MÉMOIRE

En vue de l'obtention de l'

HABILITATION À DIRIGER DES RECHERCHES DE L'UNIVERSITÉ DE TOULOUSE

Délivrée par : *l'Université Toulouse 1 Capitole (UT1 Capitole)*

Présenté et soutenu le *30/11/2017* **par :**

Elsa NEGRE

Systèmes de recommandation : généricité, évaluation et améliorations

---

JURY

Frédérick BENABEN	Professeur d'Université	Rapporteur
Jérôme DARMONT	Professeur d'Université	Rapporteur
Agnès FRONT	Maître de Conférences, HDR	Rapporteure
Franck RAVAT	Professeur d'Université	Directeur de Recherche
Camille ROSENTHAL-SABROUX	Professeur d'Université	Examinatrice
Chantal SOULÉ-DUPUY	Professeur d'Université	Examinatrice

École doctorale et spécialité :
MITT : Image, Information, Hypermédia

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (IRIT - UMR 5505)

Directeur de Recherche :
Franck RAVAT

Rapporteurs :
Frédérick BENABEN, *Jérôme DARMONT* et *Agnès FRONT*

Remerciements

Je souhaite tout d'abord exprimer ma sincère gratitude envers Camille Rosenthal-Sabroux, professeur à l'Université Paris-Dauphine et membre du LAMSADE, pour m'avoir accueillie au sein de son groupe de recherche et m'avoir offert d'excellentes conditions pour mener à bien mes activités de recherche. Son aide, son soutien et ses précieux conseils ainsi que ses qualités scientifiques et humaines ont contribué à l'épanouissement de ce travail. Je tiens également à lui exprimer toute ma reconnaissance pour la confiance qu'elle m'a accordée sur le plan scientifique mais aussi pédagogique. Enfin, je la remercie pour l'honneur qu'elle me fait en participant au jury.

Je suis très reconnaissante envers :
- Agnès Front, Maître de Conférences HDR à l'Université de Grenoble - Alpes,
- Frédérick Benaben, Professeur à l'IMT Mines Albi-Carmaux,
- Jérôme Darmont, Professeur à l'Université Lumière Lyon 2,

d'avoir été rapporteurs de mon mémoire. Je les remercie vivement pour le travail qu'ils ont consacré à son évaluation.

Je remercie également Chantal Soulé-Dupuy, Professeur à l'Université Toulouse 1 - Capitole qui m'a fait l'honneur de participer au jury.

J'adresse également mes sincères remerciements à Franck Ravat, Professeur à l'Université Toulouse 1 - Capitole. Son aide, son soutien, ses remarques et sa patience ont permis d'améliorer la qualité de ce mémoire. Au delà de son implication dans la rédaction de ce mémoire et de ses compétences scientifiques, il a été présent à de nombreuses étapes importantes de mon parcours universitaire (notamment, Master et Doctorat). Je le remercie pour sa présence, ses précieux conseils, sa persévérance et ses qualités humaines.

Je n'oublie pas les personnes avec qui j'ai pu échanger et/ou collaborer durant ces dernières années, notamment :
- Michel Gründstein, Professeur associé au LAMSADE pour nos discussions souvent animées,
- Marie-Hélène Abel, Professeur à l'Université de Technologie de Compiègne pour son soutien et sa confiance,
- Olivier Teste et Ronan Tournier de l'IRIT pour leur présence (depuis longtemps...),
- Sandro Bimonte de l'IRSTEA Clermont-Ferrand pour son enthousiasme à toute épreuve,
- les membres du groupe SIGECAD (Brice Mayag, Pierre-Emmanuel Arduin, Inès Saad, Kathia Marçal de Oliveira, Thierno Tounkara)
- et les doctorants (Ning Wang, Lynda Atif, Maude Arru, Zahra Ferdousi).

Enfin, je tiens à remercier mes parents, ma famille et mes amis qui, de près ou de loin, m'ont soutenue et encouragée.

Résumé

Nos travaux s'articulent autour de l'extraction et de l'analyse de données issues de sources hétérogènes pour les rendre facilement accessibles et exploitables par les utilisateurs/décideurs. En effet, il devient de plus en plus difficile de savoir quelles sont les données à rechercher et où les trouver lorsque la masse de données/informations s'accroît. Des techniques informatiques existent pour faciliter cette recherche et permettre une extraction pertinente des données/informations. L'une d'entre elles est la recommandation qui guide l'utilisateur lors de son exploration, en cherchant pour lui les informations susceptibles d'être pertinentes.

Un enjeu intéressant est de proposer un système de recommandation capable de s'adapter à différents cas d'applications, avec de bonnes performances du point de vue de l'utilisateur/décideur et palliant certains manques des systèmes de recommandation existants.

Dans le cadre de nos travaux, l'ensemble des données à explorer peut provenir de différents domaines (les environnements de travail collaboratif, les plateformes d'apprentissage en ligne, les entrepôts de données, les villes intelligentes, les systèmes d'alertes précoces, ...) et l'utilisateur à aider peut être un individu isolé ou une entité multiple à visée publique. Conscients que la masse de données/informations à explorer dans de tels cas peut être très importante, complexe et variée, il nous est apparu nécessaire de proposer des systèmes de recommandation appropriés pour y faire face. Nous proposons donc une approche générique de recommandation, en rupture complète avec les travaux existants, que nous instancions pour permettre de recommander soit des éléments, soit des utilisateurs, sous forme de recommandations individuelles ou à visée publique dans des domaines divers et variés.

Puis, nous nous intéressons à l'évaluation des (systèmes de) recommandations. Afin d'assurer la pertinence des recommandations du point de vue de l'utilisateur/décideur, nous proposons des méthodes pour évaluer subjectivement d'une part le système de recommandation et d'autre part les recommandations retournées.

Enfin, malgré de bonnes performances, parfois, les recommandations ne sont pas considérées comme suffisamment pertinentes. Nous proposons donc des techniques pour améliorer les (systèmes de) recommandations. Elles concernent l'amélioration des données d'entrée, le démarrage à froid et l'ajout de données/sources externes (notamment le contexte de l'utilisateur/décideur).

Nos propositions ont été validées par la participation à différents projets ainsi que le co-encadrement de thèses de Doctorat et le suivi de travaux de Master Recherche.

Mots clés : Systèmes de recommandation, Analyse de données, Aide à la décision, Systèmes d'Information.

Abstract

Our work focuses on heterogeneous data extraction and analysis in order to make it easily accessible and exploitable by users/decision-makers. Indeed, it becomes increasingly difficult to know which data to look for and where to find it when the mass of data/information increases. IT techniques exist to facilitate this research and allow relevant extraction of data/information. One of them is the recommendation that guides the user during his/her exploration, seeking for him/her information that may be relevant.

An interesting challenge is to propose a recommender system that can adapt to different applications, with good performances from user point of view and that can overcome some lacks of existing systems. As part of our work, the explored data set can come from different fields (collaborative work environments, massive open online courses, data warehouses, smart cities, early warning systems, ...) and the helped user may be an isolated individual or a multiple public entity. Aware that the mass of data/information to explore in such cases can be very large, complex and varied, it became necessary to propose appropriate recommender systems to cope. We propose a generic framework which is instantiated to recommend either items or users, as individual recommendations or public recommendations in multiple and varied fields.

Then, we are interested in evaluating the recommendations. In order to ensure the relevance of the recommendations from user point of view, we propose methods to subjectively evaluate both the recommender system and the returned recommendations.

Finally, despite good performance, sometimes, the recommendations are not considered sufficiently relevant. We therefore propose methods to enhance recommendations and recommender systems. They relate to the improvement of input data, cold start and the integration of external data/sources (including user context).

Our proposals have been validated by participation in various projects and the co-supervisor of PhD and Master theses.

Keywords : Recommender systems, Data science/analysis, Decision support systems, Information systems.

Sommaire

1 Introduction **7**
- 1.1 La recommandation . . . 8
- 1.2 Plan du mémoire . . . 9

2 Contexte et Problématique **11**
- 2.1 La recommandation . . . 11
- 2.2 Domaines d'applications . . . 12
  - 2.2.1 Entrepôts de données . . . 13
  - 2.2.2 Environnements de travail collaboratif . . . 13
  - 2.2.3 Plateformes d'apprentissage en ligne . . . 15
  - 2.2.4 Les villes numériques intelligentes . . . 15
  - 2.2.5 Systèmes d'alertes précoces . . . 16
  - 2.2.6 Recommandations individuelles ou à visée publique . . . 17
- 2.3 Verrous scientifiques . . . 17
- 2.4 Synthèse . . . 18

3 Définitions et travaux existants **19**
- 3.1 Recommandation *versus* personnalisation . . . 20
  - 3.1.1 Recommandation . . . 20
  - 3.1.2 Personnalisation . . . 20
- 3.2 Profil d'utilisateur . . . 21
- 3.3 Systèmes de recommandation . . . 22
  - 3.3.1 Catégorisation des systèmes de recommandation . . . 23
    - 3.3.1.1 Les approches basées sur le contenu . . . 23
    - 3.3.1.2 Les approches basées sur le filtrage collaboratif . . . 26
    - 3.3.1.3 Les approches hybrides . . . 28
- 3.4 Méthodes/techniques utilisées . . . 29
  - 3.4.1 Modèle d'espace vectoriel . . . 29
  - 3.4.2 Mesures de similarité . . . 30
    - 3.4.2.1 Similarité cosinus . . . 30
    - 3.4.2.2 Coefficient de corrélation de Pearson . . . 31
  - 3.4.3 Réduction de dimensions . . . 31
    - 3.4.3.1 Analyse en composantes principales . . . 32
    - 3.4.3.2 Décomposition en valeurs singulières . . . 32
  - 3.4.4 Classification . . . 33
    - 3.4.4.1 Classification supervisée . . . 33

3.4.4.2 Classification non-supervisée . . . 34
3.4.5 TF-IDF . . . 34
3.5 Evaluation de tels systèmes . . . 35
3.5.1 Jeux de données, densité et erreurs . . . 35
3.5.2 Mesures . . . 36
3.5.2.1 Justesse . . . 36
3.5.2.1.1 Justesse de la prédiction . . . 36
3.5.2.1.2 Justesse de la classification . . . 37
3.5.2.1.3 Justesse d'ordonnancement . . . 38
3.5.2.2 Autres mesures . . . 38
3.6 Des définitions à nos travaux . . . 39

4 Un système générique de recommandation 41
4.1 Système générique de recommandation . . . 42
4.2 Applications . . . 46
4.2.1 Recommandations individuelles . . . 46
4.2.1.1 Recommander des requêtes OLAP . . . 46
4.2.1.2 Recommander des supports intéressants dans les plateformes d'apprentissage numérique . . . 49
4.2.1.3 Recommander des experts dans les environnements de travail collaboratif . . . 56
4.2.2 Recommandations à visée publique . . . 62
4.2.2.1 Recommander des actions à mettre en place dans les villes . . . 62
4.2.2.2 Recommander des actions à mettre en place pour la gestion de crises (système d'alertes précoces) . . . 69
4.3 Bilan . . . 74

5 Evaluer un système de recommandation 77
5.1 Evaluation subjective du système de recommandation . . . 79
5.1.1 Prise en compte des connaissances . . . 79
5.1.2 Perception du système . . . 80
5.1.2.1 Indicateur de perception pour les systèmes d'alertes précoces . . . 82
5.1.2.1.1 Introduction d'un questionnaire . . . 82
5.1.2.1.2 Calcul de l'indicateur . . . 83
5.1.2.2 Expérimentations . . . 84
5.1.2.2.1 Contexte . . . 84
5.1.2.2.2 Protocole . . . 84
5.1.2.2.3 Résultats quantitatifs . . . 85
5.1.2.2.4 Résultats qualitatifs . . . 86
5.2 Evaluation subjective des recommandations . . . 87
5.2.1 Application à OLAP . . . 88
5.2.1.1 Cas d'étude . . . 88
5.2.1.2 Protocole expérimental . . . 89
5.2.1.3 Utilisateurs et requêtes . . . 90
5.2.1.4 Système d'évaluation sémantique . . . 91
5.2.1.5 Expérimentations et résultats . . . 91
5.3 Bilan . . . 93

6 Améliorer la recommandation **95**
6.1 Améliorer les données d'entrée : Structuration de logs/journaux . . . 97
6.1.1 Sous forme de graphe . . . 98
6.1.1.1 Construction du graphe de requêtes . . . 99
6.1.1.2 Construction de l'arbre de recouvrement . . . 100
6.1.1.3 Expérimentations . . . 102
6.1.2 Sous forme de résumé . . . 104
6.1.2.1 QSL . . . 106
6.1.2.2 Mesure de qualité d'un résumé . . . 106
6.1.2.3 Résumé d'un log . . . 106
6.1.2.4 Implémentation et tests . . . 106
6.2 Démarrage à froid . . . 108
6.2.1 Obtenir les squelettes des requêtes du log . . . 109
6.2.2 Prédire les opérations candidates . . . 110
6.2.3 Calculer les recommandations candidates . . . 111
6.2.4 Ordonner les recommandations candidates . . . 112
6.3 Ajouter des données/sources externes : Prise en compte du contexte . . . 113
6.3.1 Contexte . . . 113
6.3.2 Les facteurs de contexte de l'utilisateur . . . 114
6.3.3 Dans la pratique... . . . 117
6.3.3.1 En OLAP . . . 117
6.3.3.1.1 Familles et unités de contexte . . . 117
6.3.3.1.2 Modélisation . . . 118
6.3.3.2 Dans un environnement commercial . . . 121
6.3.3.2.1 Familles et unités de contexte . . . 121
6.3.3.2.2 Modélisation . . . 122
6.3.3.3 En gestion de crises . . . 124
6.3.3.3.1 Comportement . . . 124
6.3.3.3.2 Les facteurs de comportement . . . 125
6.3.3.3.3 Familles et unités de contexte . . . 125
6.3.3.3.4 Modélisation . . . 127
6.4 Bilan . . . 129

7 Outils, Projets, Publications **131**
7.1 Outils . . . 132
7.1.1 RecoOLAP . . . 132
7.1.2 MEMORAe-CRS . . . 134
7.2 Projets et Collaborations . . . 138
7.2.1 Vision synthétique . . . 138
7.2.2 Présentation des projets et/ou Collaborations . . . 138
7.3 Encadrements et Publications . . . 141
7.3.1 Encadrements . . . 141
7.3.2 Publications . . . 142
7.4 Bilan et Synthèse . . . 143

8 Conclusion et Perspectives générales **147**
8.1 Bilan de nos travaux 147
8.2 Perspectives 151

Bibliographie **153**

Chapitre 1

Introduction

Sommaire

1.1 La recommandation 8
1.2 Plan du mémoire 9

Des milliers de décisions sont prises chaque jour. Si certaines d'entre elles semblent aller de soi, d'autres se prennent beaucoup plus difficilement. La prise de décision est un processus cognitif complexe qui vise à sélectionner une action (ou autre) parmi différentes alternatives. Ce processus est basé sur des critères de choix, une analyse des enjeux et des options et conduit à un choix final. Plus concrètement, une des étapes du processus de prise de décision consiste à confronter un ensemble de données/informations en les explorant afin de prendre une décision pertinente.

Il est à noter que, de nos jours, les termes : donnée, information et connaissance, sont souvent utilisés de manière interchangeable. Cependant, une donnée est un fait, une valeur non traitée sans aucune interprétation ou analyse supplémentaire. Par exemple, *18* est une donnée. Une information est une donnée qui a été interprétée de telle sorte qu'elle a un sens pour une personne en particulier. Par exemple, *Max a 18 ans* donne un sens à la donnée *18* et est une information sur l'âge de Max. La connaissance est le résultat d'une réflexion sur les informations en fonction des expériences, des opinions, des idées, des avis, ... d'une personne en particulier. Par exemple, *En France, Max a le droit de voter* est une connaissance.

De nombreux travaux se sont intéressés à cette distinction, citons par exemple, [DP97, JS99, LL01, AC04, Gru12, BMFT16]. Ainsi, les données sont des éléments discrets, bruts, perçus et sélectionnés par un individu avant d'être agrégés, complétés et organisés en information. Par ailleurs, une information, captée par un individu qui se trouve dans un contexte et qui possède ses propres schémas d'interprétation et connaissances tacites antérieures, analysée, et interprétée, devient une connaissance pour cet individu.

Transformer une donnée en information voire en connaissance est un processus relativement trivial pour le cerveau humain. Cependant, lorsque le volume de données et/ou d'informations devient trop important, ce processus se complexifie.

En effet, depuis le début des années 90, le développement des technologies du web et des communications a facilité la génération d'initiatives visant à créer des opportunités pour la communication et le partage de l'information. Dans notre quotidien, nous sommes de plus en plus envahis par des données et des informations. Ce flux est souvent le résultat des évolutions technologiques comme

celles des technologies de l'information et de la communication [1] (TIC) ou de l'Internet des objets [2]. En outre, les possibilités des TIC, qui ont augmenté de manière quasi exponentielle, ont donné naissance à une énorme masse de données à traiter [Bat13]. Finalement, lorsqu'une personne seule doit prendre une décision, celle-ci peut se retrouver submergée par tant de données/informations et ne saura pas quelle décision prendre.

Des techniques informatiques pour faciliter la tâche de l'utilisateur dans son exploration de ce grand volume de données existent, par exemple en Recherche d'Information (RI), en informatique décisionnelle ou en fouille de données (*Data Mining* en anglais). L'exploration de données est un processus de prospection d'informations pertinentes, au sein d'un ensemble de données, destiné à détecter des corrélations cachées ou des informations nouvelles. Les moyens, outils et méthodes de RI cherchent à représenter des documents dans le but d'en récupérer des informations, souvent au moyen d'indexes, les traiter et établir la représentation des documents. En informatique décisionnelle, les moyens, outils et méthodes permettent de collecter, consolider, modéliser et restituer les données afin d'offrir une aide à la décision et une vue d'ensemble des données. Par exemple, les entrepôts de données sont utilisés pour collecter, ordonner et stocker des informations provenant de base de données opérationnelles et fournir ainsi un socle à l'aide à la décision. La fouille de données, quant à elle, a pour objet l'extraction d'un savoir ou d'une connaissance à partir de grandes quantités de données, c'est-à-dire trouver des structures intéressantes selon des critères fixés au préalable, et d'en extraire un maximum de connaissances. Alors que l'informatique décisionnelle permet de constater un fait et de l'expliquer, la fouille de données permet de classer les faits et de les prévoir dans une certaine mesure ou encore de les éclairer.

Malgré leurs intérêts respectifs, leur adéquation à la problématique d'aide à l'exploration des données et leurs performances, les techniques utilisées en RI, informatique décisionnelle et fouille de données demandent à l'utilisateur d'avoir un minimum d'informations/connaissances initiales ainsi que d'avoir une idée de la direction dans laquelle celui-ci veut orienter son exploration des données (en se limitant parfois à certains domaines d'applications).

1.1 La recommandation

Le monde devient de plus en plus numérique et les individus sont touchés/affectés par ces changements. L'infrastructure numérique induit un environnement informationnel qui est aussi imperceptible pour les individus que l'eau pour les poissons [MG11]. Dit autrement, nous pouvons nous sentir noyés/perdus dans cet océan (de plus en plus étendu) de données/informations (tous domaines confondus) [LVC+03] [3]. Il existe une sorte de parallélisme entre les technologies et les

1. Les notions de technologies de l'information et de la communication (TIC) et de nouvelles technologies de l'information et de la communication (NTIC) regroupent les techniques principalement de l'informatique, de l'audiovisuel, des multimédias, d'Internet et des télécommunications qui permettent aux utilisateurs de communiquer, d'accéder aux sources d'information, de stocker, de manipuler, de produire et de transmettre l'information sous toutes les formes : texte, musique, son, image, vidéo et interface graphique interactive [Wik17b].

