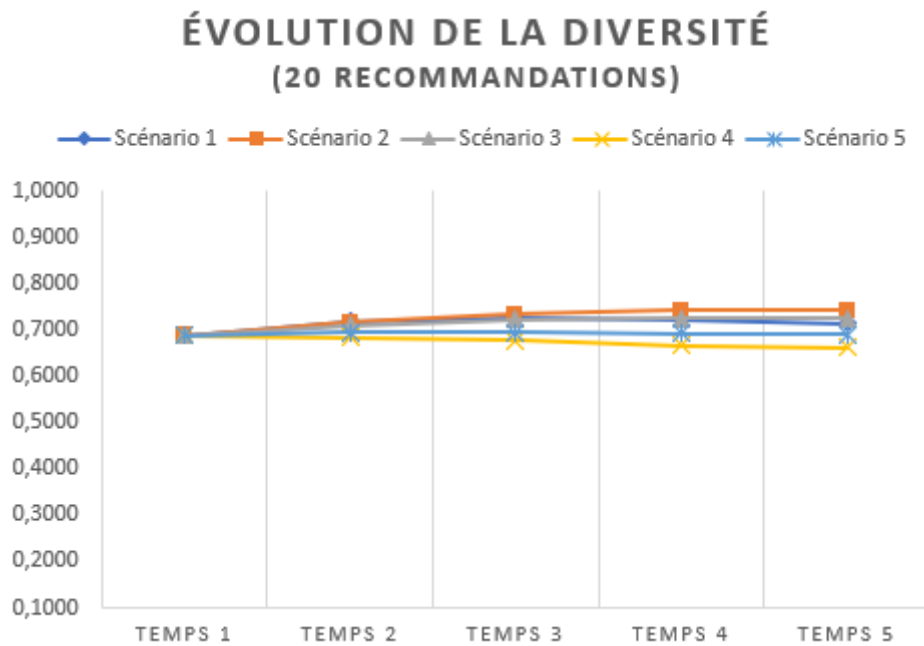


Figure 3.12. *Évolution de la diversité, pour chaque scénario, obtenue par l’algorithme introduisant plus de diversité lorsque 20 recommandations sont établies*

La différence entre les trois premiers scénarios et les deux derniers réside dans le fait que les utilisateurs, dans les trois premiers scénarios, regardent d’autres films, en plus ou à la place de ceux qui leur étaient recommandés alors que dans les deux derniers scénarios, les utilisateurs regardent seulement les films qui leur étaient recommandés. Dès lors, nous pouvons conclure pour ces algorithmes, et seulement pour ceux-ci, que le fait de regarder uniquement une partie ou la totalité des films qui étaient recommandés peut réduire le champ des possibles pour l’utilisateur.

Accepter toutes les recommandations peut augmenter le champ des possibles

Pour les algorithmes de filtrage collaboratif classiques (i.e. item-based et user-based), nous avons pu observer un tout autre constat. En effet, nous avons remarqué que la diversité sur ces algorithmes augmentait, au cours du temps, de façon plus importante pour les scénarios 1 et 4 que pour les autres (cf. **annexe 3 points 1 et 2**). Nous avons repris un exemple concernant l’algorithme item-based à la Figure 3.13. Lors de l’analyse plus générale de l’évolution de la diversité pour ces algorithmes, nous avons remarqué que la diversité augmentait jusqu’à la 4^{ème}

période de temps. Or, en réalisant une analyse plus précise par scénario, nous remarquons non seulement que cette croissance est d'autant plus marquée pour les scénarios 1 et 4, mais également que la croissance a moins tendance à ralentir à partir de la 4^{ème} période de temps par rapport aux autres scénarios.



Figure 3.13. Évolution de la diversité, pour chaque scénario, obtenue par l'algorithme item-based lorsque 10 recommandations sont établies

Ce qui distingue les scénarios 1 et 4 des autres, c'est le comportement de réponse des utilisateurs car nous considérons qu'ils regardent tous les films qui leur ont été recommandés. Pour ces algorithmes de filtrage collaboratif classiques, nous pouvons donc conclure que regarder tous les films recommandés augmente le champ des possibles pour les utilisateurs au cours du temps.

4.3. Conclusion

Nous pouvons maintenant élaborer une réponse aux deux questions que nous nous sommes posées, concernant l'évolution de nos métriques dans le temps ainsi que l'impact du comportement des utilisateurs face aux recommandations. Nous concluons par le fait que l'évolution de la diversité/nouveauté ainsi que l'impact des scénarios varient en fonction de l'algorithme appliqué.

L'algorithme aléatoire ne présente pas d'évolution au cours du temps, peu importe le scénario appliqué, et que cela soit pour la diversité ou pour la nouveauté.

L'algorithme populaire, quant à lui, montre bien une croissance sur les métriques : celle-ci apparaît dès le départ et jusqu'à la 4^{ème} période de temps pour la diversité, alors que l'on peut seulement l'observer à partir de la 3^{ème} période de temps pour la nouveauté. Cette croissance est d'autant plus forte lorsque les utilisateurs regardent également d'autres films que ceux qui leur étaient recommandés. Dans le cas inverse (i.e. lorsqu'ils regardent uniquement les films recommandés), nous pouvons même observer une décroissance de la nouveauté, avant la 3^{ème} période de temps.

Les algorithmes introduisant plus de diversité et de nouveauté présentent des niveaux de diversité plutôt stables à travers le temps alors que la nouveauté augmente légèrement jusqu'à la 4^{ème} période. En ce qui concerne la diversité, nous avons également observé qu'il pouvait y avoir une légère décroissance au cours du temps sur cette métrique lorsque les utilisateurs regardent uniquement les films qui leur étaient recommandés.