2. L'Internet des objets (*Internet of Things* - *IoT* en anglais) représente l'extension d'Internet à des choses et à des lieux du monde physique. Alors qu'Internet ne se prolonge habituellement pas au-delà du monde électronique, l'Internet des objets connectés représente les échanges d'informations et de données provenant de dispositifs présents dans le monde réel vers le réseau Internet. L'IoT est en partie responsable d'un accroissement exponentiel du volume de données générées sur le réseau, à l'origine du Big Data [Wik17a].

3. Depuis l'an 2000, *School of Information Management and Systems* de l'Université de Berkeley en Californie, a étudié la quantité de nouvelles informations créées chaque année. Elle prend en compte toute information nouvellement créée, stockée sur différents supports physiques (impression, film magnétique et optique) et vue ou entendue dans

humains : d'une part, les individus utilisent de plus en plus les technologies et deviennent « hyper connectés » et d'autre part, les systèmes (numériques) sont de plus en plus centrés sur l'utilisateur [VK14].

Des techniques informatiques pour faciliter cette recherche ainsi que l'extraction des informations pertinentes sont donc nécessaires, dans ce contexte de déluge de données et d'informations et afin d'aider au mieux l'utilisateur dans son processus de prise de décision. L'une d'entre elles est la recommandation. Le problème est donc de répondre à la question suivante : « Comment guider l'utilisateur dans son exploration des données afin qu'il trouve des informations qui sont pertinentes ? » Pour cela, nous faisons appel aux outils de recommandation afin de trouver le plus rapidement possible des informations pertinentes pour l'utilisateur. Il s'agit de guider l'utilisateur lors de son exploration de la quantité d'informations à sa disposition en cherchant pour lui, les informations qui paraissent pertinentes.
Notons qu'un utilisateur ici, peut être une organisation ou une entreprise. Auquel cas, un système de recommandation peut aider une organisation/entreprise dans ses choix stratégiques en lui indiquant quelles informations sont les plus pertinentes [Neg15b].

Finalement, la recommandation est un champ de recherche en soi. D'un point de vue informatique, dans ce mémoire, ce vaste champ de recherche est restreint aux systèmes de recommandation. Notons que, encore récemment, [JA16], par exemple, appuie le réel intérêt et la valeur ajoutée des systèmes de recommandation.

1.2 Plan du mémoire

Ce mémoire est structuré de la manière suivante :

Le chapitre 2 introduit notre problématique et différents domaines d'application. L'objectif est de présenter les concepts ayant servi de support à nos travaux ainsi que notre problématique de recherche, en précisant l'intérêt des recommandations et leur pertinence ainsi que les domaines d'applications sur lesquels portent nos travaux.

Après avoir différencié la recommandation et la personnalisation et défini/présenté les concepts inhérents aux systèmes de recommandation, nous positionnons nos travaux dans le chapitre 3.

Le chapitre 4 présente notre approche générique de recommandation. Nous y détaillons nos travaux sur les recommandations d'éléments et/ou d'utilisateurs dans différents domaines.

Le chapitre 5 détaille notre approche d'évaluation des systèmes de recommandation. Il existe différentes techniques de recommandation et de nombreux systèmes qui veulent évoluer au cours du temps en essayant d'être de plus en plus proches des attentes/besoins des utilisateurs. Il est donc nécessaire d'évaluer les systèmes de recommandation et ainsi vérifier si les recommandations proposées sont pertinentes pour un utilisateur donné.

---

quatre flux d'information empruntant des canaux électroniques (téléphone, radio, télévision et Internet). L'étude estime la quantité d'informations nouvellement créée chaque année à environ 2 exaoctets par an (1 exaoctet = 10^{18} octets).

Dans ce souci d'être toujours au plus près des besoins des utilisateurs, le chapitre 6 propose un premier pas vers l'amélioration des systèmes de recommandation en abordant la réorganisation de logs/journaux, le problème du démarrage à froid ou en intégrant des données/informations contextuelles connues pour influencer les préférences, les envies et les intérêts des utilisateurs, et donc leurs décisions.

Le chapitre 7 permet de finaliser la présentation de nos travaux. Notamment, nous y explicitons les principales fonctionnalités des outils que nous avons développés, nous y listons les projets auxquels nous avons participés ainsi que les publications associées à ces travaux.

Le chapitre 8 conclut ce mémoire et ouvre quelques perspectives de recherche.

Chapitre 2

Contexte et Problématique

Sommaire

2.1 La recommandation 11
2.2 Domaines d'applications 12
2.2.1 Entrepôts de données 13
2.2.2 Environnements de travail collaboratif 13
2.2.3 Plateformes d'apprentissage en ligne 15
2.2.4 Les villes numériques intelligentes 15
2.2.5 Systèmes d'alertes précoces 16
2.2.6 Recommandations individuelles ou à visée publique 17
2.3 Verrous scientifiques 17
2.4 Synthèse 18

Comme indiqué dans le chapitre précédent, l'utilisateur doit être aidé pour synthétiser l'information et explorer les données (quel que soit le domaine d'application). Dans ce contexte de « sur-information » et en raison de l'accroissement des capacités de calcul et de stockage, il est difficile de savoir exactement quelles informations rechercher et où les chercher : des recommandations sont donc nécéssaires.

L'objectif de ce chapitre est de présenter les concepts ayant servi de support à nos travaux ainsi que notre problématique de recherche. Après avoir introduit notre rapport aux données / informations qui nous « envahissent » au quotidien, nous exposons l'intérêt des recommandations et de leur pertinence. Enfin, nous précisons les domaines d'applications sur lesquels portent nos travaux.

2.1 La recommandation

Le terme « recommandation » a plusieurs sens, l'Académie Française [Fra32] en donne six définitions différentes, à savoir :

1. Exhortation instante, conseil pressant, *Il a fait cela malgré toutes mes recommandations.* ;
2. Action de recommander ;
3. Prière que l'Eglise catholique fait à Dieu pour les agonisants ;
4. Estime qu'on a pour la vertu, le mérite, *La sainteté de sa vie l'avait mis partout en grande recommandation.* ;

5. Titre dont on peut se prévaloir, qui vous vaut de l'estime ;
6. Administration postale : action de recommander une lettre, un paquet.

Dans le cadre de ce mémoire, nous allons nous concentrer sur la définition 2, où recommander, toujours selon [Fra32], consiste, entre autres, à « conseiller particulièrement quelque chose, exhorter une personne à quelque chose ».

Ainsi, la recommandation peut être vue comme un dialogue entre une personne experte d'un domaine donné et une autre personne qui souhaite acquérir des informations dans ce domaine. Par exemple, un bibliothécaire suggère, en fonction soit des dernières nouveautés, soit en fonction des goûts (clairement explicités) d'un de ses clients, des ouvrages à ce dernier. Cette recommandation de lecture (*reader's advisory* en anglais) existe depuis la fin du XIXème siècle [Cro05]. Ici, la recommandation est donc un moyen d'aider un client de la bibliothèque à choisir des livres parmi tous les ouvrages de la bibliothèque à propos desquels le client n'a pas de connaissances suffisantes pour les trier et évaluer leur pertinence. Ainsi, la communication et le partage de l'information semblent être le fondement de la recommandation.

La recommandation est un domaine scientifique qui cherche à adapter l'accès à l'information à un utilisateur et ainsi l'aider à choisir des éléments de contenu dans un catalogue. Dans notre contexte de déluge de données/informations, ce catalogue peut être très/trop volumineux. Des techniques informatiques ont donc été proposées pour « automatiser » la recommandation. Le problème peut être formulé de la manière suivante : comment guider l'individu/utilisateur dans son exploration de données afin qu'il trouve des informations pertinentes ? Il s'agit de guider l'utilisateur lors de son exploration de la quantité d'informations à sa disposition en cherchant pour lui, les informations qui paraissent pertinentes. C'est une forme particulière de filtrage visant à présenter les éléments d'information (films, musique, livres, images, pages web, ...) susceptibles d'intéresser l'utilisateur. Généralement, à partir de certaines caractéristiques de référence [1], le système de recommandation cherche à prédire « l'avis » que donnerait l'utilisateur à chaque élément et lui recommande ceux obtenant le meilleur « avis ».

2.2 Domaines d'applications

Les systèmes de recommandation sont mis en pratique dans de nombreux domaines, que ce soit dans le contexte d'applications commerciales, industrielles, académiques, ... En effet, parmi les systèmes de recommandation les plus connus, nous pouvons citer ceux utilisés dans le commerce électronique, comme *Amazon.com* [Ama17] dans le cadre de la recommandation de produits et *Netflix* [Net17] dans le cadre de la recommandation de films. Mais, de nos jours, de nombreux systèmes que nous utilisons au quotidien proposent des recommandations à leurs utilisateurs. Citons par exemple, la recommandation de groupes de travail ou de personnes comme sur *LinkedIn* [Lin17], la recommandation « d'amis » comme sur *Facebook* [Fac17], la recommandation de musique comme sur *last.fm* [las17] ou encore la recommandation de nouvelles comme sur *Forbes.com* [For17].

Nous avons donc souhaité travailler sur différents domaines ayant des caractéristiques diverses. Dans la continuité des travaux de Doctorat [Neg09], nous avons évolué dans le domaine des entrepôts de données, où la recommandation consiste à suggérer des éléments (requêtes) à un utilisateur dans un environnement constitué de peu d'utilisateurs. Puis, afin d'élargir nos travaux, nous nous

1. Ces caractéristiques peuvent, par exemple, provenir des éléments d'information eux-mêmes, on parle alors d'« approche basée sur le contenu » ou, elles peuvent provenir de l'environnement social, on parle alors de « filtrage collaboratif ».

sommes intéressés à des domaines où le nombre d'utilisateurs est plus conséquent, notamment afin d'avoir une matrice d'utilité plus fournie, comme dans les environnements de travail collaboratif et l'apprentissage en ligne. Mais nous nous sommes également intéressés à des domaines où les enjeux des décisions prises sont importants et où, de ce fait, la qualité des recommandations est essentielle, comme pour les villes intelligentes et les systèmes d'alertes précoces (gestion de crises). Nous présentons ci-après ces différents domaines, à savoir :

- les entrepôts de données destinés au stockage de données massives pour la prise de décisions (stratégiques) qu'un décideur doit explorer ;
- les environnements de travail collaboratif qui n'échappent ni au déluge de données ni à la révolution numérique,
- les plateformes d'apprentissage en ligne qui sont impactées par l'augmentation du nombre d'apprenants et/ou de ressources ;
- les villes numériques intelligentes (*smart cities*) où les défis sociétaux relevés par des programmes tels que H2020 [2] les ont poussées à améliorer leurs prises de décisions (stratégiques, politiques, ...) ;
- et les systèmes d'alertes précoces (*early warning systems*) qui sont développés pour la prise de décision en situation de crise et où prendre la « bonne » décision est primordial.

2.2.1 Entrepôts de données

Comme indiqué en début de chapitre, de nos jours, avec l'abondance de données à notre disposition, les volumes de données à traiter sont de plus en plus importants [LVC+03]. Apparus pour gérer de tels volumes de données issues de sources hétérogènes, les entrepôts de données constituent l'outil essentiel de collecte et de mise à disposition des données en vue de leur analyse interactive, rapide et dynamique [Inm92, JLVV01]. Pour gérer les données, les entreprises disposent d'un côté, de systèmes transactionnels (OLTP - *On-Line Transactional Processing*) qui traitent les données sources de l'entrepôt et de l'autre, de systèmes analytiques (OLAP - *On-Line Analytical Processing*) [CCS93] qui, quant à eux, traitent les données contenues dans l'entrepôt permettant ainsi une analyse efficace des données grâce à différents outils tels que les requêteurs ou les logiciels d'analyses et de fouille. La figure 2.1 [CD97] résume cette démarche où les données sources sont entreposées puis modélisées sous forme de cubes de données avant d'être analysées.

Ces systèmes décisionnels permettent donc aux décideurs d'analyser l'activité d'une entreprise ou d'une organisation en explorant des données historisées et agrégées d'un entrepôt de données [Col96, Kim96]. Cependant, l'exploration d'un entrepôt volumineux peut s'avérer fastidieuse pour un décideur [LVC+03]. Ainsi, comment guider l'utilisateur au cours de son exploration d'informations volumineuses vers des informations plus pertinentes sous la forme de requêtes OLAP ?

2.2.2 Environnements de travail collaboratif

Avec les progrès de la science et des technologies, résoudre un problème technique ou d'ingénierie demande plus que jamais de nombreuses personnes ayant des compétences dans différents domaines et capables de travailler ensemble. La collaboration est une sorte de modèle de travail de groupe qui unit les caractéristiques et les compétences de tous les membres du groupe. La collaboration est obtenue en appliquant des outils de collaboration. Le but d'un outil de collaboration est de soutenir un groupe de personnes pour atteindre un but/objectif commun qu'elles se sont fixées [LBP08].

2. Programme européen pour la recherche et le développement pour la période 2014-2020 : http ://www.horizon2020.gouv.fr.

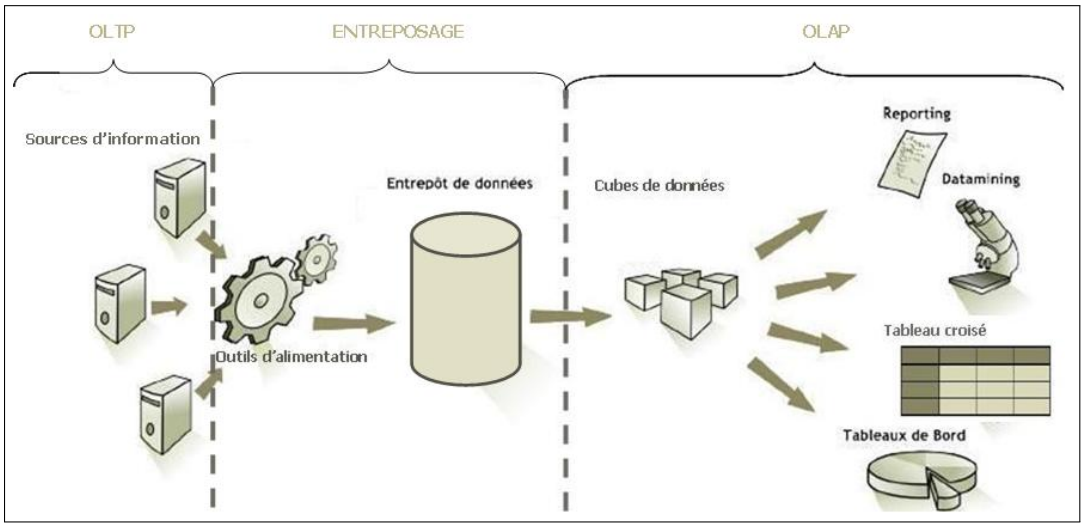

FIGURE 2.1 – Démarche décisionnelle

Avant l'ère de l'information, les gens collaboraient grâce à des outils « non technologiques » tels que le papier, les post-it ou le tableau blanc. De nos jours, l'existence de logiciels collaboratifs plus complexes et basés sur le web comme les wikis ou *Microsoft SharePoint* [Sha17] intégrés dans un environnement de travail agile nous rend plus efficaces [WGC+09]. Les environnements de travail collaboratif (*Collaborative Working Environment* - CWE) fournissent aux gens une aide dans leur travail individuel et professionnel. Les recherches en environnement collaboratif impliquent des problèmes et considérations d'ordres organisationnels, techniques et sociaux. Les éléments d'un environnement de travail collaboratif sont : le courrier électronique, les messageries instantanées, le partage d'application, la visioconférence, l'espace de travail collaboratif et la gestion de documents, la gestion des tâches et de processus, les wikis[3], les blogs (où les entrées sont catégorisées par groupe, communauté ou autres concepts permettant la collaboration). La notion d'environnement collaboratif est une combinaison de plateforme collaborative et de réseau social.

Les ordinateurs et les smartphones, par exemple, permettent aux gens de travailler ensemble indépendamment du lieu et du moment. Ces outils peuvent facilement enregistrer les comportements et activités des gens afin de juger de la participation de chacun et éventuellement d'évaluer leurs compétences. Pour avoir un groupe (collaboratif) compétitif, il faut évaluer les compétences des membres du groupe afin de mieux répartir les tâches et les ressources. Ainsi l'estimation de la compétence d'une personne est non seulement nécessaire, mais aussi cruciale pour la collaboration. Les entreprises/organisations cherchent de nouvelles façons d'acquérir, de gérer et de conserver les talents, précieux pour réussir.

Malheureusement, à l'heure actuelle, il existe un fossé entre les activités de collaboration et l'évaluation des compétences. De ce fait, comment améliorer les échanges entre les membres des environnements de travail collaboratif en tenant compte de leurs traces de collaboration ?

3. Par exemple, les pages de wikis décrivent des concepts permettant la compréhension commune par un groupe ou une communauté

2.2.3 Plateformes d'apprentissage en ligne

Les plateformes d'apprentissage en ligne (par exemple, *Massive Open Online Course* - MOOC), les jeux sérieux (*serious game*), l'apprentissage social, l'apprentissage mobile,... sans aucun doute, l'éducation et l'apprentissage n'échappent pas à la révolution numérique qui bouleverse notre façon d'apprendre, de travailler et de penser. Fini les diaporamas interminables et ennuyeux en face-à-face avec l'écran de votre ordinateur, maintenant tout le monde apprend sur Internet en suivant un MOOC ou en regardant des vidéos postées par des experts du monde entier, en participant à des forums via des réseaux collaboratifs ou en pratiquant via les jeux sérieux.

En général, les composants de tels systèmes sont les suivants :

- une communauté d'apprenants ;
- une plateforme d'apprentissage ;
- des tuteurs ;
- du contenu textuel ou multimédia (ressources) ;
- une stratégie d'apprentissage ;
- des activités de validation.

Outre son accessibilité depuis tout ordinateur ou tablette, une connexion Internet, son émancipation des contraintes de temps et de lieu (consultation flexible) et souvent une taille adaptée (études de cas, animations, interactivité...), les principaux facteurs clés pour l'adoption de la formation en ligne sont un contenu pertinent et une bonne pédagogie. Ainsi, répondre aux attentes/besoins des apprenants tout en leur permettant d'apprendre encore mieux est un défi majeur pour les plateformes d'apprentissage en ligne. Or, comment améliorer l'apprentissage des apprenants en termes de contenus et de choix pédagogiques adaptés ?

2.2.4 Les villes numériques intelligentes

Le terme « ville (numérique) intelligente », traduction de l'anglais *smart city*, désigne une ville utilisant les technologies de l'information et de la communication (TIC) pour améliorer la qualité des services urbains ou encore réduire ses coûts. Ce concept émergent désigne un type de développement urbain apte à répondre à l'évolution ou l'émergence des besoins des institutions, des entreprises et des citoyens, tant sur le plan économique, social, qu'environnemental [BAG+12, CNW+12].

La ville est un territoire intelligent ayant un caractère multidimensionnel en termes d'acteurs, de domaines ciblés et de briques technologiques. Sous l'influence de la mondialisation et l'impact des TIC qui modifient radicalement notre relation avec l'espace et le temps, la ville développe de plus en plus ses activités. La ville hiérarchique enfermée dans ses frontières locales se transforme en une ville étendue, ouverte et adaptable.

Il y a six caractéristiques/catégories pour classer les villes intelligentes et les comparer [GFK+07] : l'économie intelligente (*smart economy*), la mobilité intelligente (*smart mobility*), l'environnement intelligent (*smart environment*), des gens intelligents (*smart people*), une manière de vivre intelligente (*smart living*) et la gouvernance intelligente (*smart governance*).

Cependant, être « intelligent » est un défi de plus en plus important pour de nombreuses villes ou communautés. C'est d'un intérêt particulier dans le domaine des TIC et pour de tels systèmes où il y a des enjeux économiques, sociaux et autres. La ville doit s'adapter et mettre en place des actions/initiatives dans ce but. Par conséquent, comment aider une ville à identifier les actions à mettre en œuvre pour atteindre son objectif ?

2.2.5 Systèmes d'alertes précoces

Les catastrophes naturelles sont une cause constante de souffrance humaine et de pertes économiques à travers le monde. Le changement climatique et l'urbanisation rapide ne font qu'aggraver le problème. Pour élaborer des plans d'urgence efficaces dans les zones à risques, il est vital de disposer d'un système d'alertes précoces (*early warning system* - EWS). Un tel système doit être capable de surveiller et de prévoir le comportement de l'environnement puis d'émettre des alertes assez tôt pour permettre aux populations de prendre les mesures qui s'imposent. Il est à noter qu'un système d'alertes précoces est spécifique à un type de catastrophe (dans différents domaines : l'école [HT09], la finance [GJ02, Che09], l'environnement [AGW10, Ger10], l'humanitaire [BS00, Bak01, Int13], la biologie [GSB+10], ...) mais aussi à l'environnement pour lequel il a été mis en place (zone géographique, décisions politiques, ...). Par conséquent, il n'existe pas deux systèmes identiques.

Un système d'alertes précoces complet et efficace comprend quatre éléments. L'échec à l'une des parties peut signifier l'échec de l'ensemble du système [Uni06] :

1. Connaissance des risques : Connaissance préalable des risques encourus et planification du système de contrôle en termes de capteurs, mesures, échelles, seuils, ... (l'évaluation et la cartographie des risques aide à prioriser les besoins du système d'alertes précoces et à guider les préparatifs d'intervention et de prévention des catastrophes. Cette évaluation des risques peut être fondée sur l'expérience historique et humaine, sociale, économique, et les vulnérabilités environnementales).
2. Services d'alertes - Maintenance technique et surveillance d'alertes.
3. Diffusion des alertes : compréhensible pour les personnes à risques, les organisations et toutes les personnes qui doivent répondre à ces alertes. Les avertissements doivent atteindre leurs destinataires par le(s) meilleur(s) canal (canaux) de diffusion (adapté au destinataire) en choisissant la bonne information à diffuser (pour ne pas causer la panique mais faire en sorte que les personnes agissent).
4. Capacité d'intervention - Connaissance et préparation à agir : il est essentiel que les personnes à risques comprennent leurs risques, elles doivent respecter le service d'alerte et doivent savoir comment réagir.

Un système d'alertes précoces peut être défini comme une chaîne de systèmes de communication d'information comprenant des capteurs, de la détection, de la décision et des sous-systèmes transitoires, dans cet ordre, qui travaillent en collaboration, pour prévenir et signaler des perturbations affectant négativement la stabilité du monde physique ; et qui donnent suffisamment de temps au système de réponse pour préparer les ressources et les mesures d'intervention (actions) pour minimiser l'impact sur la stabilité du monde physique [Wai10]. Ainsi, un système d'alertes précoces est un ensemble d'outils pour prédire des aléas [Uni06, JEQR10].

Comprendre et réagir de manière adéquate aux signaux d'alertes précoces, avant que ceux-ci ne se manifestent et ne se transforment en besoins aigus est dans de nombreux cas plus efficace que de répondre seulement après que la catastrophe ait eu lieu [Swi14]. Idéalement, les signaux d'alertes précoces devraient déclencher des actions appropriées pour prévenir la population des dommages.

Les alertes et les décisions d'évacuer la population ; de déployer des équipes de secours en cas de catastrophe dans une ville/région ; ou de pré-positionner des marchandises, sont l'interface entre la préparation et la réponse. Plus tôt une alerte est lancée, plus il y a de temps pour planifier et coordonner les activités de prévention. Cependant, les informations sur le danger deviennent de plus en plus précises au fur et à mesure que le temps passe (de même, la menace devient de plus en plus concrète). Les décideurs doivent donc prendre en compte : (i) les points d'intervention : quelles sont

les actions possibles ? Combien de temps est nécessaire pour effectuer chaque action ? Combien de temps jusqu'à ce que la catastrophe frappe ? ; (ii) l'exactitude et la précision des informations sur la catastrophe : son ampleur, son chemin/parcours, ...

Par conséquent, comment aider les décideurs dans de telles situations de crises ? Quelles actions faut-il mettre en place pour améliorer la gestion de crises et le déclenchement des alertes précoces ?

2.2.6 Recommandations individuelles ou à visée publique

Chaque domaine d'application a des caractéristiques propres et certains points communs existent entre eux. Cependant, en terme de recommandation, il est difficile de les classer puisque les éléments manipulés ou les utilisateurs n'ont pas les mêmes caractéristiques. En effet, un système de recommandation dans le domaine des environnements de travail collaboratif, peut recommander, par exemple des utilisateurs ou des sujets intéressants de l'environnement. Dans le cadre de l'apprentissage en ligne, les recommandations peuvent prendre la forme, par exemple, de supports de cours pour un apprenant alors que dans les entrepôts de données, il s'agira par exemple, de requêtes recommandées à un décideur. Dans le domaine des villes numériques intelligentes, le système peut recommander, par exemple, des initiatives à mettre en place dans la ville auprès des décideurs ou encore indiquer aux citoyens des personnes à contacter. Enfin, concernant les systèmes d'alertes précoces, il peut s'agir de recommander, par exemple, des actions à mettre en place en cas de crise.

A partir de ces constats, nous avons décidé d'organiser les domaines d'applications en fonction des caractéristiques des recommandations qu'un système de recommandation peut retourner, et plus particulièrement, à qui s'adresse la (ou les) recommandation(s). Nous avons ainsi établi deux classes de recommandations :

- celles qui s'adressent à un utilisateur (ou un groupe d'utilisateurs) et qui seront utilisées dans un cadre restreint à l'utilisateur (ou au groupe), autrement dit, utilisées pour prendre des décisions qui le concerne, c'est-à-dire *individuelles*
- celles qui s'adressent à un décideur (ou groupe de décideurs) mais qui seront utilisées pour prendre des décisions qui auront un impact sur les populations, c'est-à-dire une *visée publique*.

2.3 Verrous scientifiques

Notre objectif est d'aider l'utilisateur à se retrouver dans un grand espace de données en lui proposant des recommandations pertinentes. Cependant, selon [HKTR04], une « barrière magique » empêche les systèmes de recommandation d'atteindre la perfection : les utilisateurs eux-mêmes. En effet, le manque d'avis, leur inconsistance voire leur divergence/contradiction est un réel frein.
Ainsi,

- Quelle que soit l'application (ou la classe d'application - *individuelle* ou *à visée publique*), comment concevoir un système de recommandation de qualité (au sens de la qualité logicielle [4]) ?
- Comment être sûr de cette qualité et de la pertinence des recommandations (que le système répond correctement au besoin de l'utilisateur) ?

4. Les systèmes informatisés doivent suivre les principes de l'ingénierie et de la qualité logicielle. L'ingénierie logicielle repose sur sept principes [GJM02] : la rigueur, la décomposition en sous-problèmes, la modularité, l'abstraction, l'anticipation des évolutions, la généricité et la construction incrémentale. Et, la norme ISO 9126 définit six groupes d'indicateurs de qualité logicielle : la capacité fonctionnelle, la facilité d'utilisation, la fiabilité, la performance, la maintenabilité et la portabilité.

– Comment prendre en compte des données/informations complémentaires adaptées ?

De telles questions lèvent des verrous scientifiques [BR16] liés notamment à :
- actuellement, les systèmes de recommandation sont définis pour un cas d'application spécifique limitant de ce fait l'adaptabilité, la réutilisabilité et la généricité (entre autres, liés à l'ingénierie et à la qualité logicielle[5]) de tels systèmes ;
- l'évaluation des recommandations : est une étape clé. Malheureusement, les performances des systèmes de recommandation sont souvent basées sur la précision des recommandations [Ess10], elle-même liée à la disponibilité des données...
- le démarrage à froid : ce problème se traduit par la difficulté de générer des recommandations pour des nouveaux éléments ou des nouveaux utilisateurs qui viennent d'être introduits dans le système de recommandation [SPUP01] ;
- la prise en compte du contexte de l'utilisateur : le choix des informations, leur modélisation et leur intégration ouvrent de nombreuses pistes de recherche [AT08, CCFG12].

2.4 Synthèse

Dans ce chapitre, nous avons présenté nos contextes et problématiques de recherche, à savoir mettre en place un système de recommandation afin de guider un utilisateur/décideur à travers le grand volume de données/informations à sa disposition et ainsi faciliter sa prise de décision dans de nombreux domaines comme les environnements de travail collaboratif, les plateformes d'apprentissage en ligne, les entrepôts de données, les villes intelligentes et les systèmes d'alertes précoces, ... Puis nous avons classer les recommandations en fonction du domaine d'application et du destinataire de la recommandation : les recommandations individuelles (destinées à l'utilisateur pour prendre des décisions qui le concerne) et les recommandations à visée publique (utilisées pour prendre des décisions qui auront un impact sur les populations).

Bien que les systèmes de recommandation soient définis pour des applications spécifiques, le nombre de systèmes existants et le nombre de domaines variés dans lesquels ils ont été développés, devraient permettre de tirer avantages des approches, méthodes, et outils existants. Ainsi, nos travaux ne se limitent pas à un domaine particulier mais se positionnent dans une mouvance plus générale de recherche (orientée vers la pluridisciplinarité et la transversalité) que nous pouvons même prétendre précéder.

Par ailleurs, afin d'assurer la qualité/pertinence des recommandations du point de vue de l'utilisateur, nos travaux le placent au cœur de l'évaluation des (systèmes de) recommandations en proposant des méthodes d'évaluation subjective.

Enfin, pour tenter d'être encore plus près des attentes/besoins de l'utilisateur, des axes d'amélioration existent, notamment dans le cas du démarrage à froid et de la prise en compte des données/informations contextuelles de l'utilisateur.

5. https ://www.iso.org/standard/35733.html

Chapitre 3

Définitions et travaux existants

Sommaire

3.1 Recommandation *versus* personnalisation 20
3.1.1 Recommandation 20
3.1.2 Personnalisation 20
3.2 Profil d'utilisateur 21
3.3 Systèmes de recommandation 22
3.3.1 Catégorisation des systèmes de recommandation 23
3.3.1.1 Les approches basées sur le contenu 23
3.3.1.2 Les approches basées sur le filtrage collaboratif 26
3.3.1.3 Les approches hybrides 28
3.4 Méthodes/techniques utilisées 29
3.4.1 Modèle d'espace vectoriel 29
3.4.2 Mesures de similarité 30
3.4.2.1 Similarité cosinus 30
3.4.2.2 Coefficient de corrélation de Pearson 31
3.4.3 Réduction de dimensions 31
3.4.3.1 Analyse en composantes principales 32
3.4.3.2 Décomposition en valeurs singulières 32
3.4.4 Classification 33
3.4.4.1 Classification supervisée 33
3.4.4.2 Classification non-supervisée 34
3.4.5 TF-IDF 34
3.5 Evaluation de tels systèmes 35
3.5.1 Jeux de données, densité et erreurs 35
3.5.2 Mesures 36
3.5.2.1 Justesse 36
3.5.2.2 Autres mesures 38
3.6 Des définitions à nos travaux 39

Dans ce chapitre, après avoir différencié recommandation et personnalisation et présenté des définitions liées aux systèmes de recommandation, nous positionnons nos travaux.

3.1 Recommandation *versus* personnalisation

Avant toute autre chose, il est important de différencier la recommandation de la personnalisation qui sont souvent confondues dans de nombreux domaines [Neg15b].

3.1.1 Recommandation

De notre point de vue, un élément recommandé (film, livre, ...) est un élément existant, par exemple issu d'un ensemble d'éléments déjà visualisés dans une base de données ou un catalogue. L'élément recommandé peut, par conséquent, être complètement différent de l'élément initial car ils ne sont pas forcément issus du même catalogue ou ne s'adressent pas au même utilisateur ou encore, peuvent avoir des caractéristiques très différentes. De manière générale, la démarche de recommandation, à partir d'un ensemble d'éléments initial Q, renvoie un ensemble d'éléments Q' tel que, en général, $Q' \not\subseteq Q$ et $Q \not\subseteq Q'$ (au sens de l'inclusion ensembliste) [KWFH04, Neg09, MK09, Mar12].

Exemple 3.1.1– *Supposons qu'un utilisateur, passionné de « POMME » cherche à agrandir sa bibliothèque, qui contient majoritairement des livres de cuisine et notons cet ensemble de livres déjà possédés, notre ensemble Q. Des recommandations possibles pourraient être, des livres de cuisine (contenant des recettes de cuisine à base de pommes), mais aussi des livres de jardinage (expliquant par exemple comment planter et entretenir un pommier) et des livres de voyages (par exemple, sur la ville de New-York (aux Etats-Unis) dont le surnom est « la grosse pomme »), qui sont notre ensemble de livres recommandés Q'. Notons donc que l'ensemble d'éléments initial (les livres de la bibliothèque, étant majoritairement des livres de recettes de cuisine, Q) et l'ensemble de recommandations (contenant des livres de cuisine mais aussi des livres de jardinage et des livres de voyages, Q') ont peu de caractéristiques communes.*

3.1.2 Personnalisation

De notre point de vue, la personnalisation, quant à elle, correspond plus à une inclusion. En effet, dans le domaine des bases de données, par exemple, la personnalisation de requête est considérée comme un ajout de contraintes, issues du profil de l'utilisateur, à une requête donnée et se traduit par l'ajout de conditions de sélection [KI04, BGM+05]. De manière plus générale, la démarche de personnalisation, à partir d'un ensemble d'éléments Q renvoie un ensemble d'éléments Q' tel que les caractéristiques de Q' sont incluses dans celles de Q, $Q' \subseteq Q$ (au sens de l'inclusion ensembliste) [Neg09, Mar12].

Exemple 3.1.2– *Supposons que l'ensemble d'éléments Q à personnaliser soit les livres de recettes de cuisine contenant des recettes à base de pommes, et supposons que les préférences de l'utilisateur se portent uniquement sur les desserts, un livre mieux adapté aux préférences de l'utilisateur, c'est-à-dire personnalisé, Q' sera, par exemple, un livre de recettes de desserts à base de pommes.*

3.2 Profil d'utilisateur

Que ce soit dans le cadre de la personnalisation ou de la recommandation, il est important de connaître au mieux l'utilisateur : ses attentes/besoins mais aussi son environnement et ses connaissances propres, c'est-à-dire des données, informations et connaissances qui doivent être extraites et intégrées dans le système informatique.

Les données de l'utilisateur sont intégrées au sein du système informatique par l'intermédiaire de ce que nous appelons un profil utilisateur. Un profil utilisateur est un ensemble de données qui influencent le comportement d'un dispositif informatique en fonction des besoins/attentes de l'utilisateur [BDR91, LU03, BR07, DTBC08, CJSV09, TPJ+12]. Il peut contenir des données telles que, par exemple, identifiant, état civil, âge, sexe, préférences, compétences, historique des interactions avec le système, ... [Neg15b].

Notons que les données du profil de l'utilisateur sont représentées différemment selon les besoins, mais qu'en général elles sont stockées sous la forme de couples attribut-valeur où chaque couple représente une propriété du profil. Les propriétés peuvent éventuellement être regroupées par catégories. Dans l'exemple 3.2.1, le tableau 3.1 illustre le profil de l'utilisateur *Marie* qui contient pour le groupe *Livre*, l'attribut *Genre* ayant pour valeur *Thriller/Policier* et l'attribut *Auteur* ayant pour valeurs *S. King* et *M. Connelly*.

Ce profil se construit avec ou sans l'aide de l'utilisateur. En effet, lors de la construction du profil, il faut distinguer la collecte explicite de la collecte implicite des données. La collecte de données est explicite lorsque l'utilisateur exprime explicitement son avis, par exemple, classer une collection d'éléments en fonction de ses préférences, choisir l'élément le plus pertinent, créer une liste d'éléments intéressants pour l'utilisateur. *A contrario*, la collecte de données est implicite lorsque les préférences/avis de l'utilisateur sont induits en « analysant » les actions de l'utilisateur, par exemple, regarder quels éléments ont été visualisés par l'utilisateur, garder une trace des navigations de l'utilisateur (achats réalisés, éléments sur lesquels l'utilisateur s'est attardé, ...), analyser le réseau social de l'utilisateur et découvrir ses goûts/aversions [1].

Enfin, il est à noter que ce profil de l'utilisateur évolue au cours du temps.

Exemple 3.2.1– *Le tableau 3.1 illustre un extrait du profil de l'utilisateur Marie basé sur les informations/préférences qu'elle a données [Neg15b].*

Identifiant		1
Informations Personnelles	Utilisateur	Marie
	Sexe	Femme
Livre	Genre	Thriller/Policier
	Auteur	S. King, M. Connelly
Film	Genre	Jeunesse, Dessins Animés, Humour
	Réalisateur	G. Lucas

TABLE 3.1 – Extrait du profil de l'utilisateur *Marie*

1. Pour plus de détails sur les profils d'utilisateurs : [BR07, DTBC08, TPJ+12] en français et [BDR91, LU03, STB+16] en anglais.

3.3 Systèmes de recommandation

Les techniques utilisées dans les systèmes de recommandation ont été initiées dans des domaines de recherche tels que la recherche d'informations (*Information Retrieval*) [Ken71, Sal83, BYRN99] et l'apprentissage automatique (*Machine Learning*) [GK65, JD88, CH06]. Les systèmes de recommandation, quant à eux, ont commencé à être étudiés dans les années 90 en fonction de leur domaine d'application. Le premier système de recommandation permettait la recommandation de documents [RIS+94] puis ont suivi des systèmes de recommandation dans des domaines d'applications tels que la recherche web [BYHM04, WBC07], le e-commerce [SKR01], l'exploitation des usages du web (*Web Usage Mining*) [SCDT00, FS02, PPPS03, BYHMD05], les nouvelles [BP00] et bien d'autres. Le problème de la recommandation peut se résumer à l'estimation de scores pour des éléments qui n'ont pas encore été vus par un utilisateur. En effet, le nombre d'éléments ainsi que le nombre d'utilisateurs du système peuvent être très importants ; il est, de ce fait, difficile que chaque utilisateur voit tous les éléments ou que chaque élément soit évalué par tous les utilisateurs. Il est donc nécessaire d'estimer les scores pour les éléments non encore évalués.

Cette évaluation est habituellement basée sur les scores donnés par un utilisateur à d'autres éléments et sur une autre information qui sera formellement décrite ci-dessous. Lorsqu'il est possible d'estimer les scores pour les éléments non encore évalués, les éléments ayant les scores les plus élevés peuvent être recommandés à l'utilisateur. Plus formellement, [AT05] formule le problème de recommandation dans le domaine du e-commerce comme suit :

Définition 3.3.1 – *Recommandation en e-commerce*
Soit C l'ensemble de tous les utilisateurs, I l'ensemble de tous les éléments possibles qui peuvent être recommandés (comme des livres, des films ou des restaurants) et soit u une fonction qui mesure l'utilité d'un élément i à l'utilisateur c, c'est-à-dire $u : C \times I \to \mathbb{R}$.
Alors pour chaque utilisateur $c \in C$, nous voulons choisir l'élément $i' \in I$ qui maximise[2] *l'utilité à l'utilisateur : $\forall c \in C, i'_c = argmax_{i \in I} u(c, i)$.*

Dans les systèmes de recommandation, l'utilité d'un élément est habituellement représentée par un score qui indique comment un utilisateur particulier a aimé un élément particulier.