Finalement, les algorithmes item-based et user-based proposent une croissance sur les mesures de diversité et de nouveauté, jusqu'à la 4^{ème} période de temps. Cette croissance est accentuée et prolongée lorsque les utilisateurs regardent tous les films qui leur étaient recommandés.

5. Étude de l'enfermement des utilisateurs dans leurs préférences dominantes

Avec tous les éléments de réponse obtenus précédemment, nous allons à présent être en mesure de répondre à notre principale et dernière question de recherche, à savoir :

« Les systèmes de recommandation enferment-ils les utilisateurs, au cours du temps, dans leurs préférences dominantes ? ».

Grâce à la littérature, nous avons appris que la création de bulles filtrantes pouvait se mesurer à travers plusieurs critères principaux liés aux valeurs obtenues sur nos métriques de diversité et de nouveauté. Premièrement, il est important de distinguer l'enfermement dû aux systèmes de recommandation et celui dû aux facteurs humains (Geschke et al., 2019 ; Nikolov et al., 2015). C'est pourquoi, nous comparerons les mesures de diversité et de nouveauté obtenues (cf. **annexe 3 points 5 et 7**) avec les métriques globales (cf. Figure 3.9 et Figure 3.11) reflétant la diversité et la nouveauté présentes naturellement à chaque période de temps dans la base de données. Ensuite, l'apparition de bulles filtrantes peut se faire de manière progressive (Sumpter, 2018). Dans cette optique, plusieurs chercheurs ont étudié l'enfermement à travers l'évolution de la diversité et/ou de la nouveauté (Castagnos et al., 2014 ; Lathia, 2010 ; Nguyen et al., 2014 ; Nikolov et al., 2015). Alors, les conclusions que nous avons pu tirer dans la section précédente, concernant l'analyse temporelle, nous seront également utiles, ici, afin d'étudier l'enfermement. Troisièmement, il sera intéressant de comparer entre eux les systèmes de recommandation étudiés sur les valeurs de diversité/nouveauté obtenues, comme cela a été fait par plusieurs chercheurs (Castagnos et al., 2014 ; Szlavik et al., 2011).

Étant donné que plus d'un critère entre en jeu, il est difficile de déterminer si un système de recommandation enferme oui ou non les utilisateurs dans leurs préférences dominantes. Toutefois, nous allons établir, pour chaque algorithme, une réponse nuancée à cette question en se basant sur les trois critères présentés ci-dessus à savoir : la comparaison avec les mesures de diversités et nouveautés globales, l'évolution temporelle et la comparaison avec les autres algorithmes.

Pour chaque algorithme étudié, nous constituerons également un tableau permettant d'avoir une représentation plus visuelle des éléments qui favorisent un élargissement du champ des possibles pour les utilisateurs avec l'utilisation de l'algorithme considéré.

5.1. Algorithme établissant des recommandations aléatoires

Selon nous, l'algorithme aléatoire respecte les trois critères induisant le fait qu'il n'enferme pas les utilisateurs dans une bulle. Tout d'abord, il propose des résultats de diversité et de nouveauté toujours supérieurs aux diversités et nouveautés globales. Ensuite, concernant l'évolution temporelle, les résultats de diversité et de nouveauté restent stables au cours du temps, peu importe le nombre de recommandations ou le scénario appliqué. Il est assez logique que ceux-ci ne montrent pas de croissance étant donné que les valeurs de départ sont déjà très élevées. N'affichant pas de diminution au cours du temps et présentant des valeurs élevées, nous considérons que cet algorithme respecte le critère concernant l'évolution temporelle. Finalement, nous constatons que ce système de recommandation présente des résultats de diversité et de nouveauté pratiquement toujours supérieurs à ceux obtenus par les autres algorithmes.

Donc, aucun élément particulier ne favorise un élargissement du champ des possibles pour les utilisateurs étant donné que nous observons des valeurs toujours élevées sur nos métriques (cf. Figure 3.14). Nous pouvons alors supposer que ce système de recommandation aléatoire n'enferme pas les utilisateurs dans une bulle filtrante.

Aléatoire	Élargissement du champ des possibles favorisé par
Nombre de recommandations	RIEN car diversité/nouveauté déjà élevées
Période de temps	
Scénario particulier	

Figure 3.14. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme aléatoire

5.2. Algorithme établissant des recommandations populaires

À l'inverse, l'algorithme populaire, est probablement celui qui respecte le moins les critères établis. En effet, il affiche des résultats de diversité et de nouveauté toujours bien inférieurs, non seulement, aux diversités et nouveautés globales, mais également aux résultats obtenus par tous les autres algorithmes étudiés. En ce qui concerne l'évolution des métriques, la diversité a tendance à augmenter, de manière importante, dès la première période de temps et jusqu'à la quatrième, où elle atteint un ralentissement. La nouveauté, quant à elle, commence seulement

à croître réellement à la troisième période de temps et avant cette période, on peut même observer une décroissance lorsque les utilisateurs regardent uniquement les films recommandés.

Cet algorithme est donc, plausiblement, celui qui a le plus tendance à enfermer les utilisateurs dans une bulle par rapport aux autres étudiés étant donné qu’il propose des niveaux de nouveauté et diversité assez bas. Toutefois, nous ne pouvons pas affirmer pleinement qu’il enferme les utilisateurs étant donné qu’il montre, malgré tout, des diversités et nouveautés croissantes au cours du temps, signifiant un élargissement, au fur et à mesure, du champ des possibles pour les utilisateurs (cf. Figure 3.15).

Populaire	Élargissement du champ des possibles favorisé par
Nombre de recommandations	Élevé
Période de temps	Jusqu’à la 4 ^{ème} période de temps pour la diversité À partir de la 3 ^{ème} période de temps pour la nouveauté
Scénario particulier	Les utilisateurs ne regardent pas uniquement les films recommandés

Figure 3.15. Éléments en faveur d’un élargissement du champ des possibles pour les utilisateurs avec l’algorithme populaire

5.3. Algorithme de filtrage collaboratif item-based

Pour cet algorithme item-based ainsi que pour les autres algorithmes de filtrage collaboratif, notre avis sera plus nuancé quant à l’enfermement potentiel des utilisateurs par rapport aux deux premiers algorithmes présentés car il est moins clair d’établir si un critère est respecté ou non.

Abordons tout d’abord notre premier critère, à savoir la comparaison des résultats avec les diversités et nouveautés globales. Pour la diversité, nous constatons clairement qu’elle est toujours inférieure à la diversité globale avec un écart entre les deux groupes assez important. Quant à la nouveauté, les résultats obtenus sont déjà plus proches des nouveautés globales tout en restant inférieurs à celles-ci sauf pour la dernière période de temps lorsque 20 et 100 recommandations sont établies. Afin de savoir s’il existe une réelle distinction entre les résultats obtenus sur nos métriques et les mesures globales, nous avons réalisé des tests de Student (cf. Figure 3.16) comparant les mesures globales avec les résultats obtenus en moyenne par l’algorithme à chaque période de temps. Suite à cela, nous pouvons confirmer que les résultats de diversité obtenus par l’algorithme sont inférieurs aux mesures globales. Par contre, en fixant

un seuil critique à 5%, nous ne rejetons pas le fait que les résultats de nouveauté obtenus par l'algorithme soient à peu près équivalents aux nouveautés globales.

Diversité			
	10 recom	20 recom	100 recom
t-test	0,00000009	0,00000071	0,00000245

Nouveauté			
	10 recom	20 recom	100 recom
t-test	0,0964	0,1025	0,1022

Figure 3.16. t-tests entre les mesures globales et les résultats de diversité/nouveauté obtenus en moyenne à chaque période par l'algorithme item-based

Passons maintenant au second critère, qui concerne l'évolution temporelle. Précédemment, nous avons pu observer que les mesures de diversité et de nouveauté, obtenues par cet algorithme, croissent au cours du temps, particulièrement lorsque les utilisateurs regardent tous les films recommandés, jusqu'à atteindre un certain ralentissement à partir de la quatrième période. Cette croissance nous induit donc une augmentation du champ des possibles pour les utilisateurs mais en ayant toutefois une certaine stagnation sur le long terme.

Notre dernier point consiste à comparer cet algorithme avec les autres algorithmes étudiés. Dans les graphiques repris à l'**annexe 3 point 4**, nous remarquons que l'algorithme item-based fournit des diversités et nouveautés systématiquement inférieures à celles obtenues par trois algorithmes (i.e. ceux introduisant plus de diversité et de nouveauté et celui aléatoire). Toutefois, la nouveauté de l'algorithme item-based augmente dans des proportions plus importantes que ces trois autres algorithmes, qui eux, présentent une nouveauté plutôt constante, pour finalement atteindre des niveaux de nouveauté presque équivalents. Nous pouvons également comparer cet algorithme à celui user-based. Nous remarquons qu'à partir de 20 recommandations établies, l'algorithme item-based propose une diversité et une nouveauté plus importantes que l'algorithme user-based, l'écart se creusant de plus en plus avec l'augmentation du nombre de recommandations.

En conclusion, nous ne pouvons pas déterminer si l'algorithme enferme les utilisateurs dans leurs préférences dominantes ou pas. Mais, il est important de noter que plus on avance dans le temps (avant l'atteinte d'un point de stagnation) et plus le nombre de recommandations augmente, alors plus l'algorithme pourra élargir le champ des possibles pour les utilisateurs surtout lorsqu'ils regardent tous les films recommandés (cf. Figure 3.17).

Item-based	Élargissement du champ des possibles favorisé par
Nombre de recommandations	Élevé
Période de temps	Jusqu'à la 4 ^{ème} période de temps
Scénario particulier	Les utilisateurs regardent tous les films recommandés

Figure 3.17. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme item-based

5.4. Algorithme de filtrage collaboratif user-based

Pour cet algorithme user-based, nous avons commencé par comparer les résultats de diversité et de nouveauté (cf. **annexe 3 points 5 et 7**) obtenus, avec les mesures globales (cf. Figure 3.9 et Figure 3.11). Toujours dans cette optique de comparaison avec les mesures globales, nous avons réalisé des tests de Student (cf. Figure 3.18). Suite à cela, nous tirons les mêmes conclusions que pour l'algorithme item-based. Les résultats de diversités obtenus par l'algorithme user-based sont bien inférieurs aux diversités globales. Quant aux résultats de nouveauté, bien que ceux-ci soient légèrement inférieurs aux nouveautés globales, nous ne pouvons pas rejeter, avec un seuil d'erreur de 5%, le fait que ceux-ci soient à peu près équivalents aux nouveautés globales.

Diversité

	10 recom	20 recom	100 recom
t-test	0,00000015	0,00000005	0,00000076

Nouveauté

	10 recom	20 recom	100 recom
t-test	0,0904	0,0907	0,1013

Figure 3.18. t-tests entre les mesures globales et les résultats de diversité/nouveauté obtenus en moyenne à chaque période par l'algorithme user-based

L'évolution des mesures de diversité et de nouveauté obtenues par l'algorithme user-based étant globalement similaire à l'évolution observée pour l'algorithme item-based, nous en tirons les mêmes conclusions, à savoir que l'algorithme aurait tendance, au cours du temps, à accroître le champ des possibles pour les utilisateurs, particulièrement lorsque ceux-ci regardent tous les films recommandés, tout en atteignant une certaine stagnation sur le long terme.