Exemple 3.3.1 – *Ici, les éléments sont des films que les utilisateurs Arnaud, Patrick, Marie et Elsa sont susceptibles d'avoir notés. Par exemple, l'utilisateur Arnaud a donné au film « Harry Potter » le score de 3 (sur 10). Nous obtenons la matrice $C \times I$:*

$u(c,i)$	*Harry Potter*	*L'Age de glace*	*L'Age de glace 2*	*OSS 117*	*Bienvenue chez les Ch'tis*
Elsa		8		2	7
Marie	9	8		3	6
Arnaud	3	5		5	
Patrick	5	3		3	3

Notons qu'une cellule (c, i) de la matrice correspond au score d'utilité assigné au film i par l'utilisateur c.

Le problème central des systèmes de recommandation vient du fait que cette utilité u n'est pas habituellement définie sur l'espace complet $C \times I$, mais seulement sur un certain sous-ensemble

2. L'argument maximum ou argmax représente la valeur de la variable pour laquelle la valeur de la fonction concernée atteint son maximum.

de celui-ci. Cela signifie que u doit être extrapolée à l'espace entier $C \times I$. Dans les systèmes de recommandation, l'utilité est typiquement représentée par les scores et est d'abord définie sur les éléments précédemment évalués par les utilisateurs. Par conséquent, le moteur de recommandation devrait pouvoir prévoir les scores des combinaisons non évaluées d'éléments × utilisateurs et publier des recommandations appropriées basées sur ces prévisions.

Une fois que les scores inconnus sont estimés, des recommandations sont émises en choisissant le(s) score(s) le(s) plus élevé(s) parmi tous les scores prévus pour cet utilisateur, selon la formule donnée dans la définition 3.3.1.
Alternativement, nous pouvons recommander les meilleurs éléments à un utilisateur, c'est-à-dire les éléments les plus pertinents pour un utilisateur ou un ensemble d'utilisateurs à un élément, c'est-à-dire les utilisateurs les plus intéressés par l'élément.

3.3.1 Catégorisation des systèmes de recommandation

Les systèmes de recommandation sont habituellement classifiés en fonction des scores déjà évalués, utilisés pour estimer les scores manquants [HSRF95, BS97, KK06] :

- méthode basée sur le contenu : l'utilisateur se verra recommander des éléments semblables (au sens d'une mesure de similarité entre éléments) à ceux qu'il a préférés dans le passé.
  Exemple 3.3.2– *Considérons la matrice Utilisateurs × Films de l'exemple 3.3.1, le système va affecter une utilité à des films qu'Elsa n'a pas encore évalués. Cette utilité est calculée en fonction des scores des films déjà évalués par Elsa. Les films « L'Age de glace » et « Bienvenue chez les Ch'tis » ont été évalués quasiment de la même manière. A partir de cette observation et du fait que les films « L'Age de glace » et « L'Age de glace 2 » sont très « proches », le système assigne au film « L'Age de glace 2 » pour Elsa, le meilleur score attribué par Elsa aux films « L'Age de glace » et « Bienvenue chez les Ch'tis », c'est-à-dire 8.*

- méthode collaborative ou basée sur le filtrage collaboratif : l'utilisateur se verra recommander des éléments que d'autres utilisateurs ayant des goûts et des préférences similaires (au sens d'une similarité entre utilisateurs et éléments) ont aimé dans le passé.
  Exemple 3.3.3– *Considérons à nouveau la matrice Utilisateurs × Films de l'exemple 3.3.1, le système va chercher les utilisateurs similaires à Elsa : ici, Marie car Marie et Elsa ont noté tous les films quasiment de la même manière. Puis, à partir de l'observation des scores que Marie a donnés aux films, le système assigne au film « Harry Potter » pour Elsa, le score que Marie a donné à « Harry Potter », c'est-à-dire 9.*

- méthode hybride : combinaison des deux méthodes précédentes.

3.3.1.1 Les approches basées sur le contenu

Pour les recommandations basées sur le contenu (voir [PB07] pour plus de détails), le système fait coïncider les caractéristiques de l'élément du catalogue avec le profil de l'utilisateur pour décider de sa pertinence pour l'utilisateur. La tâche de recommandation consiste donc à déterminer quels éléments coïncident le mieux avec les préférences de l'utilisateur. Une telle approche ne requiert pas une grande communauté d'utilisateurs ou un gros historique d'utilisation du système. La figure 3.1 illustre ce processus.

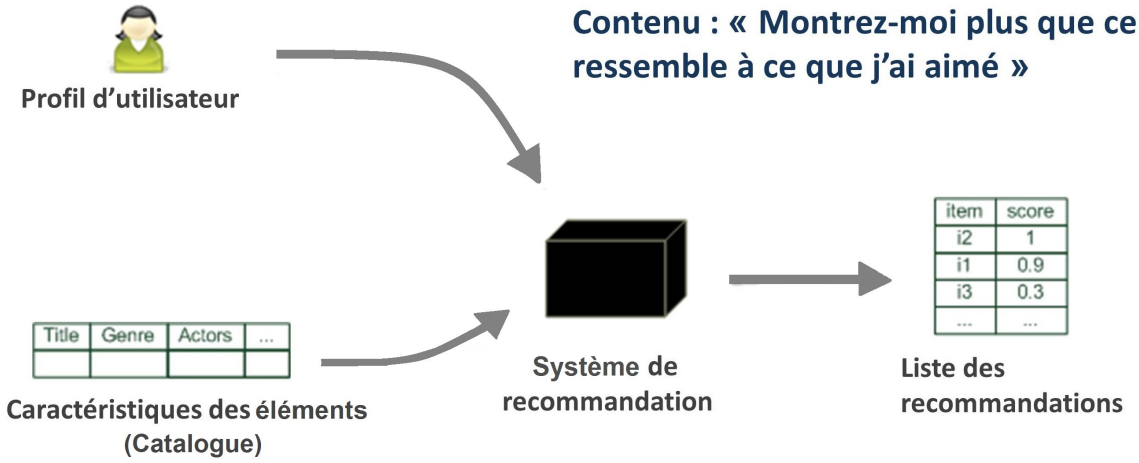

FIGURE 3.1 – Le système de recommandation basé sur le contenu vu comme une boîte noire (adapté de [JZFF10])

La manière la plus simple de décrire un catalogue d'éléments est d'avoir une liste explicite des caractéristiques de chaque élément (on parle aussi d'attributs, de profil d'élément, ...). Pour un livre, par exemple, on peut utiliser le genre, le nom des auteurs, l'éditeur ou toute autre information relative au livre, puis stocker ces caractéristiques (dans une base de données par exemple). Quand le profil de l'utilisateur est exprimé sous forme d'une liste d'intérêts basée sur les mêmes caractéristiques, la tâche de recommandation consistera donc à faire coïncider les caractéristiques de l'élément et le profil de l'utilisateur.

La coïncidence entre les caractéristiques des éléments et le profil de l'utilisateur peut être mesurée de différentes manières : l'indice de Dice [Dic45] ou d'autres mesures de similarité [BYRN99], le TF-IDF (*Term Frequency-Inverse Document Frequency*) [SWY75], les techniques basées sur la similarité des espaces vectoriels (kNN [BP00], méthode de Rocchio [Roc71], les approches bayésiennes [PB07], les arbres de décision [Qui93], ...) couplés avec des techniques statistiques comme le χ^2 [CS93] par exemple, lorsqu'il y a trop de mots-clés.

Les systèmes de recommandation basés sur le contenu ont comme avantages :

- de recommander des éléments similaires à ceux que les utilisateurs ont aimés dans le passé ;
- de prendre en compte le profil des utilisateurs qui est la clé pour avoir des recommandations les plus pertinentes pour l'utilisateur ;
- faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments fonctionne pour de nombreux types de données (textuelles, numériques, ...) puisqu'on utilise généralement des listes de mots-clés ;
- les données relatives aux autres utilisateurs sont inutiles ;
- il n'y a pas de problème de démarrage à froid [3] lorsqu'un nouvel élément est ajouté au catalogue

3. Bien que les systèmes de recommandation permettent de recommander des éléments pertinents à un utilisateur, ils s'avèrent inopérants lorsque de nouveaux éléments sont ajoutés au catalogue ou lorsque les utilisateurs sont différents ou nouveaux. Ce problème du démarrage à froid (*cold-start problem*) est rencontré lorsque les recommandations sont nécessaires pour des éléments et/ou des utilisateurs pour lesquels nous n'avons aucune information explicite ou

ou de faible densité puisqu'il s'agit de faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments ;
- il est possible de faire des recommandations à des utilisateurs avec des goûts « uniques » ;
- de recommander de nouveaux éléments ou même des éléments qui ne sont pas populaires.

Cependant, de tels systèmes ont aussi des inconvénients :
- tous les contenus ne peuvent pas être représentés avec des mots-clés (exemple : les images, ...) ;
- des éléments représentés par le même ensemble de mots-clés ne peuvent pas être distingués ;
- les utilisateurs ayant visualisé un très grand nombre d'éléments sont un problème (trop d'informations dans le profil de l'utilisateur à faire coïncider avec les caractéristiques des éléments) ;
- lorsqu'un nouvel utilisateur commence à utiliser le système, il n'existe pas d'historique ;
- risque de « sur-spécialisation », c'est-à-dire que l'on se limite aux éléments similaires (réponses trop homogènes) ;
- les profils des utilisateurs restent difficiles à élaborer et, qui plus est, il faut faire attention à l'évolution des intérêts de l'utilisateur et les prendre en compte ;
- pour produire des recommandations précises, l'utilisateur doit fournir un *feedback*[4] sur les suggestions retournées mais cela est chronophage pour les utilisateurs ; finalement,
- ces systèmes sont entièrement basés sur les scores d'éléments et les scores d'intérêt : moins il y a de scores, plus l'ensemble de recommandations possibles est limité.

Exemple 3.3.4– *Considérons un catalogue de livres tel que l'extrait présenté dans le tableau 3.2 [Neg15b] ainsi que l'extrait du profil de l'utilisateur Marie basé sur ses précédents achats de livres et sur les informations qu'elle a données concernant ses préférences, tel qu'illustré par le tableau 3.3 [Neg15b]. Nous souhaitons proposer à Marie des livres susceptibles de lui plaire. Il s'agit de faire coïncider le tableau 3.2 et le tableau 3.3 et de définir quels livres correspondent le mieux aux préférences de Marie. Le tableau 3.4 résume la coïncidence entre les deux tableaux (oui : les informations coïncident; non : les informations ne coïncident pas). Ainsi à partir du tableau 3.4 [Neg15b], intuitivement, les livres « Shining » et « Millenium » pourraient être recommandés à Marie (puisque les caractéristiques de ces deux livres coïncident le mieux avec les préférences de Marie).*

Titre	Genre	Auteur	Prix	Mots-clés
Shining	Thriller	S. King	19,50	Alcoolisme, Colorado, Médium, Surnaturel, Hôtel, ...
Millenium	Policier	S. Larsson	23,20	Journalisme, Investigation, Meurtre, Suède, Politique, ...
Le Journal de Bridget Jones	Romance	H. Fielding	8,50	Célibataire, Humour, Amour, Trentenaire, Journal intime, ...

TABLE 3.2 – Extrait du catalogue de livres

implicite. Il y a donc plusieurs problèmes liés au démarrage à froid [Klo09] : nouvel utilisateur et nouvel élément.

4. Un *feedback* utilisateur est un retour, une appréciation de l'utilisateur suite à une action du système.

Titres	Genres	Auteurs	Prix	Mots-clés
Docteur Sleep, Joyland, Intervention Suicide, Darling Lilly, ...	Policier, Thriller	S. King, M. Connelly, ...	15	Détective, Meurtre, Surnaturel, ...

TABLE 3.3 – Extrait du profil de l'utilisateur *Marie*

Préférences utilisateur	Shining	Millenium	Bridget Jones
Genre	oui	oui	non
Prix	oui	oui	oui
Mots-clés	oui	oui	non

TABLE 3.4 – Coïncidence entre les caractéristiques des livres et les préférences (profil) de Marie

3.3.1.2 Les approches basées sur le filtrage collaboratif

Les systèmes basés sur le filtrage collaboratif (voir [SFHS07] pour plus de détails) produisent des recommandations en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs [Neg15b]. De tels systèmes ne tentent pas d'analyser ou de comprendre le contenu des éléments à recommander. Ils sont capables de suggérer de nouveaux éléments à des utilisateurs qui ont des préférences similaires à d'autres. La méthode consiste à faire des prévisions automatiques (filtrage) sur les intérêts d'un utilisateur en collectant des avis de nombreux utilisateurs. L'hypothèse sous-jacente de cette approche est que ceux qui ont aimé un élément spécifique dans le passé ont tendance à aimer cet élément spécifique (ou un autre très « proche »), de nouveau dans l'avenir. La figure 3.2 illustre ce processus.

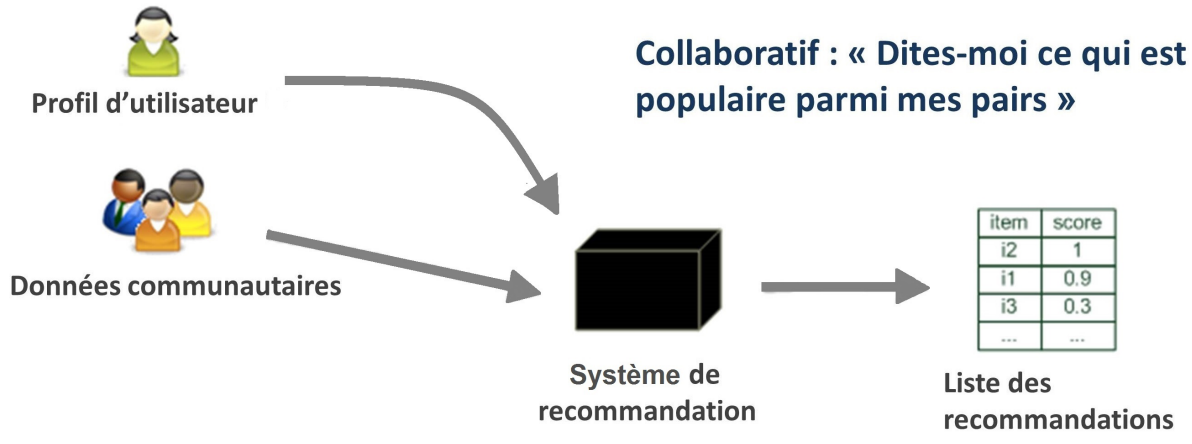

FIGURE 3.2 – Le système de recommandation collaboratif vu comme une boîte noire (adapté de [JZFF10])

L'idée des approches collaboratives est d'essayer de prédire l'opinion qu'aura l'utilisateur sur les différents éléments et d'être en mesure de recommander le « meilleur » élément à chaque utilisateur en fonction des goûts/avis précédents de l'utilisateur et des avis d'autres utilisateurs qui lui sont semblables. Les principales étapes de cette approche sont :

1. de nombreuses préférences d'utilisateurs sont enregistrées ;
2. un sous-groupe d'utilisateurs est repéré dont les préférences sont similaires à celles de l'utilisateur qui cherche la recommandation ;
3. une *moyenne* des préférences pour ce groupe est calculée ;
4. la fonction de préférence qui en résulte est utilisée pour recommander des options/éléments à l'utilisateur qui cherche la recommandation.

Notons que la notion de similitude/similarité doit être définie.

Il est possible de distinguer trois fonctionnements différents : les approches *Item-to-Item* basées sur la similarité entre les éléments (*items*), les approches *User-to-User* basées sur la similarité entre les utilisateurs (*users*) et les autres approches. Nous illustrons la différence entre les deux premières dans l'exemple suivant.

Exemple 3.3.5– *Dans cet exemple, nous reprenons la matrice Utilisateurs × Films de l'exemple 3.3.1 qui indique les scores donnés par les utilisateurs Arnaud, Patrick, Marie et Elsa à des films.*

Dans l'approche Item-to-Item, les recommandations sont faites en trouvant les éléments qui ont le même intérêt pour plusieurs utilisateurs. Elsa et Marie aiment « l'Age de glace » et « Bienvenue chez les Ch'tis ». Cela suggère que, en général, les utilisateurs qui aiment « l'Age de glace » aimeront aussi « Bienvenue chez les Ch'tis », donc « Bienvenue chez les Ch'tis » pourra être recommandé à Arnaud (qui aime « l'Age de glace »). Notons que cette approche s'adapte à un nombre très important d'utilisateurs et/ou d'éléments.

Dans l'approche User-to-User, les recommandations sont faites en trouvant des utilisateurs ayant des avis similaires. Elsa et Marie aiment toutes les deux « L'Age de glace » et « Bienvenue chez les Ch'tis » et n'ont pas aimé « OSS 117 » ; il semble qu'elles ont des avis similaires, ce qui suggère qu'en général Marie et Elsa sont du même avis. Donc, « Harry Potter » est une bonne recommandation pour Elsa (puisque Marie l'aime). Notons que cette approche n'est pas adaptée à un nombre très important d'utilisateurs.

Les systèmes de recommandation collaboratifs, par leur diversité, s'appuient donc sur de nombreuses techniques, qu'il s'agisse de :

- similarité entre utilisateurs (coefficient de corrélation de Pearson [RN88], ...) ou de sélection de voisinage (les algorithmes basés sur la recherche de voisinage (kNN [AWY14], ...)) pour les approches *User-to-User* ;
- similarité entre éléments (la mesure de similarité cosinus [Qam10], ...) pour les approches *Item-to-Item* ;
- techniques de prédiction de scores (analyse en composantes principales, ACP [Pea01], factorisation de matrices (décomposition en valeurs singulières, SVD [GK65]), analyse sémantique latente, LSA [DDF+90], règles d'association, approches bayésiennes [FGG97], ...) pour les autres approches.

Les systèmes de recommandation collaboratifs ont comme avantages :

- d'utiliser les scores d'autres utilisateurs pour évaluer l'utilité des éléments ;

– de trouver des utilisateurs ou groupes d'utilisateurs dont les intérêts correspondent à l'utilisateur courant et,
– plus il y a d'utilisateurs plus il y a de scores : meilleurs sont alors les résultats.

Cependant, de tels systèmes ont aussi des inconvénients :
– trouver des utilisateurs ou groupes d'utilisateurs similaires est difficile et,
– le système de recommandation se heurte à la faible densité de la matrice *Utilisateurs* × *Eléments* ; de plus,
– il existe aussi le problème du démarrage à froid (*cold-start problem*) : lorsqu'un nouvel utilisateur utilise le système, ses préférences ne sont pas connues et lorsqu'un nouvel élément est ajouté au catalogue, personne ne lui a attribué de score,
– dans les systèmes avec un grand nombre d'éléments et d'utilisateurs, le calcul croît linéairement ; des algorithmes appropriés sont donc nécessaires,
– la « non-diversité » : il n'est pas utile de recommander tous les films avec l'acteur *Antonio Banderas* à un utilisateur qui a aimé l'un d'eux dans le passé.

3.3.1.3 Les approches hybrides

Un système de recommandation hybride utilise des composants ou la logique de différents types d'approches de recommandation (voir [Bur02] pour un aperçu). Par exemple, un tel système peut utiliser à la fois des connaissances extérieures et les caractéristiques des éléments combinant ainsi des approches collaboratives et basées sur le contenu [Neg15b].