Il nous reste maintenant à comparer l’algorithme user-based avec l’ensemble des algorithmes étudiés. Tout comme l’algorithme item-based, celui user-based présente des niveaux de diversité et de nouveauté inférieurs à trois algorithmes (i.e. ceux introduisant plus de diversité et de nouveauté et celui aléatoire) mais propose une croissance de la nouveauté plus importante que ceux-ci pour, finalement, atteindre des niveaux de nouveauté presque équivalents.

Cet algorithme semble donc assez similaire en tout point à l’algorithme item-based. Toutefois, nous pouvons les comparer. Lorsque seulement 10 recommandations sont établies, l’algorithme user-based propose une diversité et une nouveauté plus importante que l’algorithme item-based et ce, pendant les trois premières périodes de temps. Par contre, à partir de 20 recommandations, il affiche une diversité et une nouveauté inférieures à celles obtenues avec l’algorithme item-based. Cependant, nous tenons à confirmer ces observations. Nous avons alors réalisé des tests de Student comparant les résultats obtenus par l’algorithme item-based et ceux obtenus par l’algorithme user-based (cf. Figure 3.19). Grâce à ceux-ci, nous pouvons affirmer, avec un taux d’erreur de 0,2%, que l’algorithme item-based présente une diversité plus importante que l’algorithme user-based lorsque 20 ou 100 recommandations sont établies.

t-test	10 recom	20 recom	100 recom
Diversité	0,4415	0,0013	0,0000
Nouveauté	0,1323	0,4537	0,0996

Figure 3.19. t-tests entre les résultats obtenus par les algorithmes item-based et user-based.

Finalement, nous pouvons conclure que cet algorithme enferme plus les utilisateurs dans une bulle par rapport à l’algorithme item-based à partir de 20 recommandations établies. Nous pouvons également noter que plus on avance dans le temps plus cet enfermement diminue, surtout lorsque les utilisateurs regardent tous les films recommandés, mais avec une certaine stagnation à long terme (cf. Figure 3.20).

User-based	Élargissement du champ des possibles favorisé par
Nombre de recommandations	Faible
Période de temps	Jusqu’à la 4 ^{ème} période de temps
Scénario particulier	Les utilisateurs regardent tous les films recommandés

Figure 3.20. Éléments en faveur d’un élargissement du champ des possibles pour les utilisateurs avec l’algorithme user-based

5.5. Algorithme de filtrage collaboratif introduisant une plus grande nouveauté au sein des recommandations

Comparons, tout d'abord, les résultats de diversité et de nouveauté obtenus par cet algorithme introduisant plus de nouveauté avec les mesures de diversité et nouveauté globales. Nous remarquons que cet algorithme présente une diversité toujours inférieure à la diversité globale. En ce qui concerne les résultats de nouveauté obtenus, ceux-ci dépassent la nouveauté globale à partir de la 4^{ème} période de temps lorsque 10 recommandations sont établies, à partir de la 2^{ème} période de temps lorsque 20 recommandations sont établies et dès le départ lorsque 100 recommandations sont établies.

En ce qui concerne l'évolution de nos mesures sur cet algorithme, nous avons pu établir les constats suivants : la nouveauté présente une légère évolution jusqu'à la quatrième période de temps alors que la diversité reste généralement stable au cours du temps, voire décroît lorsque les utilisateurs regardent uniquement les films recommandés.

La comparaison avec les autres algorithmes nous donne, finalement, un constat plutôt favorable. En effet, cet algorithme présente des niveaux de nouveauté toujours supérieurs à ceux obtenus par tous les autres algorithmes (sauf celui aléatoire). Il en va pratiquement de même pour la diversité si ce n'est que l'algorithme introduisant plus de diversité présente des niveaux de diversité assez proches lorsque 10 et 20 recommandations sont établies. Afin d'étudier s'il existe une réelle distinction des résultats obtenus par l'algorithme introduisant plus de nouveauté et celui introduisant plus de diversité, nous avons réalisé des tests de Student étudiant la probabilité que nous ne puissions pas admettre qu'un groupe de résultats soit supérieur à l'autre (cf. Figure 3.21).

t-test	10 recom	20 recom	100 recom
Diversité	0,1513	0,0838	0,0001
Nouveauté	0,0972	0,0879	0,0750

Figure 3.21. *t-tests* entre les résultats obtenus par l'algorithme introduisant plus de nouveauté et celui introduisant plus de diversité

Suite à cela, nous pouvons affirmer que les groupes de résultats obtenus par les deux algorithmes sont plutôt équivalents hormis en ce qui concerne la diversité lorsque 100 recommandations sont établies, où l'algorithme introduisant plus de nouveauté présente des résultats de diversité supérieurs à ceux obtenus par l'algorithme introduisant plus de diversité.

Cela peut paraître paradoxal mais, pour rappel, nous avons effectivement constaté lors de l'étude du nombre de recommandations, que la diversité proposée par l'algorithme introduisant plus de diversité, diminuait lorsque l'on passait de 20 à 100 recommandations alors qu'elle continue d'augmenter pour l'algorithme introduisant plus de nouveauté.

En conclusion, nous aurions tendance à dire que cet algorithme enferme moins les utilisateurs dans une bulle, dès le départ, par rapport aux algorithmes de filtrage collaboratif classiques (i.e. item-based et user-based) étant donné qu'il propose des niveaux de diversité et de nouveauté plus élevés et également une nouveauté parfois plus élevée que la nouveauté globale. Toutefois, il est important de noter que cet algorithme n'élargit pas le champ des possibles pour les utilisateurs au cours du temps étant donné qu'il présente une diversité constante et une nouveauté avec une faible croissance (cf. Figure 3.22).