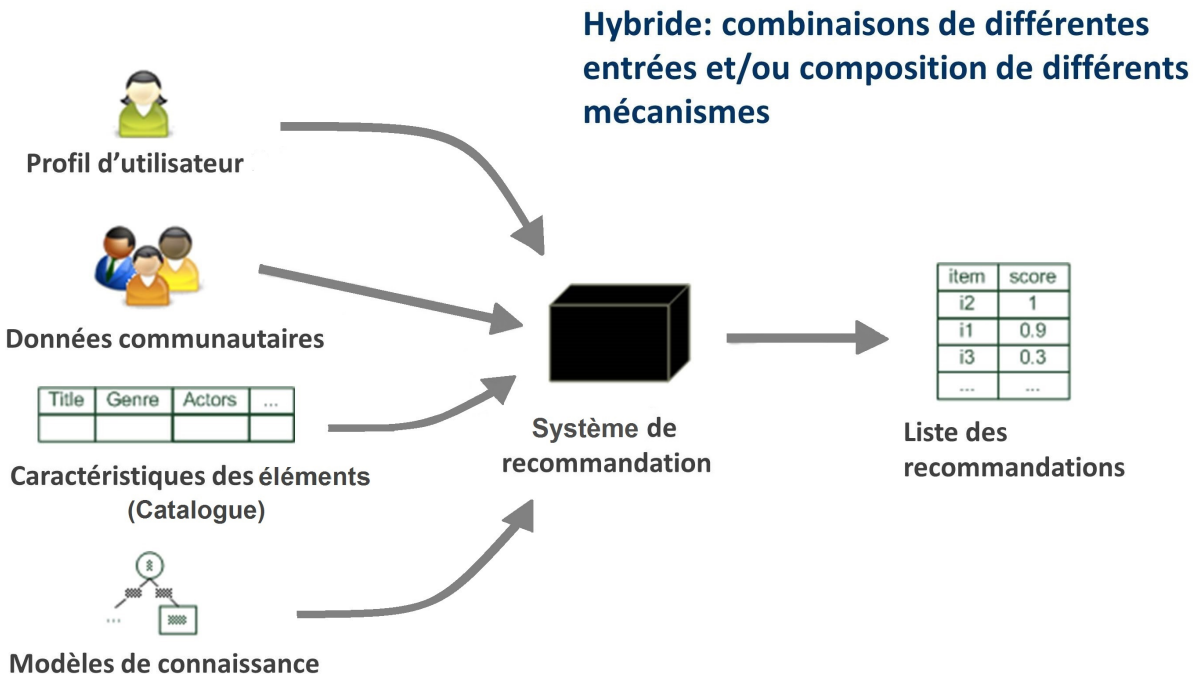

FIGURE 3.3 – Le système de recommandation hybride vu comme une boîte noire (adapté de [JZFF10])

Il est à noter que le terme « hybride » est un artefact de l'évolution historique des systèmes de recommandation où certains types de sources de connaissances ont été exploités en premier lieu, conduisant à des techniques bien établies qui ont ensuite été combinées. Finalement, si la recommandation est considérée comme un problème dont la solution peut s'appuyer sur des sources de connaissances multiples, alors la seule question qui se pose est : quelles sources sont les plus appropriées à une tâche donnée et comment elles peuvent être utilisées le plus efficacement possible. La figure 3.3 illustre ce processus. Notons que les approches hybrides étant une combinaison d'approches, elles ont les avantages tout en limitant les inconvénients des approches qui la composent [Neg15b].

Il existe trois grandes catégories de combinaisons de systèmes de recommandation pour concevoir un système de recommandation hybride [Bur02, JZFF10] : la combinaison monolithique (*monolithic hybridization design*), la combinaison parallèle (*parallelized hybridization design*) et la combinaison tubulaire (*pipelined hybridization design*).

Monolithique décrit une conception d'hybridation qui intègre les aspects de différentes stratégies de recommandation en un seul algorithme. Différents systèmes de recommandation y contribuent puisque l'approche hybride utilise des données d'entrée additionnelles qui sont spécifiques à un autre algorithme de recommandation, ou bien les données d'entrée sont complétées par une technique et exploitées par une autre. Par exemple, un système de recommandation basé sur le contenu qui exploite également des données communautaires pour déterminer des similarités entre éléments relève de cette catégorie.

Les deux autres approches hybrides nécessitent au moins deux mises en œuvre de recommandations séparées qui sont combinées en conséquence.

Sur la base de leurs données d'entrée, les systèmes hybrides de recommandation parallèles fonctionnent indépendamment l'un de l'autre et produisent des listes de recommandations distinctes. Dans une étape ultérieure d'hybridation, leurs sorties sont combinées en un ensemble final de recommandations.

Lorsque plusieurs systèmes de recommandation sont joints dans une architecture tubulaire, la sortie de l'un des systèmes de recommandation devient une partie des données d'entrée du système suivant.

3.4 Méthodes/techniques utilisées

Quelles que soient les approches évoquées précédemment, certaines mesures de similarité, métriques ou distances et autres techniques sont utilisées dans le cadre des systèmes de recommandation. Nous ne prétendons pas donner une liste exhaustive de toutes les méthodes existantes, mais nous tentons de donner un aperçu des méthodes les plus utilisées.

Une mesure de similarité est, en général, une fonction qui quantifie le rapport entre deux objets, comparés en fonction de leurs points de ressemblance et de dissemblance. Les deux objets comparés sont, bien entendu, de même type. Toutefois, il est important de noter qu'une valeur de mesure de similarité entre x et y peut être différente de celle entre y et x.

3.4.1 Modèle d'espace vectoriel

De manière générale, le profil de l'utilisateur, les données communautaires ainsi que les caractéristiques des éléments peuvent être représentés par des vecteurs. La représentation de ces différents

ensembles sous forme de vecteurs dans un espace vectoriel commun est connu sous le nom de modèle d'espace vectoriel (*vector space model*) [Neg15b].

Dans le cadre des systèmes de recommandation, il s'agit donc de représenter les éléments et les utilisateurs sous forme vectorielle.

Exemple 3.4.1 *Dans notre exemple, nous simplifions ici la représentation vectorielle en se limitant à des vecteurs de taille 4 (par souci de lisibilité et de facilité des calculs), en affichant des mots-clés communs (ici relatifs aux genres de livres/films) aux utilisateurs et aux éléments. Nous indiquons par la valeur 1 la présence du mot-clé dans les caractéristiques de l'élément ou dans le profil de l'utilisateur et par la valeur 0 son absence. A partir des profils de Marie (voir le tableau 3.1) et Arnaud (supposons qu'il aime le genre « Romance » et « Jeunesse ») et de l'extrait de catalogue de livres (voir le tableau 3.2), nous obtenons les représentations vectorielles suivantes :*

	Policier/Thriller	*Humour*	*Jeunesse*	*Romance*
c_1 : *Marie*	*1*	*1*	*1*	*0*
c_2 : *Arnaud*	*0*	*0*	*1*	*1*
i_1 : *Shining*	*1*	*0*	*0*	*0*
i_2 : *Le Journal de Bridget Jones*	*0*	*1*	*0*	*1*

Cette matrice va être utilisée tout au long de cette section.

3.4.2 Mesures de similarité

Nous présentons ici les mesures de similarité les plus utilisées/connues [Neg15b]. Il est à noter que les mesures et techniques présentées ci-après sont majoritairement basées sur le modèle vectoriel et/ou sont des approches syntaxiques[5]. Les avantages et inconvénients liés au modèle vectoriel ainsi que ceux liés aux approches syntaxiques sont présentés dans le tableau 3.5[6] [Neg15b].

3.4.2.1 Similarité cosinus

La similarité cosinus [Qam10] est fréquemment utilisée en tant que mesure de ressemblance entre deux documents [BYRN99]. Ici, il s'agit de calculer le cosinus de l'angle entre les représentations vectorielles des éléments ou des utilisateurs à comparer. La similarité obtenue $sim_{cosinus} \in [0, 1]$. Dans le cas de deux éléments i_1 et i_2, la formule est :

$$sim_{cosinus}(i_1, i_2) = \frac{\vec{i_1}.\vec{i_2}}{\|\vec{i_1}\|\|\vec{i_2}\|}$$

Exemple 3.4.2 *Pour les éléments i_1 (Shining) et i_2 (Le Journal de Bridget Jones), nous avons donc $sim_{cosinus}(i_1, i_2) = \frac{\vec{i_1}.\vec{i_2}}{\|\vec{i_1}\|\|\vec{i_2}\|} = \frac{0}{\sqrt{1}.\sqrt{2}} = 0$ qui montre que ces deux éléments n'ont rien en commun. De la même manière, il est possible de comparer les utilisateurs c_1 et c_2, tels que $sim_{cosinus}(c_1, c_2) \approx 0,408$ ou encore de calculer la similarité entre un élément et un utilisateur,*

5. Par exemple, une mesure de similarité syntaxique permet de comparer des documents textuels en se basant sur les chaînes de caractères qui les composent : les chaînes de caractères « voiture » et « voiturier » peuvent être considérées comme très proches, alors que « voiture » et « automobile » pourront être considérées comme très différentes. Une telle mesure sur les chaînes de caractères fournit une valeur obtenue algorithmiquement.

6. L'inconvénient majeur du modèle vectoriel peut être pallié par la lemmatisation, l'élimination des mots-vides et le TF-IDF.

	Avantages	Inconvénients
Modèle vectoriel	• Quelle que soit la technique utilisée, basée sur le modèle vectoriel, a de ce fait, le même format initial, à savoir, la représentation vectorielle. • Les techniques basées sur le modèle vectoriel sont faciles à développer, il s'agit uniquement de calcul vectoriel.	• Des mots identiques considérés comme peu pertinents peuvent parfois trop influer sur la valeur de la similarité.
Approches syntaxiques	• Les techniques basées sur l'approche syntaxique ne laissent pas de place aux exceptions. • Elles sont donc facilement automatisables.	• Par définition, les techniques basées sur l'approche syntaxique ne prennent pas en compte la sémantique.

TABLE 3.5 – Avantages et inconvénients liés au modèle vectoriel et aux approches syntaxiques

tel que $sim_{cosinus}(c_1, i_1) = 0,577$ *et* $sim_{cosinus}(c_2, i_1) = 0$ *qui montre que l'élément* i_1 *(Shining) est plus pertinent pour* c_1 *(Marie) que pour* c_2 *(Arnaud).*

3.4.2.2 Coefficient de corrélation de Pearson

Le coefficient de corrélation de Pearson [RN88] calcule la similarité entre deux éléments ou deux utilisateurs comme le cosinus de l'angle entre leurs représentations vectorielles centrées-réduites. La similarité obtenue $sim_{pearson} \in [-1, 1]$. Dans le cas de deux utilisateurs c_1 et c_2, la formule est :

$$sim_{pearson}(c_1, c_2) = sim_{cosinus}(c_1 - \overline{c_1}, c_2 - \overline{c_2})$$

où $\overline{c_1}$ (resp. $\overline{c_2}$) représente la moyenne de c_1 (resp. c_2).

Exemple 3.4.3 *En gardant des vecteurs de taille* 4*, la moyenne pour* c_1*,* $\overline{c_1} = \frac{3}{4}$*, pour* c_2*,* $\overline{c_2} = \frac{1}{2}$ *et pour* i_1*,* $\overline{i_1} = \frac{1}{4}$*, nous avons* $sim_{pearson}(c_1, c_2) = sim_{cosinus}(c_1 - \overline{c_1}, c_2 - \overline{c_2}) = \frac{-\frac{1}{2}}{\sqrt{\frac{3}{4}}.\sqrt{1}} \approx -0,577$*. De la même manière,* $sim_{pearson}(c_1, i_1) \approx 0,333$ *et* $sim_{pearson}(c_2, i_1) \approx -0,577$*. Cela montre, selon le coefficient de corrélation de Pearson, que l'élément* i_1 *(Shining) est plus pertinent pour* c_1 *(Marie).*

Si nous nous intéressons aux performances de chaque mesure, [Hua08] et [SGM00] ont tous les deux montré que les performances de la similarité cosinus et du coefficient de Pearson sont très proches (quelle que soit la densité de la matrice d'utilité pour des volumes de données assez grands).

3.4.3 Réduction de dimensions

Dans le cadre des systèmes de recommandation où la problématique peut se résumer à l'estimation de scores dans une matrice d'utilité (*Utilisateurs* × *Eléments*), un constat s'impose : la matrice peut contenir un nombre très grand d'utilisateurs et/ou d'éléments [Neg15b]. Or cela peut

poser un problème lors de l'exploration et l'analyse de ces données. Pour cela, il est fondamental de mettre en place des outils de traitement de données permettant une meilleure compréhension de la valeur des connaissances disponibles dans ces données.

La réduction de dimensions est l'une des plus vieilles approches permettant d'apporter une réponse à ce problème. Son objectif est de sélectionner ou d'extraire un sous-ensemble optimal de la matrice selon un critère défini à l'avance. La sélection de ce sous-ensemble permet d'éliminer les informations non-pertinentes et redondantes selon le critère utilisé. Cette sélection/extraction permet donc de réduire la dimension de l'espace de recherche et de rendre l'ensemble des données plus représentatif du problème [Fod02, Gué06].

3.4.3.1 Analyse en composantes principales

L'Analyse en composantes principales (ACP) [Pea01, HZC04] est une méthode d'analyse des données issue de la statistique multivariée, qui consiste à transformer des variables liées entre elles (dites « corrélées » en statistique) en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées « composantes principales », ou axes principaux. L'ACP, en tant que technique de réduction de dimensions, permet de réduire le nombre de variables et de rendre l'information moins redondante (voir [wik15] pour une introduction détaillée). Il s'agit d'une approche à la fois géométrique et statistique.

Mathématiquement, l'ACP est un simple changement de base : passer d'une représentation dans la base canonique [7] des données initiales à une représentation dans la base de facteurs définis par les vecteurs propres de la matrice des corrélations.

3.4.3.2 Décomposition en valeurs singulières

Le procédé de décomposition en valeurs singulières (*Singular Value Decomposition* - SVD) [GK65] d'une matrice est un outil important de factorisation de matrices rectangulaires réelles ou complexes. La SVD est un procédé de réduction de données [Bak05].

Mathématiquement, soit M une matrice $n \times p$ (à n lignes et p colonnes), la décomposition SVD (voir [JZFF10], p.28, pour un exemple de SVD dans le cas des systèmes de recommandation pour la matrice d'utilité $Utilisateurs \times Eléments$) est un algorithme de factorisation qui permet d'exprimer M comme le produit de trois matrices particulières, U, Σ et V telles que :

$$M = U\Sigma V^T$$

avec :

- U : matrice m × m orthogonale [8]
- Σ : matrice m × n à diagonale positive [9]
- V^T : matrice n × n orthogonale [10]

7. Dans l'espace des matrices à n lignes et p colonnes, la base canonique est l'ensemble des matrices $M_{i,j}$ qui présentent un 1 à l'intersection de la $i^{ème}$ ligne avec la $j^{ème}$ colonne et 0 partout ailleurs.

8. Une matrice réelle A carrée d'ordre n est dite orthogonale si elle vérifie l'une des propriétés équivalentes suivantes :
   - ${}^TAA = I_n$;
   - $A^TA = I_n$;
   - A est inversible et $A^{-1} = {}^TA$.

avec I_n : matrice Identité d'ordre n.

9. Une matrice à diagonale positive contient des réels positifs ou nuls sur sa diagonale et toutes les autres cellules contiennent des valeurs nulles.

10. La matrice transposée V^T, d'une matrice V, est obtenue en échangeant les lignes et les colonnes.

Notons que, par convention, les valeurs Σ_i sont rangées par ordre décroissant de i. Alors, la matrice Σ est déterminée de façon unique par M (mais U et V ne le sont pas).

En utilisant des techniques de réduction de dimensions telles que l'ACP ou la SVD, nous réduisons le bruit des données à dimensions élevées, nous améliorons l'évolutivité (passage à l'échelle) et nous abordons le problème de la faible densité de la matrice d'utilité. Mathématiquement, l'ACP et la SVD ont des propriétés communes. Leurs performances diffèrent donc de peu. Notons cependant que pour des dimensions supérieures à 3, la SVD perd en précision par rapport à l'ACP [SKKR00, kBGM15].

3.4.4 Classification

La classification, par définition, permet de classer des utilisateurs ou des éléments ; c'est un système de classement organisé et hiérarchisé de catégorisation d'utilisateurs ou d'éléments. On distingue deux types de classification : la classification supervisée et la classification non-supervisée.

3.4.4.1 Classification supervisée

L'objectif de la classification supervisée est de définir des règles permettant de classer des utilisateurs ou des éléments dans des classes à partir de variables qualitatives ou quantitatives les caractérisant. On dispose au départ d'un échantillon dit d'apprentissage dont le classement est connu, c'est-à-dire les classes dans lesquelles les utilisateurs ou les éléments vont être répartis sont connues à l'avance. Cet échantillon est utilisé pour l'apprentissage des règles de classement. Notons que traditionnellement, un deuxième échantillon indépendant, dit de validation ou de test est utilisé pour valider la classification.

Parmi les méthodes/techniques de classification supervisée connues, nous pouvons citer :

- la méthode des k plus proches voisins (*k-Nearest Neighbors* - kNN) [CH06]. Cette méthode simple consiste à comparer un utilisateur/élément de classe inconnue à tous les utilisateurs/éléments stockés. Il s'agit ensuite de choisir pour ce nouvel utilisateur/élément, la classe majoritaire parmi ses k plus proches voisins. Plus formellement, à partir des utilisateurs/éléments x' de l'ensemble d'apprentissage et d'un nouvel utilisateur/élément X que nous voulons classer :

  1. Déterminer les k plus proches utilisateurs/éléments de X en calculant la distance entre X et tous les x' ;
  2. Compter les occurrences de chaque classe pour les k plus proches utilisateurs/éléments ;
  3. Attribuer à X la classe la plus fréquente.

  Notons que l'algorithme nécessite de connaître k, le nombre de voisins à considérer, qui peut être déterminé par validation croisée. De même l'algorithme nécessite une distance entre utilisateurs/éléments, qui peut être mesurée selon différentes méthodes dont celles qui ont été présentées en début de section.

- les arbres de décision [Qui86].
  Un arbre de décision est une représentation graphique d'une procédure de classification, plus particulièrement sous forme d'arbre, où les extrémités de chaque branche (ou feuilles) représentent les différents résultats possibles en fonction des décisions prises à chaque étape.

Appliqué à la recommandation, les nœuds de l'arbre représentent les caractéristiques des éléments et sont utilisés pour partitionner les données, par exemple, en fonction de l'existence ou de la non-existence d'un mot-clé dans le profil.

3.4.4.2 Classification non-supervisée

Contrairement à la classification supervisée, où les classes sont connues au préalable et où l'objectif est de construire une règle de classement pour prédire comment de nouvelles observations (utilisateurs ou éléments) vont être classées ; en classification non-supervisée, les classes ne sont pas connues au préalable. L'objectif de la classification non-supervisée (*clustering*) est de trouver comment classer les observations (utilisateur ou éléments) à partir des descripteurs disponibles (voir [JD88] pour plus de détails).

Parmi les méthodes/techniques de classification non-supervisée connues, nous pouvons citer : la méthode de partitionnement des Kmeans [HW79]. Il est supposé l'existence de K classes distinctes puis l'algorithme s'appuie sur K centres [11] de classes $\mu_1, ..., \mu_K$ parmi les utilisateurs ou les éléments. Puis les étapes suivantes sont réalisées itérativement :

1. Pour chaque utilisateur/élément qui n'est pas un centre de classe, l'algorithme repère le centre de classe le plus proche et définit ainsi K classes $P_1, ..., P_K$, où $Pj = \{ensemble\ des\ points\ les\ plus\ proches\ du\ centre\ \mu_j\}, \forall j \in [1, K]$;
2. Dans chaque nouvelle classe P_j, l'algorithme définit le nouveau centre de classe μ_j comme étant le barycentre des points de P_j.

L'algorithme s'arrête selon un critère d'arrêt fixé : soit le nombre limite d'itérations est atteint, soit l'algorithme a convergé, c'est-à-dire qu'entre deux itérations les classes formées restent les mêmes, soit l'algorithme a « presque » convergé, c'est-à-dire que l'inertie intra-classe ne s'améliore quasiment plus entre deux itérations.

La classification supervisée et la classification non-supervisée ont des performances similaires avec des données à forte dimensionnalité comme avec des données à faible dimensionnalité [BG09]. Elles sont principalement utilisées avec des approches à base de filtrage collaboratif. Notons cependant que la classification non-supervisée présente des avantages reconnus (limite de la surcharge d'informations, réduction de la dimensionnalité, ...) pour les systèmes de recommandation [PZW11, SM11].

3.4.5 TF-IDF

Outre les mesures de similarité, les techniques de réduction de dimensions et les méthodes de classification, il existe d'autres approches. Nous présentons ici le TF-IDF.

Dans le contexte des systèmes de recommandation, le TF-IDF (*Term Frequency-Inverse Document Frequency*) permet d'évaluer l'importance d'un terme contenu dans un profil utilisateur ou dans les caractéristiques d'un élément, relativement à un ensemble d'utilisateurs ou à un catalogue d'éléments. Le poids augmente proportionnellement avec le nombre d'occurrences du terme dans le profil de l'utilisateur ou les caractéristiques de l'élément. Il varie également en fonction de la fréquence du terme dans l'ensemble des utilisateurs ou du catalogue. Ainsi, la fréquence inverse du profil utilisateur (resp. caractéristique de l'élément) (*idf*) est une mesure de l'importance du terme dans l'ensemble des utilisateurs (resp. du catalogue).

11. Ces centres peuvent être soit choisis pour leur représentativité, soit désignés aléatoirement.

Dans le cas du TF-IDF, elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme les plus discriminants. Il s'agit de calculer le logarithme de l'inverse de la proportion des profils utilisateurs (resp. des caractéristiques des éléments) qui contiennent le terme : $idf_k = log\frac{|C|}{|\{c_j:t_k\in c_j\}|}$ (resp. $idf_k = log\frac{|I|}{|\{i_j:t_k\in i_j\}|}$)où $|C|$ (resp. $|I|$) est le nombre total de (profils) utilisateurs (resp. nombre total d'éléments) et $|\{c_j : t_k \in c_j\}|$ (resp. $|\{i_j : t_k \in i_j\}|$) est le nombre de profils (resp. de caractéristiques d'éléments) où le terme t_k apparaît. Finalement, le poids s'obtient en multipliant les deux mesures : $tfidf_{k,j} = tf_{k,j}.idf_k$.

Notons que les nouveaux vecteurs obtenus par le TF-IDF peuvent être la base de calcul des mesures de similarité qui comparent des vecteurs (comme celles vues précédemment).

Exemple 3.4.4 *Si nous nous intéressons aux utilisateurs (et nous nous restreignons toujours à quatre termes), notre exemple contient deux profils (c_1 et c_2), donc $|C| = 2$. Pour le terme « Thriller/Policier », seul le profil de c_1 le contient. Par conséquent, $idf_{Thriller/Policier} = log(\frac{2}{1}) \approx 0,301$. De même, le terme « Jeunesse » apparaît dans les deux profils, donc $idf_{Jeunesse} = log(\frac{2}{2}) = 0$. Le calcul du TF-IDF du terme « Thriller/Policier » est donc : $tfidf_{Thriller/Policier,c_1} = tf_{Thriller/Policier,c_1}.idf_{Thriller/Policier} = \frac{1}{4}.log\frac{2}{1} \approx 0,075$ pour le profil de c_1 et $tfidf_{Thriller/Policier,c_2} = tf_{Thriller/Policier,c_2}.idf_{Thriller/Policier} = 0.log\frac{2}{1} = 0$ pour le profil de c_2. Celui du terme « Jeunesse » est donc : $tfidf_{Jeunesse,c_1} = tf_{Jeunesse,c_1}.idf_{Jeunesse} = \frac{1}{4}.log\frac{2}{2} = 0$ pour le profil de c_1 et $tfidf_{Jeunesse,c_2} = tf_{Jeunesse,c_2}.idf_{Jeunesse} = \frac{1}{4}.log\frac{2}{2} = 0$ pour le profil de c_2.*

Les représentations vectorielles de c_1 et c_2 avec le TF-IDF sont :

	Thriller/Policier	*Humour*	*Jeunesse*	*Romance*
c_1	$0,075$	$0,075$	*0*	*0*
c_2	*0*	*0*	*0*	$0,075$

Il est à noter que le TF-IDF est plutôt utilisé dans les approches à base de contenu et permet d'obtenir de bonnes performances [NR17]. Il est cependant essentiellement utilisé en prétraitement (nettoyage, réduction de dimensions, ...) pour une utilisation par d'autres techniques mathématiques.

3.5 Evaluation de tels systèmes

Les techniques et systèmes évoluent au cours du temps en essayant d'être de plus en plus proches des attentes/besoins des utilisateurs. Pour cela, il est nécessaire d'évaluer les systèmes de recommandation et ainsi vérifier s'ils sont pertinents pour l'utilisateur en fonction du contexte, des objectifs à atteindre, du temps de réponse, de la prise en compte de certains critères, ... [Neg15b]

3.5.1 Jeux de données, densité et erreurs

Dans le cadre des systèmes de recommandation, l'intérêt se porte sur des groupes spécifiques d'une population (des acheteurs en ligne, des utilisateurs du web, ...) à qui il faut proposer/suggérer des éléments adaptés et personnalisés. De plus, nous avons besoin d'un jeu de données qui peut être synthétique ou, mieux, des données « historiques » d'interaction d'utilisateurs avec un système, c'est-à-dire une collection de profils utilisateurs avec des préférences, des scores, des transactions, ... Utiliser un même jeu de données pour évaluer différents systèmes de recommandation facilite la comparaison des performances de ces différents systèmes.

Il faut cependant être attentif à la densité du jeu de données qui correspond au rapport entre les cellules vides et les cellules remplies de la matrice $Utilisateurs \times Eléments$, de la façon suivante :

		L'élément a été proposé par le système de recommandation	
		Oui	Non
L'élément a été « aimé » par l'utilisateur	Oui	Prédictions correctes	Faux négatifs
	Non	Faux positifs	Omissions correctes

TABLE 3.6 – Types d'erreurs

$$densité = 1 - \frac{|R|}{|I|.|C|}$$

où R est l'ensemble des scores, I l'ensemble des éléments et C l'ensemble des utilisateurs qui font partie du jeu de données. Notons que la valeur de la densité appartient à l'intervalle $[0, 1]$ où une valeur proche de 0 indique une forte densité et une valeur proche de 1 indique une faible densité du jeu de données.

Cependant, les résultats de l'évaluation des systèmes de recommandation *via* des jeux de données « historiques » ne peuvent pas être comparés aux études réalisées avec de vrais utilisateurs. En effet, comme illustré par le tableau 3.6 (traduit de [JZFF10]) :

- si l'élément recommandé était réellement pertinent pour l'utilisateur alors c'est une prédiction correcte ;
- si l'élément recommandé n'était pas indiqué comme pertinent pour l'utilisateur alors il peut s'agir d'un faux positif car le système propose un élément qui n'est pas indiqué, dans l'historique, comme pertinent mais cela peut être dû au fait que l'utilisateur ne connaissait pas cet élément et peut-être que l'utilisateur l'aurait trouvé pertinent s'il en avait eu connaissance ;
- de même, si l'élément n'est pas recommandé, alors il est difficile de déterminer si l'utilisateur aurait indiqué l'élément non proposé comme pertinent ; dans ce cas, il s'agit d'un faux négatif ;
- si l'élément n'était pas recommandé et qu'il n'était pas indiqué comme pertinent pour l'utilisateur alors c'est une omission correcte.

3.5.2 Mesures

Il existe de nombreuses mesures pour évaluer la qualité des systèmes de recommandation. Les plus utilisées sont les mesures de justesse (*accuracy*) (voir [HKTR04] pour plus de détails). Ici, nous nous focalisons sur les mesures les plus connues pour l'évaluation de systèmes de recommandation à partir de jeux de données.

3.5.2.1 Justesse

Il existe trois types de justesse : la justesse de la prédiction des recommandations, la justesse de la classification des recommandations et la justesse de l'ordonnancement des recommandations.

3.5.2.1.1 Justesse de la prédiction

La justesse de prédiction permet d'évaluer la capacité d'un système de recommandation à prédire correctement l'avis d'un utilisateur pour un élément donné.

Issues du domaine statistique, les mesures les plus utilisées sont l'erreur absolue moyenne (*Mean Absolute Error*, MAE) et la racine carrée de l'erreur quadratique moyenne (*Root Mean Square Error*, RMSE). Dans le cadre des systèmes de recommandation, ces mesures sont utilisées pour mesurer la

différence entre les scores de recommandations prédits, $reco(c,i)$ et les scores réels, $r_{c,i}$ pour tous les utilisateurs $c \in C$ et tous les éléments testés $i \in Itest_c$.

MAE calcule donc la déviation moyenne, l'écart, entre les scores prédits et les scores réels, telle que :

$$MAE = \frac{\sum_{c \in C} \sum_{i \in Itest_c} |reco(c,i) - r_{c,i}|}{\sum_{c \in C} |Itest_c|}$$

RMSE, quant à elle, est la racine carrée de la moyenne (arithmétique) des carrés des écarts entre les scores prédits et les scores réels, c'est-à-dire l'amplitude moyenne de l'erreur. Elle est similaire à la MAE, mais accentue les plus grands écarts (elle donne un poids relativement plus élevé aux erreurs importantes) :

$$RMSE = \sqrt{\frac{\sum_{c \in C} \sum_{i \in Itest_c} (reco(c,i) - r_{c,i})^2}{\sum_{c \in C} Itest_c}}$$

Les valeurs de MAE et RMSE appartiennent à l'intervalle $[0, +\infty)$ et sont orientées négativement, c'est-à-dire les résultats sont bons lorsque les valeurs sont petites. Afin de faciliter l'interprétation des résultats, ces mesures peuvent être normalisées [12]. Il existe donc NMAE et NRMSE qui sont les mesures normalisées de MAE et RMSE.

3.5.2.1.2 Justesse de la classification

Le but d'une tâche de classification dans le cadre des systèmes de recommandation est d'identifier les éléments les plus pertinents pour un utilisateur donné. Le rappel et la précision sont les deux mesures les plus connues (issues du domaine de la recherche d'informations [BYRN99]).

La précision, $Précision_c$, calcule le nombre de succès pour l'utilisateur c, $succès_c$, c'est-à-dire le nombre d'éléments pertinents correctement recommandés pour l'utilisateur c, par rapport au nombre total d'éléments recommandés à l'utilisateur c. La précision est telle que :

$$Précision_c = \frac{Nombre\ de\ succès\ pour\ c}{Nombre\ total\ d'éléments\ recommandés\ à\ c}$$

En revanche, le rappel, $Rappel_c$, calcule le rapport entre le nombre de succès pour c, $succès_c$ et le nombre maximal théorique de succès, $succès_T$ au regard de la taille de l'ensemble de test. En fait, le rappel est défini par le nombre d'éléments pertinents recommandés en fonction du nombre d'éléments pertinents que possède l'ensemble de test. Le rappel est tel que :

$$Rappel_c = \frac{Nombre\ de\ succès\ pour\ c}{Nombre\ total\ d'éléments\ pertinents\ existants} = \frac{|succès_c|}{|succès_T|}$$

Un système de recommandation parfait fournit des recommandations dont la précision et le rappel sont égaux à 1 (l'algorithme trouve la totalité des éléments pertinents (rappel) et ne fait aucune erreur (précision)). Cependant, les algorithmes sont plus ou moins précis, et plus ou moins pertinents. Il arrive donc souvent d'avoir un système très précis mais peu performant (précision $\approx 0,99$, rappel $\approx 0,10$) ou très performant mais peu précis (précision $\approx 0,10$, rappel $\approx 0,99$). Pour pallier ce problème, il est préférable d'utiliser la $F\text{-}Mesure$ qui combine la précision et le rappel, telle que :

$$F_c = \frac{2 \cdot (Précision_c \cdot Rappel_c)}{(Précision_c + Rappel_c)}$$

12. Une mesure est normalisée lorsque ses valeurs appartiennent à l'intervalle de valeurs dans $\mathbb{R}$, $[0, 1]$.

3.5.2.1.3 Justesse d'ordonnancement

Les mesures de justesse d'ordonnancement étendent les résultats des mesures de justesse de classification en prenant en compte la position relative des succès dans la liste des recommandations. Nous détaillons ici les mesures de justesse d'ordonnancement les plus connues : le *Rank score* [BHK98] et le *Lift index* [LLL98].

Le *Rank score* ou *R-Score* [BHK98] étend la mesure de rappel en prenant en compte les positions des éléments corrects dans une liste ordonnée (cela est important dans les systèmes de recommandation car des éléments situés en bas de la liste peuvent être négligés par les utilisateurs). Le *Rank Score* est défini comme le rapport entre le score des éléments corrects et le meilleur score théorique réalisable, comme :

$$RankScore = 100.\frac{\sum_c RankScore_c}{\sum_c RankScore_c^{max}}$$

avec $Rankscore_c = \sum_{i \in succès_c} \frac{max(r_{c,i}-d;0)}{2^{\frac{rank(i)-1}{\alpha-1}}}$ où d correspond à un vote neutre et α correspond au rang de l'élément dans la liste tel qu'il y a une probabilité de 50-50 que l'utilisateur note cet élément ; et $RankScore_c^{max}$ est le score maximal atteignable si tous les éléments recommandés à l'utilisateur c avaient été en première position de la liste des recommandations (ordonnés par valeur de vote).

Le *Lift Index* [LLL98] divise la liste ordonnée des recommandations en dix déciles [13] D_k égaux et compte le nombre de succès pour l'utilisateur c dans chaque décile avec $\sum_{k=1}^{10} D_k = |succès_c|$, tel que :

$$liftIndex_c = \begin{cases} \frac{1.D_{1,c}+0,9.D_{2,c}+...+0,1.D_{10,c}}{\sum_{k=1}^{10} D_{k,c}} & si\ |succès_c| > 0 \\ 0 & sinon \end{cases}$$

Notons que, par rapport au *Rank score*, le *Lift index* attribue encore moins de poids aux succès situés en haut de la liste.

3.5.2.2 Autres mesures

Il existe, bien entendu, un très grand nombre de mesures pour quantifier la qualité des systèmes de recommandation et toutes ne sont pas présentées ici. En plus des mesures de justesse, nous pouvons également citer les mesures basées sur :

- la couverture des utilisateurs (*user coverage*) [AZM14], lorsque de très nombreux utilisateurs C utilisent le système et qu'il s'agit de vérifier l'attitude du système de recommandation face à de nouveaux utilisateurs avec peu de scores, comme :
$$Ucov = \frac{\sum_{c \in C} \rho_c}{|C|} \text{ où } \rho_c = \begin{cases} 1 & si\ le\ nombre\ de\ recommandations\ > 0 \\ 0 & sinon \end{cases}$$
- la diversité des recommandations, comme la similarité intra liste (*Intra-List Similarity* - ILS) [ZMKL05] qui calcule la similarité deux à deux des éléments recommandés et plus l'ILS est faible, plus les éléments recommandés sont diversifiés.

13. En statistique descriptive, un décile est chacune des neuf valeurs qui divisent un jeu de données, triées selon une relation d'ordre, en dix parts égales, de sorte que chaque partie représente $\frac{1}{10}$ de l'échantillon de population. Dans le cas de recommandations ordonnées, les déciles sont les valeurs qui partagent l'ensemble de ces recommandations en dix parties égales, de sorte que le premier décile correspond au score au-dessous duquel se situent 10 % des scores, le neuvième décile est le score au-dessous duquel se situent 90 % des scores.

3.6 Des définitions à nos travaux

Dans ce chapitre, après avoir distingué la recommandation de la personnalisation, nous avons abordé différentes approches pour proposer des recommandations à un utilisateur. Les trois approches les plus courantes sont les approches basées sur le contenu, les approches collaboratives et les approches hybrides (qui sont une combinaison des autres). Puis, nous avons précisé différentes techniques utilisées dans le cadre des systèmes de recommandation, que ce soit en mesurant la similarité entre utilisateurs, entre éléments ou entre utilisateurs et éléments (Similarité cosinus, Coefficient de corrélation de Pearson), ou en réduisant le nombre de dimensions afin de faciliter l'analyse des relations entre les utilisateurs, entre les éléments ou entre les utilisateurs et les éléments (ACP, SVD), ou en classifiant/partitionnant des utilisateurs ou des éléments (classification supervisée et non-supervisée) ou encore, via d'autres techniques (TF-IDF). Puis, nous avons présenté différentes techniques pour évaluer la qualité des systèmes de recommandation. A partir d'un jeu de données (de préférence « historique ») plus ou moins dense, il est possible de mesurer la justesse de précision (MAE, RMSE), la justesse de la classification (précision, rappel, F-mesure), la justesse d'ordonnancement (*Rank score, Lift index*) ou encore la couverture des utilisateurs (UCov) ou la diversité des recommandations proposées (ILS). Notons que chacune des techniques a des caractéristiques propres et qu'elles sont complémentaires les unes des autres.

Ainsi, les systèmes de recommandation sont historiquement plutôt basés sur le contenu ou sur le filtrage collaboratif et suggèrent des éléments ou des utilisateurs. Ils sont « sur-spécialisés » et de ce fait, spécifiques au domaine d'application pour lequel ils ont été définis.

Parmi les différentes approches de recommandation existantes, nous nous sommes focalisés sur les approches hybrides qui sont, pour rappel, une combinaison d'approches et qui, en général, permettent de tirer parti des avantages des approches combinées en limitant leurs inconvénients. De plus, nous ne voulons pas nous limiter quant à la nature des recommandations : les approches doivent pouvoir recommander des éléments et/ou des utilisateurs. D'autre part, nous avons classé (voir dans le chapitre précédent) les recommandations en fonction du domaine d'application et du destinataire de la recommandation. En outre, au vu du nombre de techniques d'évaluation existantes, tout système de recommandation peut être amélioré.

Ainsi, notre objectif est de présenter nos travaux sur les systèmes de recommandation hybrides selon trois dimensions/axes, à savoir :

1. la nature de la recommandation : un élément ou un utilisateur,
2. la classe à laquelle appartient le recommandation : recommandation individuelle ou recommandation à visée publique,
3. l'amélioration proposée des recommandations.

Chaque système de recommandation proposé peut être visualisé selon ces trois axes comme illustré en 3D par la figure 3.4 et en 2D par la figure 3.5. Notons que les cases vides indiquent que nous n'avons pas (encore) traité ces cas.

Finalement, la « sur-spécialisation » va à l'encontre des principes de l'ingénierie et de la qualité logicielle (prônant, entre autres, la généricité des systèmes informatisés). Nous souhaitons donc offrir un système générique de recommandation (adaptatif avec un niveau d'abstraction élevé), capable de recommander des éléments et/ou des utilisateurs quel que soit le domaine d'application. En effet,

à notre connaissance, il n'existe pas de système générique de recommandation utilisable quel que soit le domaine d'application.