Plus de nouveauté	Élargissement du champ des possibles favorisé par
Nombre de recommandations	Élevé
Période de temps	Rien pour la diversité Faible croissance jusqu'à la 4 ^{ème} période de temps pour la nouveauté
Scénario particulier	Les utilisateurs ne regardent pas uniquement les films recommandés

Figure 3.22. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme introduisant plus de nouveauté

5.6. Algorithme de filtrage collaboratif introduisant une plus grande diversité au sein des recommandations

Cet algorithme introduisant plus de diversité présente globalement les mêmes constats sur nos trois critères que pour l'algorithme introduisant plus de nouveauté. En ce qui concerne la comparaison de ces deux algorithmes, nous avons déjà identifié le fait qu'ils proposent à peu près des mêmes niveaux de diversité et nouveauté hormis lorsque 100 recommandations sont établies, car dans ce cas, l'algorithme introduisant une plus grande diversité affiche une diversité plus faible que l'algorithme introduisant une plus grande nouveauté.

Pour conclure, nous suggèrerions le fait que cet algorithme enferme moins, dès le départ, les utilisateurs que les algorithmes de filtrage collaboratif classiques tout en, cependant, n'élargissant pas réellement le champ des possibles pour les utilisateurs au cours du temps. De

plus, nous ne conseillerions pas l'utilisation de cet algorithme lorsque le nombre de recommandations est important ou lorsque les utilisateurs regardent uniquement les films recommandés (cf. Figure 3.23).

Plus de diversité	Élargissement du champ des possibles favorisé par
Nombre de recommandations	Faible
Période de temps	Rien pour la diversité Faible croissance jusqu'à la 4 ^{ème} période de temps pour la nouveauté
Scénario particulier	Les utilisateurs ne regardent pas uniquement les films recommandés

Figure 3.23. Éléments en faveur d'un élargissement du champ des possibles pour les utilisateurs avec l'algorithme introduisant plus de diversité

Conclusion des analyses

Pour conclure ces analyses, nous mettrions tout d'abord en avant le fait que, de manière générale, plus le **nombre de recommandations augmente**, plus cela a tendance à **élargir le champ des possibles** pour les utilisateurs, bien qu'il existe quelques exceptions que nous allons rappeler ci-dessous. Toutefois, nous avons également appris à travers la littérature qu'il est important de faire attention à ne pas proposer un nombre de recommandations trop grand car les utilisateurs ne les consulteraient pas toutes.

Ensuite, nous nous sommes rendu compte qu'il était compliqué de déterminer avec exactitude si un algorithme enferme les utilisateurs dans une bulle ou pas. Mais, nous avons tout de même pu identifier, pour chaque algorithme, des cas plus favorables quant à l'élargissement du champ des possibles pour les utilisateurs. En comparant les systèmes de recommandation sur les valeurs de diversité et de nouveauté qu'ils proposent, nous avons pu établir un classement des algorithmes indiquant ceux qui ont le plus tendance à enfermer les utilisateurs contre ceux qui, au contraire, élargissent le plus leur champ des possibles (cf. Figure 3.24). Nous allons dès à présent revenir brièvement sur les éléments importants identifiés pour chacun des algorithmes et qui nous ont permis de construire ce classement.

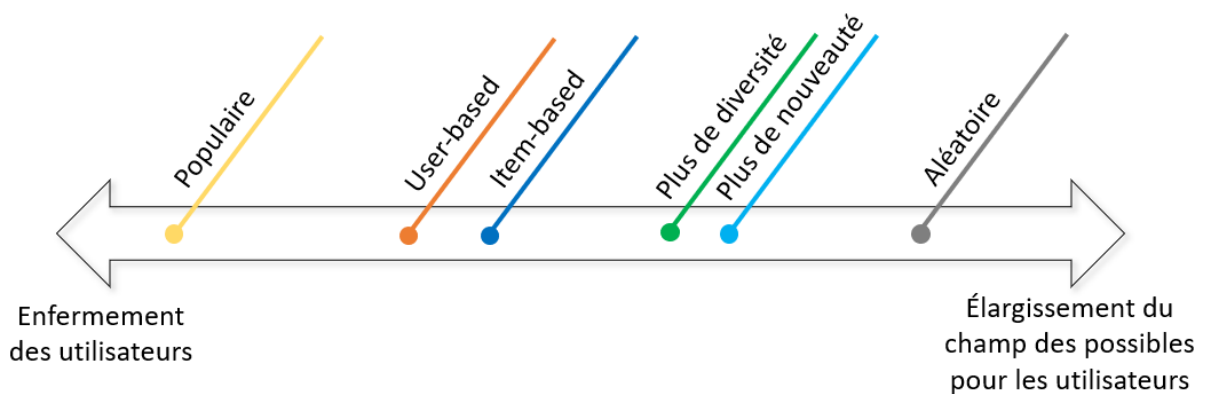


Figure 3.24. Classement des algorithmes quant à l'enfermement des utilisateurs qu'ils provoquent

Nous avons supposé que **l'algorithme aléatoire** est celui qui **élargit le plus le champ des possibles** pour les utilisateurs étant donné qu'il propose des niveaux de diversité et de nouveauté élevés et constants et ce, peu importe le nombre de recommandations, la période de temps ou le scénario appliqués.