Dans le chapitre 4, nous présentons une approche générique de recommandation pouvant être instanciée pour les recommandations *individuelles* d'éléments et/ou d'utilisateurs et pour les recommandations *à visée publique* d'éléments et/ou d'utilisateurs, sans amélioration (cela correspond à la partie supérieure de la figure 3.4).

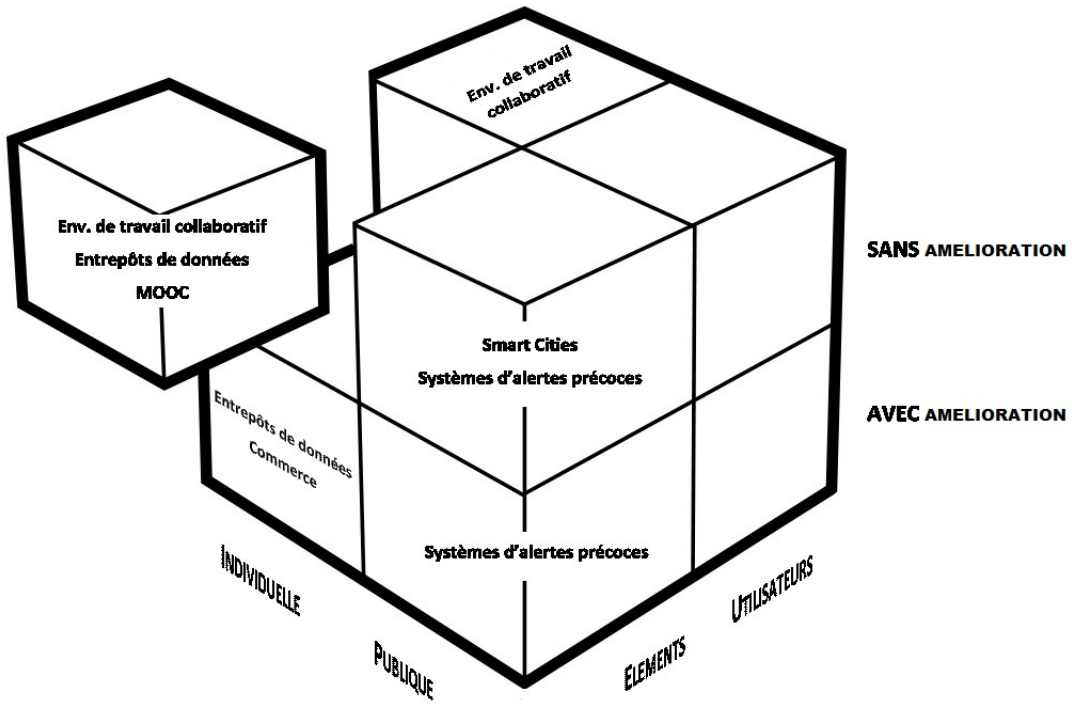

FIGURE 3.4 – Intersection de nos travaux selon les trois axes (nature, classe et contexte) - Visualisation en 3D

	AVEC amélioration		SANS amélioration	
	Individuelle	Publique	Individuelle	Publique
Element(s)	Entrepôts de données Commerce	Systèmes d'alertes précoces	Environnements de travail collaboratif Entrepôts de données MOOC	Smart Cities Systèmes d'alertes précoces
Utilisateur(s)			Environnements de travail collaboratif	

FIGURE 3.5 – Intersection de nos travaux selon les trois axes (nature, classe et contexte) - Visualisation en 2D

Chapitre 4

Un système générique de recommandation

Sommaire

4.1 Système générique de recommandation 42
4.2 Applications 46
4.2.1 Recommandations individuelles 46
4.2.1.1 Recommander des requêtes OLAP 46
4.2.1.2 Recommander des supports intéressants dans les plateformes d'apprentissage numérique 49
4.2.1.3 Recommander des experts dans les environnements de travail collaboratif 56
4.2.2 Recommandations à visée publique 62
4.2.2.1 Recommander des actions à mettre en place dans les villes 62
4.2.2.2 Recommander des actions à mettre en place pour la gestion de crises (système d'alertes précoces) 69
4.3 Bilan 74

Dans ce mémoire, nous présentons nos travaux de recherche sur les systèmes de recommandation hybrides selon trois axes (voir les figures 3.4 et 3.5) : la nature de la recommandation (élément ou utilisateur), sa classe d'appartenance (individuelle ou à visée publique) et l'amélioration proposée.

Dans ce chapitre, nous nous intéressons aux recommandations d'éléments et/ou d'utilisateurs, qu'elles soient individuelles ou à visée publique, sans amélioration. Les recommandations individuelles s'adressent à un utilisateur (ou groupe d'utilisateurs) et seront utilisées par l'utilisateur (ou au groupe) pour prendre des décisions qui le concerne. Les recommandations à visée publique s'adressent à un décideur (ou groupe de décideurs) mais seront utilisées pour prendre des décisions qui auront un impact sur les populations.

Afin de pallier la « sur-spécialisation » des systèmes de recommandation existants, nous proposons un système générique de recommandation (d'éléments et/ou d'utilisateurs) permettant de proposer des recommandations individuelles mais également des recommandations à visée publique, quel que soit le domaine d'application. Ce chapitre est organisé de la manière suivante : la section 4.1 présente notre système générique de recommandation, puis dans la section suivante, nous illustrons son applicabilité pour la recommandation individuelle dans des domaines tels que OLAP, les plateformes d'apprentissage numérique et les environnements de travail collaboratif ainsi que pour la recommandation à visée publique dans les villes « intelligentes » et la gestion de crises.

4.1 Système générique de recommandation

Pour rappel, la recommandation peut se résumer par le problème d'estimation de scores pour des entités qui n'ont pas encore été vues/évaluées par un utilisateur. En effet, le nombre d'entités ainsi que le nombre d'utilisateurs du système peuvent être très importants ; il est, de ce fait, difficile que chaque utilisateur voit toutes les entités ou que chaque entité soit évaluée par tous les utilisateurs. Il est donc nécessaire d'estimer les scores pour les entités non encore évaluées.

Cette évaluation est habituellement basée sur les scores donnés par un utilisateur à d'autres entités. Lorsqu'il est possible d'estimer les scores pour les entités non encore évaluées, les entités ayant les scores les plus élevés peuvent être recommandées à l'utilisateur.

Les systèmes de recommandation existants sont spécifiques à chaque application. Or, les systèmes de recommandation, comme tout système informatisé, sont régis par les principes de l'ingénierie et de la qualité logicielle. La généricité, selon laquelle un système réutilisable/adaptable a beaucoup plus de valeur qu'un système dédié, est l'un de ces principes [GJM02]. Ainsi, la spécialisation des systèmes de recommandation va à l'encontre du principe de généricité et donc, de l'ingénierie et de la qualité logicielle. Par conséquent, dans cette section nous proposons un système générique de recommandation, c'est-à-dire qui fonctionne quel que soit le cas d'application afin d'avoir un système adaptatif, avec un niveau d'abstraction, favorisant, entre autres, l'interopérabilité et la réutilisabilité. A notre connaissance, il n'existe pas, à ce jour, un tel système de recommandation.

Ainsi, notre approche générique estime les scores des entités non encore évaluées et recommande les entités ayant les scores les plus adaptés au besoin de l'utilisateur. Par analogie avec la définition 3.3.1 de [AT05], nous proposons la définition suivante dans un cadre générique :

Définition 4.1.1 – *Recommandation « générique »*
Soit C l'ensemble de tous les utilisateurs/décideurs (une personne seule ou un groupe de personnes), I l'ensemble de toutes les entités (éléments ou personnes) possibles qui peuvent être recommandées et soit u une fonction qui mesure l'utilité d'une entité i à l'utilisateur c, c'est-à-dire $u : C \times I \rightarrow \mathbb{R}$. [1]
Alors pour chaque utilisateur $c \in C$, nous voulons choisir l'entité $i' \in I$ qui maximise l'utilité à l'utilisateur : $\forall c \in C, i'_c = argmax_{i \in I} u(c, i)$.

Notre approche générique utilise donc à la fois : l'ensemble des utilisateurs, l'ensemble des entités recommandables, la matrice d'utilité et un log relatif à l'ensemble des entités recommandables, c'est-à-dire un historique contenant les précédentes interactions entre les utilisateurs du système et les entités recommandables (voir la partie *Entrées*, en violet, de la figure 4.1).

A partir de la définition 4.1.1, nous avons élaboré la fonction suivante : $GenericReco(C, I, Log, Mu, PretraitementL, PretraitementM, Generation, Ordonnancement, Defaut, t, \prec, c, Cons) \rightarrow Rec$, qui, à partir de :

- l'ensemble des utilisateurs C,
- l'ensemble des entités recommandables I,
- un log/journal relatif aux entités de I : Log,
- une matrice d'utilité Mu $(C \times I)$,
- une fonction de prétraitement du log : $PretraitementL$,
- une fonction de prétraitement de la matrice d'utilité : $PretraitementM$,
- une fonction de génération de recommandations candidates sur l'ensemble I : $Generation$,

1. Notons que cette fonction d'utilité $u : C \times I \rightarrow \mathbb{R}$ est représentée sous la forme d'une matrice $C \times I$ et est nommée : Matrice d'utilité.

- une fonction d'ordonnancement des recommandations candidates : $Ordonnancement$,
- une fonction qui retourne une recommandation par défaut (au cas où le système ne trouve pas de recommandations candidates) : $Defaut$,
- un seuil $t \in \mathbb{R}$ (pour la fonction de génération),
- un ordre sur I : $\prec$ (pour la fonction d'ordonnancement),
- l'utilisateur/décideur : $c \in C$ pour qui le système va suggérer des recommandations,
- une contrainte de restriction $Cons$ (pour le prétraitement de la matrice d'utilité),

retourne un ensemble de recommandations Rec.

Cette fonction est associée à un algorithme qui requiert (en entrées) des données hétérogènes, doit prédire les scores correspondant aux recommandations candidates et doit les retourner à l'utilisateur par ordre de pertinence/intérêt. Ainsi, notre proposition repose sur le processus suivant, en trois étapes (voir partie *Etapes*, en bleu, de la figure 4.1) :

1. Prétraitement du log et/ou de la matrice d'utilité,
2. Génération des recommandations candidates en cherchant les entités recommandables qui coïncident le mieux avec les attentes de l'utilisateur (via la matrice d'utilité),
3. Ordonnancement des recommandations candidates afin de présenter les recommandations à l'utilisateur par ordre d'intérêt.

Cette approche est générique dans le sens où elle n'impose pas une méthode spécifique de prétraitement du log, de génération ou d'ordonnancement des recommandations candidates. Au lieu de cela, ces actions sont laissées comme paramètres de l'approche qui, de ce fait, peut être instanciée de différentes manières afin de changer la méthode de calcul des recommandations (en changeant ces paramètres, la manière dont les recommandations sont calculées change).

Notons que nous nous focalisons ici sur une approche hybride (voir la section 3.3.1.3). En effet, les approches hybrides ont la particularité de tirer avantage des approches collaboratives et à base de contenu, en limitant leurs inconvénients. De plus, la prise en compte d'un log/journal relatif aux entités de I permet d'accéder à une sorte d'historique du système, sorte de modèle de connaissances, introduisant ainsi des données/informations/connaissances supplémentaires.

Enfin, la figure 4.1 illustre notre proposition où un utilisateur (ou groupe d'utilisateurs) a un besoin (partie *Besoin* en vert), et notre système, à partir des entrées (partie *Entrées* en violet), procède en trois étapes (partie *Etapes* en bleu) afin de retourner des sorties (recommandations - partie *Sorties* en jaune) répondant au mieux au besoin de l'utilisateur. L'algorithme 1 détaille notre système générique de recommandation.

Prétraitement du log ou de la matrice d'utilité

Au vu du nombre d'utilisateurs et du nombre d'entités recommandables, le log, relatif aux interactions entre les utilisateurs et les entités recommandables, peut être très volumineux en plus d'être difficilement exploitable (de par son format non structuré). Le log a donc besoin d'être prétraité. A partir du log, la fonction de prétraitement $PretraitementL : Log \rightarrow PretraitementL(Log)$ permet d'obtenir un log prétraité. Par exemple, [Neg09] prétraite un log en utilisant la classification (*k-médoïdes*) (abordée en section 3.4.4). Notons que cette fonction de prétraitement peut être l'identité, auquel cas, le log prétraité est le log initial.

La matrice d'utilité, quant à elle, souffre parfois de faible densité car il est difficile d'obtenir les notes/avis des utilisateurs. Par ailleurs, dans le cadre de recommandations individuelles, suggérer

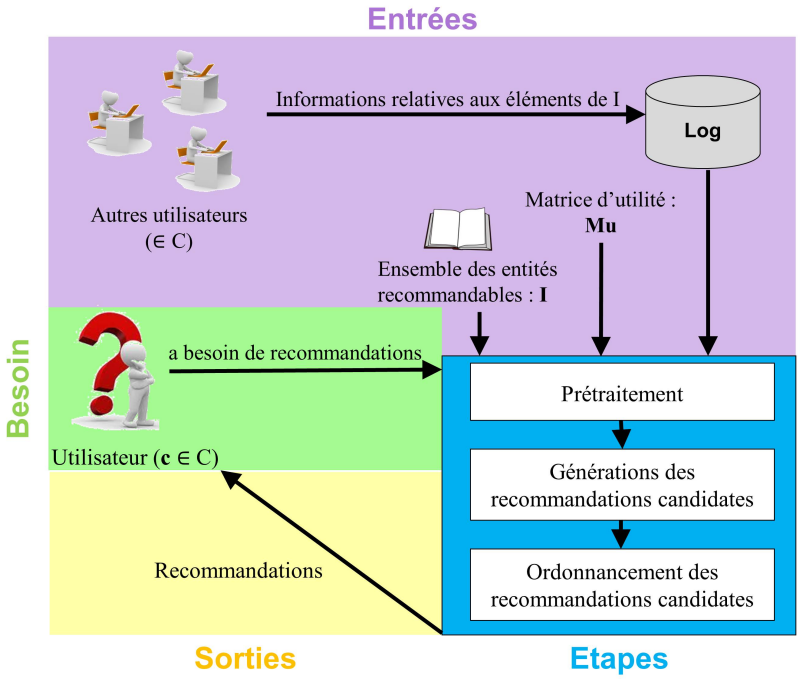

FIGURE 4.1 – Vue d'ensemble de notre système générique de recommandation

des recommandations à partir de l'intégralité du catalogue (et donc de la matrice d'utilité intégrale) ne pose pas de problème (outre celui de faible densité). Par contre, dans le cadre de recommandations à visée publique, utiliser l'intégralité du catalogue peut s'avérer dangereux. Par exemple, dans le cadre de la gestion de crises, la matrice d'utilité peut contenir des scores associés à des actions pouvant être mises en place pour limiter les effets d'un incendie comme des scores associés à des actions pour limiter les effets d'un tsunami ; or, si le système recommande des actions destinées à un incendie dans le cas d'un tsunami, cela peut coûter la vie à la population : restreindre la matrice aux incendies dans le cas de recommandations pour limiter les effets d'un incendie est donc important. Ainsi, la matrice d'utilité a besoin d'être prétraitée. A partir de la matrice d'utilité Mu et d'une contrainte de restriction $Cons$, la fonction de prétraitement $PretraitementM : Mu, Cons \rightarrow PretraitementL(Mu, Cons)$ permet d'obtenir une matrice d'utilité prétraitée. Le prétraitement de matrice est souvent réalisé via la réduction de dimensions (et notamment la décomposition en valeurs singulières [Bak05]) (abordée en section 3.4.3). Notons que, comme précédemment pour le log, cette fonction de prétraitement peut être l'identité, auquel cas, la matrice prétraitée est la matrice initiale.

Génération des recommandations candidates

A partir du log prétraité $\mathcal{L}$, de l'ensemble des entités recommandables I, d'une fonction de génération $Generation$ entre les entités du log prétraité et les entités recommandables (selon un seuil t), un ensemble de recommandations candidates est calculé grâce à la fonction $GenCand(\mathcal{L}, I, Generation, t)$. Cette fonction renvoie un ensemble de recommandations candidates ($SetCand$). Notons que dans certains cas, cet ensemble peut être vide. Les recommandations candidates peuvent être obtenues, par exemple, via des techniques de prédiction de scores ([WABN15c] utilise une approche bayé-

Algorithme 1 Système générique de recommandation : GenericReco

Entrées : C : l'ensemble des utilisateurs
I : l'ensemble des entités recommandables
Log : le log relatif à I
Mu : la fonction d'utilité de $i \in I$ pour $c \in C$
$PretraitementL$: une fonction de prétraitement des données du Log
$PretraitementM$: une fonction de prétraitement de la matrice d'utilité Mu
$Generation$: une fonction de concordance entre les données prétraitées du Log et l'ensemble I
$Ordonnancement$: une fonction d'ordonnancement sur l'ensemble I
$Defaut$: une fonction qui retourne une recommandation par défaut
t : un seuil ($\in \mathbb{R}$)
$\prec$: un ordre sur l'ensemble I
c : l'utilisateur courant : $c \in C$
$Cons$: une contrainte de restriction

Sorties : Rec : un ensemble ordonné de recommandations

$\mathcal{L} \leftarrow PretraitementL(Log)$
$\mathcal{MU} \leftarrow PretraitementM(Mu, Cons)$
$SetCand \leftarrow GenCand(\mathcal{L}, I, Generation, t)$
Si $SetCand \neq \emptyset$ **Alors**
  $Rec \leftarrow Ordonnancement(SetCand, \prec)$
Sinon
  $Rec \leftarrow Defaut(\mathcal{L})$
Fin Si
Retourne Rec

sienne) où sont sélectionnées les entités recommandables de I ayant un score supérieur au seuil t (abordées en section 3.3).

Ordonnancement des recommandations candidates

Dans l'étape précédente, un ensemble de recommandations candidates est calculé. L'objectif de cette étape est d'ordonner les recommandations, étant donné un critère de satisfaction, ou relation d'ordre, exprimé par l'utilisateur. Ainsi, à partir de l'ensemble de recommandations candidates $SetCand$ et d'une relation d'ordre $\prec$, la fonction d'ordonnancement $Ordonnancement$: $SetCand, \prec \rightarrow Ordonnancement(SetCand, \prec)$ permet d'obtenir l'ensemble ordonné des recommandations. Par exemple, un ordre sur les recommandations peut être relatif au score qu'elles ont obtenu (du score le plus élevé au score le plus bas), ou relatif à leur fréquence d'apparition dans le log.
Comme remarqué précédemment, l'ensemble de recommandations candidates peut être vide. Dans ce cas, il pourra être utile de proposer à l'utilisateur une recommandation par défaut (qui sera la même quel que soit l'utilisateur). Ainsi, à partir du log prétraité, la fonction $Defaut : \mathcal{L} \rightarrow Defaut(\mathcal{L})$ permet d'obtenir une (de) telle(s) recommandation(s). Par exemple, une recommandation par défaut peut être l'entité recommandable la plus fréquente du log ou la dernière entrée dans l'ensemble I (la plus récente).

Synthèse

Afin de pallier la « sur-spécialisation » des systèmes de recommandation, nos travaux ont abouti à la proposition d'un modèle générique de recommandation. Notre approche s'appuie sur l'ensemble des utilisateurs, l'ensemble des entités recommandables, une matrice d'utilité et un log/journal relatif aux entités recommandables. Elle est composée de trois étapes : prétraitement, génération et ordonnancement des recommandations candidates. Et finalement, elle retourne des recommanda-

tions répondant aux attentes/besoins de l'utilisateur. Notre approche est générique dans le sens où aucune technique particulière de prétraitement, génération ou ordonnancement n'est imposée. Cette proposition est en rupture par rapport aux travaux existants qui motivent des modèles spécifiques à chaque cas d'application, rendant certains systèmes complètement ad'hoc.