À l'inverse, **l'algorithme recommandant des films populaires** est celui qui présente des **diversités et nouveautés les plus faibles**. Nous avons tout de même pu y observer une croissance au cours du temps, ce qui **réduit, au fur et à mesure, la bulle** dans laquelle les utilisateurs sont potentiellement enfermés. De plus, nous avons constaté que, pour élargir le champ des possibles des utilisateurs à travers l'utilisation de cet algorithme, il est **préférable** que ceux-ci ne **consultent pas uniquement les films qui leur sont recommandés**.

En ce qui concerne les **algorithmes de filtrage collaboratif « classiques »** (i.e. item-based et user-based), malgré que ceux-ci aient des niveaux de diversité et de nouveauté toujours plus faibles que les autres algorithmes de filtrage collaboratif étudiés, ils ont montré une croissance sur les métriques à travers le temps. Cela signifie alors que le **champ des possibles est élargi pour les utilisateurs au cours du temps** mais en atteignant, toutefois, un **point de stagnation** à un moment donné. Afin d'éviter un enfermement, il est également **favorable** que les utilisateurs **acceptent toutes les recommandations** fournies par ces algorithmes. Pour distinguer plus précisément ces deux algorithmes, celui **user-based** est légèrement plus efficace, en termes de diversité et de nouveauté, lorsque le **nombre de recommandations est faible** (i.e. 10 recommandations) alors que l'algorithme **item-based** présente une nette amélioration sur ces métriques, par rapport à l'algorithme user-based, lorsque le **nombre de recommandations augmente**.

Quant aux **deux autres algorithmes de filtrage collaboratif étudiés** (i.e. ceux introduisant plus de diversité et plus de nouveauté), nous avons remarqué qu'ils permettaient d'offrir une diversité et une nouveauté plus importantes que les algorithmes de filtrage collaboratif classiques (i.e. item-based et user-based), et ce dès la première période de temps et même avec un faible nombre de recommandations. Cela est intéressant car nous savons alors **qu'il n'est pas nécessaire d'attendre dans le temps ou de devoir proposer un nombre plus important de recommandations** pour élargir le champ des possibles pour les utilisateurs. De plus, nous avons constaté que ces algorithmes proposent un **contenu plus diversifiés lorsque les utilisateurs ne regardent pas uniquement les films qui leur sont recommandés**. En comparant ces deux algorithmes, nous pouvons constater qu'ils proposent des résultats plutôt équivalents, en termes de diversité et de nouveauté, lorsque 10 ou 20 recommandations sont établies. Par contre, nous déconseillerions d'appliquer l'algorithme introduisant plus de diversité lorsque 100 films sont recommandés car la diversité et la nouveauté s'en trouvent réduites et le temps de réalisation nécessaire est croissant.

Conclusion

Bref résumé de l'étude

Tout au long de ce mémoire, nous avons tenté de répondre à la question suivante : « *Les systèmes de recommandation enferment-ils les utilisateurs, au cours du temps, dans leurs préférences dominantes ?* ». Avant d'entamer nos propres recherches, nous avons souhaité étudier les avis et analyses de plusieurs chercheurs en ce qui concerne les systèmes de recommandation dans leur globalité mais également plus précisément l'apparition potentielle de bulles filtrantes et les conséquences qui en découlent.

Ensuite, nous avons établi notre méthodologie nous permettant d'étudier l'apparition d'un potentiel enfermement des utilisateurs par les systèmes de recommandation. Pour ce faire, nous avons choisi d'analyser l'évolution temporelle de la diversité et de la nouveauté obtenues au sein des recommandations fournies par plusieurs algorithmes.

Après cela, notre phase d'expérimentations a été possible grâce à un travail de codage important avec le langage de programmation Python. Cela nous a alors permis d'analyser les résultats de diversité et de nouveauté obtenus. Pour chacun des algorithmes étudiés, nous avons analysé l'impact du nombre de recommandations, de la période de temps et du scénario appliqué (i.e. celui traduisant le comportement de réponse des utilisateurs face aux recommandations) sur nos métriques.

Tout ceci nous a permis de répondre à notre question de recherche. Nous avons conclu qu'il n'était pas possible de confirmer ou d'infirmer un enfermement des utilisateurs, au cours du temps, par les systèmes de recommandation. Toutefois, nous avons pu apporter une réponse nuancée pour chacun des algorithmes étudiés.

L'algorithme établissant des recommandations de manière aléatoire a été identifié comme celui offrant le plus de possibilités aux utilisateurs, peu importe le nombre de recommandations, la période de temps ou leur comportement de réponse. À l'inverse, le système recommandant des films populaires a montré des diversités et nouveautés les plus faibles mais grandissantes au cours du temps. Le même constat avait été établi dans une étude réalisée par Szlavik et al. (2011) où ils avaient pu observer une diversité plus importante au sein de recommandations aléatoires qu'au sein de recommandations populaires.

Pour l'algorithme établissant des recommandations populaires ainsi que pour ceux ayant pour but d'incorporer plus de diversité ou de nouveauté, nous avons également constaté qu'un contenu plus diversifié était recommandé aux utilisateurs lorsque ceux-ci ne regardaient pas uniquement les films qui leur étaient recommandés.

Ces deux algorithmes, ayant pour but d'incorporer plus de diversité et de nouveauté, présentent effectivement des valeurs plus élevées sur ces métriques par rapport à la majorité des algorithmes étudiés et ce, dès la première période de temps. Castagnos et al. (2014) avaient également observé une diversité constante au cours du temps pour l'algorithme intégrant plus de diversité.

En ce qui concerne les algorithmes de filtrage collaboratif classiques (i.e. item-based et user-based), nous avons pu observer un élargissement du champ des possibles pour les utilisateurs au cours du temps, jusqu'à l'atteinte d'un point de stagnation. Cette croissance observée sur ces deux systèmes de recommandation est d'autant plus vérifiée lorsque les utilisateurs acceptent toutes les recommandations qui leur sont proposées. Ce constat de croissance est assez surprenant car il va à l'encontre de ce qui avait été observé par Nguyen et al. (2014) ainsi que par Castagnos et al. (2014) qui avaient respectivement constaté une diversité décroissante et une diversité stable au cours du temps.

Limites et pistes de recherche

Malgré le fait que nous avons essayé de réaliser une étude la plus complète et sérieuse possible, nous pouvons tout de même identifier certaines limites à ce travail, amenant à d'autres recherches qu'il serait intéressant d'accomplir par la suite.

Premièrement, à travers notre question de recherche, nous nous sommes concentrés uniquement sur les métriques permettant d'étudier l'enfermement potentiel des utilisateurs. Toutefois, il se peut qu'un algorithme propose des recommandations diversifiées et nouvelles mais qui ne soient pas pertinentes pour l'utilisateur. Szlavik et al. (2011) avaient, en effet, démontré que des recommandations établies de manière aléatoire présentaient une diversité importante mais diminuaient la qualité de l'expérience ressentie par les utilisateurs. Dans cette optique, il pourrait être intéressant de considérer non seulement les métriques analysant l'enfermement, telles que la diversité et la nouveauté, mais également celles étudiant l'exactitude, telles que celles présentées à ce sujet dans la revue de littérature.

Deuxièmement, nous avons choisi de réaliser notre étude sur une base de données où les items correspondent à des films car c'est généralement cela qui est appliqué dans les recherches sur les systèmes de recommandation. Dès lors, il pourrait être intéressant de réaliser la même étude mais avec une autre base de données afin de voir si nous arriverions aux mêmes conclusions.

La troisième limite que nous avons pu identifier concerne les formules appliquées pour étudier l'enfermement des utilisateurs. Effectivement, il existe de multiples façons de calculer les métriques de diversité et de nouveauté. Nous n'arriverions peut-être pas aux mêmes conclusions si nous avons procédé différemment pour calculer ces métriques. C'est pourquoi, il pourrait être intéressant, dans des recherches futures, d'appliquer d'autres manières de calculer la diversité et la nouveauté.

Quatrièmement, dès le départ de notre étude, nous avons pris la décision de conserver uniquement les données relatives aux utilisateurs présents dans toutes les divisions temporelles de la base de données. Cela a été choisi afin de pouvoir analyser l'évolution des recommandations qui sont proposées à ces utilisateurs. Cependant, il se peut que nous ayons introduit un biais en prenant cette décision car nous ne considérons qu'une partie des utilisateurs. Afin de contrer cela, des recherches futures pourraient s'intéresser à identifier une façon d'intégrer l'apparition et la disparition d'utilisateurs au cours du temps tout en étant capables d'étudier l'évolution des recommandations qui leur sont proposées.

Enfin, la dernière limite que nous avons pu identifier concerne les algorithmes étudiés. En effet, nous avons déjà comparé plusieurs algorithmes de filtrage collaboratif ainsi qu'un algorithme établissant des recommandations aléatoires et un autre recommandant des films populaires. Il s'agit d'algorithmes standards et qui sont les plus étudiés dans la littérature. Nous avons choisi de ne pas analyser les algorithmes basés sur le contenu car d'autres études ont démontré qu'ils avaient plus tendance à enfermer les utilisateurs (Castagnos et al., 2014). Nous avons alors préféré nous concentrer sur les algorithmes pour lesquels cet enfermement potentiel des utilisateurs est plus flou. Toutefois, il pourrait tout de même être intéressant d'étendre notre analyse à d'autres algorithmes dont ceux basés sur le contenu afin de confronter la littérature.

Bibliographie

- Abisheva, A., Garcia, D., & Schweitzer, F. (2016). When the filter bubble bursts: Collective evaluation dynamics in online communities.
- Aggarwal, C. C. (2016). *Recommender systems: The textbook*. Cham: Springer International Publishing AG. doi:10.1007/978-3-319-29659-3
- Alani, M. M., Tawfik, H., Saeed, M., & Anya, O. (2018). *Applications of big data analytics: Trends, issues, and challenges* (1st 2018 ed.). Cham: Springer. doi:10.1007/978-3-319-76472-6
- Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *The Journal of Economic Perspectives*, 31(2), 211-235. doi:10.1257/jep.31.2.211
- Bethke, A. (2016). *Recommender Systems — It's Not All About the Accuracy*. En ligne <https://gab41.lab41.org/recommender-systems-its-not-all-about-the-accuracy-562c7dceeaff>
- Bozdog, E., & Van den Hoven, M. J. (2015). Breaking the filter bubble: Democracy and design. *Ethics and Information Technology*, 2015, 17(4), 249-265. doi:10.1007/s10676-015-9380-y
- Breese, J. S., Heckerman, D., & Kadie, C. (2013). Empirical analysis of predictive algorithms for collaborative filtering.
- Castagnos, S., Brun, A., & Boyer, A. (2014). La diversité : Entre besoin et méfiance dans les systèmes de recommandation. *Revue I3 - Information Interaction Intelligence*,
- Chen, L., & Pu, P. (2008). A cross-cultural user evaluation of product recommender interfaces. Paper presented at the 75-82. doi:10.1145/1454008.1454022
- Cremonesi, P., & Turrin, R. (2010). Time-evolution of IPTV recommender systems. Paper presented at the 105-114. doi:10.1145/1809777.1809801
- Datta, A., Tschantz, M. C., & Datta, A. (2014). Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination.
- Fleder, D., & Hosanagar, K. (2008). Blockbuster culture's next rise or fall: The effect of recommender systems on sales diversity.

- Geschke, D., Lorenz, J., & Holtz, P. (2019). The triple-filter bubble: Using agent-based modelling to test a meta-theoretical framework for the emergence of filter bubbles and echo chambers. *British Journal of Social Psychology*, 58(1), 129-149. doi:10.1111/bjso.12286
- Goel, S., Kumar, R., Kumar, M., & Chopra, V. (2019). An efficient page ranking approach based on vector norms using sNorm(p) algorithm. *Information Processing and Management*, 56(3), 1053-1066. doi:10.1016/j.ipm.2019.02.004
- Hallinan, B., & Striphos, T. (2016). Recommended for you: The netflix prize and the production of algorithmic culture. *New Media & Society*, 18(1), 117-137. doi:10.1177/1461444814538646
- Han, J., & Yamana, H. (2017). A survey on recommendation methods beyond accuracy. *IEICE Transactions on Information and Systems*, E100.D(12), 2931-2944. doi:10.1587/transinf.2017EDR0003
- Helberger, N., Karppinen, K., & D'acunto, L. (2018). Exposure diversity as a design principle for recommender systems. *21(2)*, 191-207. doi:10.1080/1369118x.2016.1271900
- Herlocker, J., Konstan, J., Terveen, L., & Riedl, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5-53. doi:10.1145/963770.963772
- Hosanagar, K., Fleder, D., Lee, D., & Buja, A. (2014). Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation. *Management Science*, 60(4), 805-823. doi:10.1287/mnsc.2013.1808
- Jay F. Nunamaker Jr., Judee K. Burgoon & Justin Scott Giboney Guest Editors. (2016). Special Issue: Information Systems for Deception Detection, *Journal of Management Information Systems*, 33:2, 327-331, doi: 10.1080/07421222.2016.1205928
- Kaminskas, M., & Bridge, D. (2016). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1), 1-42. doi:10.1145/2926720
- Koren, Y. (2010). Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4), 89-97. doi:10.1145/1721654.1721677

- Lathia, N. K. (2010). *Evaluating collaborative filtering over time* UCL (University College London).
- MacKenzie, I., Meyer, C., & Noble, S. (2013). *How retailers can keep up with consumers*. En ligne <https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers>
- Netflix Technology Blog. (2012). *Netflix Recommendations: Beyond the 5 stars (Part 1)*. En ligne <https://medium.com/netflix-techblog/netflix-recommendations-beyond-the-5-stars-part-1-55838468f429>
- Nguyen, J., & Zhu, M. (2012). Content-boosted matrix factorization techniques for recommender systems. doi:10.1002/sam.11184
- Nguyen, T., Hui, P., Harper, F., Terveen, L., & Konstan, J. (2014). Exploring the filter bubble: The effect of using recommender systems on content diversity. Paper presented at the 677-686. doi:10.1145/2566486.2568012
- Nikolov, D., Oliveira, D. F. M., Flammini, A., & Menczer, F. (2015). Measuring online social bubbles. *PeerJ Computer Science*, 1, e38. doi:10.7717/peerj-cs.38
- Pariser, E. (2011). *The filter bubble : what the Internet is hiding from you*. New York: Penguin press.
- Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop* (Vol. 2007, pp. 5-8).
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. Paper presented at the 158-167. doi:10.1145/352871.352887
- Senecal, S., & Nantel, J. (2004). The influence of online product recommendations on consumers' online choices. *Journal of Retailing*, 80(2), 159-169. doi:10.1016/j.jretai.2004.04.001
- Skourletopoulos, G., Mastorakis, G., Mavromoustakis, C. X., Dobre, C., & Pallis, E. (2018). *Mobile big data: A roadmap from models to technologies*. Cham: Springer. doi:10.1007/978-3-319-67925-9

- Sumpter, D. (2018). *Outnumbered: From Facebook and Google to fake news and filter-bubbles - the algorithms that control our lives*. London: Bloomsbury Sigma.
- Sunstein, C. R. (2007). *Republic.com 2.0*. Princeton: Princeton university press.
- Szlavik, Z., Kowalczyk, W. J., & Schut, M. C. (2011). Diversity measurement of recommender systems under different user choice models.
- Tourwé, T. (2012). *Recommender Systems: An Overview of the State-of-the-art*. En ligne <https://www.sirris.be/blog/recommender-systems-overview-state-art>
- Vargas, S., & Castells, P. (2011). *Rank and relevance in novelty and diversity metrics for recommender systems* ACM. doi:10.1145/2043932.2043955
- Wikipédia. (2019). *Entropie de Shannon*. En ligne https://fr.wikipedia.org/wiki/Entropie_de_Shannon
- Wikipédia. (2020). Comma-separated values. En ligne https://fr.wikipedia.org/wiki/Comma-separated_values
- Yuan, Y., Luo, X., & Shang, M. (2018). Effects of preprocessing and training biases in latent factor models for recommender systems. *Neurocomputing*, 275, 2019-2030. doi:10.1016/j.neucom.2017.10.040
- Yu, Y., Wang, C., & Gao, Y. (2014). Attributes coupling based item enhanced matrix factorization technique for recommender systems.
- Zanker, M., Rook, L., & Jannach, D. (2019). Measuring the impact of online personalisation: Past, present and future. *International Journal of Human-Computer Studies*, doi:10.1016/j.ijhcs.2019.06.006
- Zhou, T., Kuscsik, Z., Liu, J., Medo, M., Wakeling, J. R., Zhang, Y., & Parisi, G. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(10), 4511-4515.
- Ziegler, C.-N., Mcnee, S. M., Konstan, J. A., & Lausen, G. (2005). Improving recommendation lists through topic diversification. *Proceedings of the 14th International Conference on World Wide Web - WWW 05*. doi: 10.1145/1060745.1060754