Dans la suite de ce chapitre, nous instancions notre système générique de recommandation sur différents cas d'applications que ce soit dans le cadre de recommandations individuelles ou à visée publique.

4.2 Applications

De notre point de vue, recommander des requêtes OLAP (*On-Line Analytical Process*) à un analyste qui interroge un entrepôt de données, recommander des supports d'apprentissage à un apprenant sur une plateforme d'apprentissage en ligne ou recommander des personnes expertes dans un environnement de travail collaboratif sont des recommandations individuelles, dans le sens où elles s'adressent à un utilisateur particulier qui les utilisera dans un cadre restreint : l'analyste doit mener à bien ses analyses, l'apprenant doit combler ses lacunes et une personne (ou un groupe de personnes) doit trouver une personne compétente sur un sujet donné.

Par contre, recommander des actions à mettre en place dans une ville ou en cas de crise sont des recommandations à visée publique, dans le sens où, si elles sont suivies, elles auront un impact sur les populations : les citoyens de la ville et les populations situées dans la zone de crise.

Il est à noter que, notre but étant de valider la pertinence de notre modèle générique, nous l'avons instancié dans des domaines d'applications ayant des objectifs et des enjeux différents.

4.2.1 Recommandations individuelles

Dans cette section, nous nous intéressons aux recommandations individuelles, c'est-à-dire qui s'adressent à un utilisateur (ou groupe d'utilisateurs) et qui seront utilisées dans un cadre restreint à l'utilisateur (ou au groupe). Plus précisément, dans le cadre de nos travaux, nous nous sommes intéressés aux domaines tels que OLAP, l'apprentissage numérique et les environnements de travail collaboratif. En particulier, en OLAP : recommander des requêtes pour faciliter l'exploration des données ; en apprentissage numérique : recommander des supports intéressants pour améliorer l'apprentissage des apprenants ; et en environnements de travail collaboratif : recommander des experts pour améliorer la collaboration, ont été/sont des défis captivants. Ces différents cas d'applications seront explicités dans les sections suivantes.

4.2.1.1 Recommander des requêtes OLAP

De nombreuses données sont collectées et partagées dans des bases de données qui sont explorées à des fins analytiques. Le problème de recommander une requête de base de données à un utilisateur a attiré l'attention depuis la fin des années 2000 [CEP09, KBG+09, SDP09]. Un exemple typique d'analyse de base de données est un entrepôt de données exploré par des décideurs utilisant des requêtes OLAP [Sar00]. Un entrepôt de données peut être considéré comme une grande base de données avec une structure de données particulière, partagée par de nombreux analystes qui ont divers intérêts et points de vue, explorés par des séquences de requêtes. Ainsi, dans les systèmes OLAP, le décideur navigue interactivement dans un cube de données (sous-ensemble de l'entrepôt de données) en lançant une séquence de requêtes, ce que nous appelons une session d'analyse (ou session). Dans un tel contexte, la recommandation de requêtes est particulièrement pertinente

puisque l'utilisateur a la tâche fastidieuse d'explorer un grand espace de données afin de trouver des informations précieuses/pertinentes afin de prendre rapidement des décisions judicieuses.

Les travaux présentés dans [Neg09, GMNS09, GMNS11, Neg11, MN11] avaient pour objectif de proposer des recommandations, sous forme de requêtes OLAP, à un utilisateur interrogeant un cube de données. Cette proposition tire parti de ce qu'ont fait les autres utilisateurs lors de leurs précédentes explorations du même cube de données. Nous avons proposé un système de recommandation qui, à partir d'un ensemble de séquences de requêtes, correspondant aux explorations du cube de données faites précédemment par différents utilisateurs, et de la séquence de requêtes de l'utilisateur courant, propose un ensemble de requêtes pouvant faire suite à la séquence de requêtes courante.

En instanciant la définition 4.1.1, nous avons défini une recommandation en OLAP (sous forme de requête) comme étant une requête $q \in Q$ (ensemble de toutes les requêtes possibles), telle que son utilité pour une session $s \in S$ (ensemble de toutes les sessions possibles) est maximale.

Définition 4.2.1 – *Recommandation de requêtes OLAP*
Soit Q l'ensemble de toutes les requêtes possibles et S l'ensemble de toutes les sessions d'analyse, étant donné un journal/log de sessions et une session courante lancée par un utilisateur qui a besoin d'aide, et soit u une fonction qui mesure l'utilité d'une requête q pour une session s, c'est-à-dire $u : Q \times S \rightarrow \mathbb{R}$. Alors pour chaque session $s \in S$, la requête recommandée $q' \in Q$ est celle qui maximise l'utilité pour la session : $\forall s \in S, q'_s = argmax_{q \in Q} u(s, q)$.

Exemple 4.2.1 – *Il est possible d'illustrer la matrice d'utilité $Q \times S$ où un score indique que la requête a été pertinente pour la session :*

u(s,q)	*Requête 1*	*Requête 2*	*Requête 3*	*Requête 4*	*Requête 5*
Session 1	*10*	*10*			
Session 2	*10*		*10*		
Session 3	*10*	*10*		*10*	*10*
Session 4	*10*	*10*		*10*	*10*

Notons qu'une cellule (s, q) de cette matrice d'utilité correspond au score d'utilité assigné à la requête q pour la session s. Notons que si la requête appartient à la session alors la requête est intéressante. Ici, la ligne de la Session 1 indique que les requêtes 1 et 2 ont été lancées lors de la session 1. De plus, si la requête n'a pas été lancée, il n'est pas possible de lui donner un score.

A l'image de notre approche générique (section 4.1 et figure 4.1), l'approche de recommandation de requêtes OLAP se décompose en trois étapes, comme illustré sur la figure 4.2 (les quatre parties colorées correspondent à celles de la figure 4.1) :

1. Prétraitement du log,
2. Génération des requêtes candidates en cherchant d'abord les sessions qui coïncident le mieux avec la session courante puis en prédisant ce que pourrait être la prochaine requête,
3. Ordonnancement des recommandations candidates afin de présenter les requêtes à l'utilisateur par ordre d'intérêt.

Dans le cadre des sessions d'analyse OLAP, la fonction *GenericReco* devient $GenericReco(\emptyset, \emptyset, Log, Mu, PretraitementL, Identité, Generation, Ordonnancement, Defaut, t, \prec, c, \emptyset)$ où l'ensemble des sessions est obtenu par le biais du log et l'ensemble des requêtes recommandables est obtenu via la fonction *Generation* (la liste exhaustive de toutes les sessions d'analyse ou de toutes les requêtes possiblement lançables sur un entrepôt de données serait beaucoup trop conséquente).

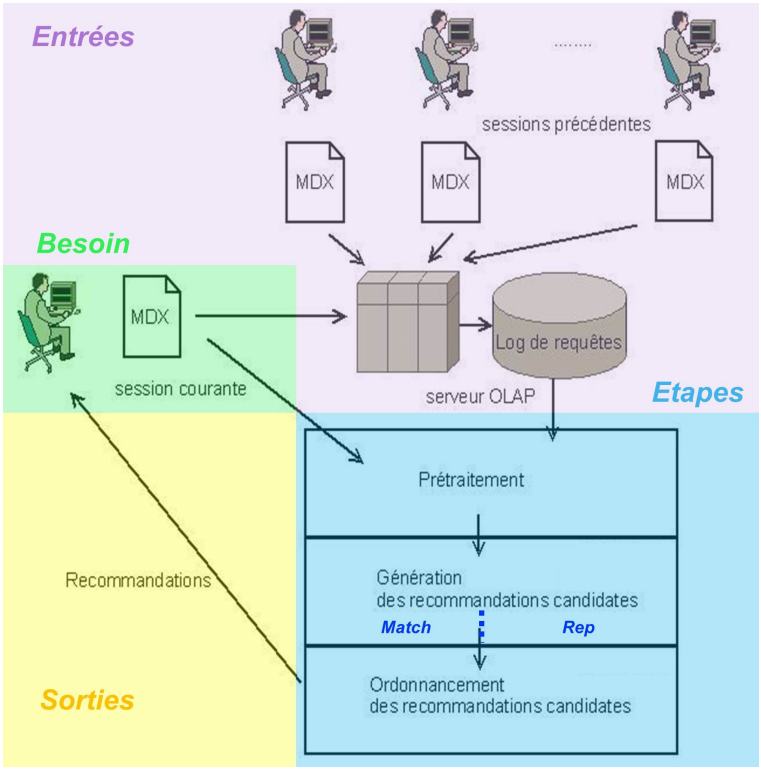

FIGURE 4.2 – Vue d'ensemble de notre approche de recommandation de requêtes OLAP

Combinaison	Prétraitement	Génération	Ordonnacement
ClusterH	K-médoïdes	Sous-séquences + médoïde du successeur	Proximité (Hausdorff + Hamming)
ClusterSP	K-médoïdes	Sous-séquences + médoïde du successeur	Proximité (Hausdorff + plus court chemin)
EdH	Identité	Distance de Levenshtein + dernier	Proximité (Hausdorff + Hamming)
EdSP	Identité	Distance de Levenshtein + dernier	Proximité (Hausdorff + plus court chemin)

TABLE 4.1 – Récapitulatif des combinaisons pour le système de recommandation de requêtes OLAP

Nous avons instancié cette approche de plusieurs manières en combinant différentes mesures de similarité, techniques de classification, ordonnancements, ... (*Approximate String Matching* [Nav01], *K-medoïds*, distance de Hausdorff [Hau14], ...). Le tableau 4.1 récapitule les instanciations [Neg09]. A partir des expérimentations menées, il apparait que EdH et $EdSP$ ont des performances similaires (avec un biais sur le calcul de la distance de Hausdorff favorisant EdH) et meilleures que celles de $ClusterH$ et $ClusterSP$.

Synthèse

Dans cette instanciation de notre modèle générique de recommandation, il apparait que :
- l'utilisateur a pour besoin d'explorer un cube de données efficacement (par le biais de requêtes OLAP),
- les données d'entrée à notre disposition sont l'ensemble des sessions d'analyse (S), l'ensemble

des requêtes recommandables (Q), un log contenant des requêtes précédemment lancées par d'autres utilisateurs et la matrice d'utilité $Q \times S$,
- les étapes sont :
  1. le prétraitement du log par partitionnement (via les K-médoïdes - très connue, facile à utiliser et plus adaptée aux requêtes OLAP que d'autres techniques de partitionnement (les opérations de type moyenne entre requêtes étant compliquées)) ou l'identité,
  2. la génération des requêtes candidates via les fonctions *Match* et *Rep*, la fonction *Match* calcule la similarité entre (les requêtes de) la session courante et (les requêtes de) les sessions du log, la fonction *Rep* retourne la requête q de la session s qui représente le mieux la session s (par exemple, la dernière requête de la session),
  3. l'ordonnancement des requêtes candidates en fonction de leur proximité avec la requête courante (via la distance de Hausdorff combinée soit à la distance de Hamming (facile à utiliser) soit au plus court chemin (qui prend en compte une spécificité d'OLAP : les hiérarchies)),
- par défaut, la requête qui, dans le log, a le plus grand nombre de requêtes différentes qui la précède, est retournée,
- le système retourne un ensemble ordonné de requêtes OLAP pertinentes permettant à l'utilisateur d'explorer efficacement le cube de données.

4.2.1.2 Recommander des supports intéressants dans les plateformes d'apprentissage numérique

Cours en ligne ouverts et massifs (*Massive open online course* - MOOC), jeux sérieux (*serious game*), l'apprentissage social,... , nul doute, l'éducation et l'apprentissage n'échappent pas à la révolution numérique qui submerge notre façon d'apprendre, de travailler et de penser. Aujourd'hui, tout le monde apprend sur Internet en suivant un MOOC ou en visionnant des vidéos postées par des experts du monde entier, en participant à des forums via des réseaux collaboratifs ou en pratiquant des jeux sérieux. Répondre aux attentes/besoins des apprenants tout en leur permettant d'apprendre mieux et de surmonter leurs lacunes est un défi majeur pour les plateformes d'apprentissage numérique.

Dans [BCCN17], nous nous concentrons sur des plateformes telles que les MOOCs (basées sur Open edX [edX15]). Les MOOCs sont utilisés par de nombreux apprenants et contiennent de nombreux cours dont les ressources d'apprentissage proviennent de différents enseignants. De plus, l'apprentissage s'étale dans le temps. Nous souhaitons donc offrir un système de recommandation qui supporte l'évolution/les changements au cours du temps (à base d'agents pour s'adapter aux changements - apprentissage et évolution) et adapté au nombre important d'apprenants (à base de filtrage collaboratif) afin de recommander des ressources pertinentes aux apprenants.

À cet effet, nous proposons un système multi-agent avec un ensemble d'agents autonomes agissant au nom de leurs croyances, préférences et objectifs. Les agents appartiennent à un système coopératif qui vise à être attentif au parcours d'apprentissage de chaque apprenant et à lui assurer un bon soutien en fournissant des ressources appropriées dans chaque situation unique. Les agents sont également capables d'apprendre en utilisant leurs expériences passées et donc d'améliorer leur processus décisionnel en ajustant leurs stratégies. Ceci est particulièrement approprié dans le cadre des systèmes de recommandation qui seront en mesure d'améliorer les recommandations au fil du temps. De multiples travaux existent afin d'aider les apprenants dans le choix de la bonne formation/cours

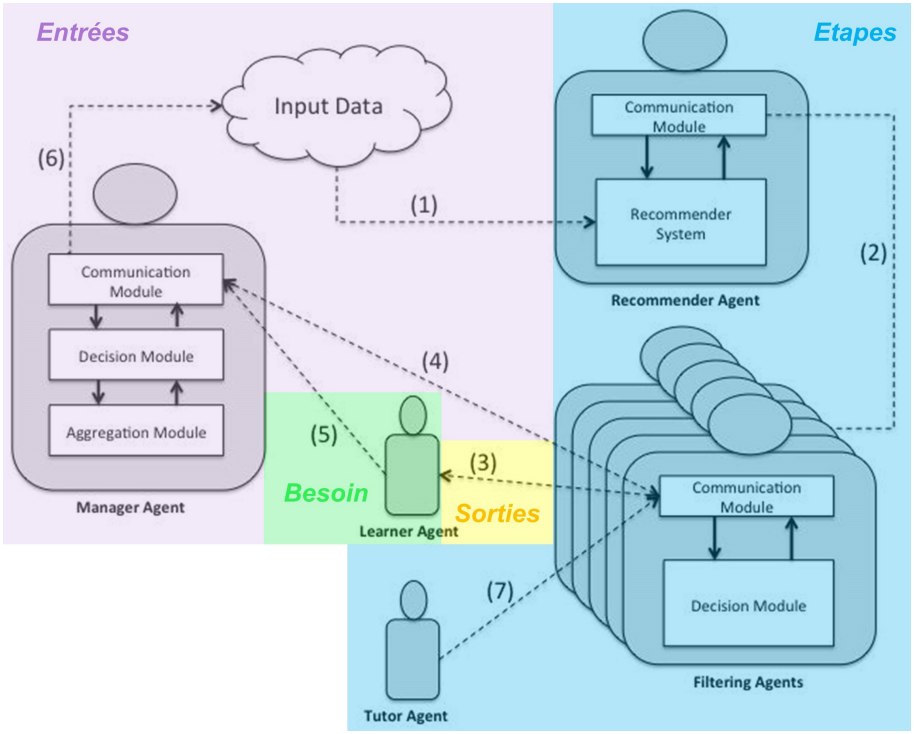

FIGURE 4.3 – Vue d'ensemble de notre approche à base d'agents pour les plateformes d'apprentissage numérique

au moyen de systèmes de recommandation [FB06, Hsu08, EAJ+10, ACQL14, OS15, BC15, SBRJ16]. Cependant, à notre connaissance, il n'existe pas de système de recommandation à base d'agents avec une approche par filtrage collaboratif. Ainsi, nous introduisons un tel système qui recommande des ressources d'apprentissage pertinentes aux apprenants afin de les aider à surmonter leurs lacunes.

Le système de recommandation à base d'agents que nous proposons est supporté par plusieurs types d'agents représentés sur la figure 4.3 (les quatre parties colorées correspondent à celles de la figure 4.1). Le processus de recommandation est lancé par l'agent recommandant (*Recommender Agent*) qui a accès à plusieurs logs stockés dans des documents *JSON* [2] et contenant des informations de référence sur les paquets de données d'événements. Les événements sont émis par le serveur, le navigateur ou l'appareil mobile pour capturer des informations sur les interactions avec le didacticiel et le tableau de bord de l'instructeur [edX15]. Ces données ne sont pas toutes pertinentes. L'agent recommandant est en mesure de juger de la pertinence des événements.

Chaque agent possède un module de communication mis en œuvre pour permettre l'échange de messages entre les agents en fonction du protocole de communication. En outre, il permet l'interprétation et la construction des messages. Le protocole de communication spécifie les actions que les agents sont autorisés à faire lors du processus de recommandation. Le tableau 4.2 présente une vue partielle des primitives utilisées par les agents pour communiquer.

2. JSON (JavaScript Object Notation - Notation Objet issue de JavaScript) est un format léger d'échange de données.

Primitive	Commentaire	Contexte
$updateData(ma, ra, < data >)$	L'agent gestionnaire ma envoie une demande à l'agent recommandant ra pour mettre à jour les données	L'agent gestionnaire ma lance le processus en envoyant les données d'entrée à l'agent recommandant ra
$updateRanking(ta, fa, < resource, ranking >)$	L'agent tuteur ta commande l'agent de filtrage fa pour mettre à jour un classement des ressources	Selon ses préférences locales, ta prend la décision de mettre à jour un classement des ressources
$information(fa, ma, < info >)$	L'agent de filtrage fa retourne des informations sur la recommandation à l'agent gestionnaire ma	Après avoir choisi une ressource donnée, fa communique cette décision pour information à l'agent gestionnaire ma
$feedback(la, ma, < resource, score >)$	L'agent apprenant la envoie un commentaire (*feedback*) à l'agent gestionnaire ma concernant une ressource	Après avoir évalué l'utilité d'une ressource, la retourne un score à l'agent gestionnaire ma

TABLE 4.2 – Protocole de communication des agents

La figure 4.4 montre les interactions entre les agents afin d'aider les apprenants dans leur expérience d'apprentissage en suggérant des ressources utiles. Sur la base des données d'entrée, l'agent recommandant (RecomAg) fournit un classement des recommandations potentiellement intéressantes (ressources) à l'agent de filtrage concerné (FilteringAg). Les agents de filtrage choisissent le plus approprié, l'envoient à l'agent apprenant (LearnerAg) et communiquent cette décision pour information à l'agent gestionnaire (ManagerAg). Chaque agent de filtrage est associé à un cours et peut être personnalisé par l'agent gestionnaire concerné qui a la possibilité de mettre à jour le classement en fonction de ses préférences. À la fin du processus de recommandation, l'agent apprenant envoie ses commentaires pour évaluer la ressource recommandée à l'agent gestionnaire qui centralise tous les commentaires et met à jour ses règles de décision (si nécessaire). Les nouveaux faits et les règles mises à jour sont ajoutés aux données d'entrée afin de les actualiser.

Agent recommandant (*Recommender Agent*)
En instanciant la définition 4.1.1, le problème de la recommandation de ressources dans le cadre d'une plateforme d'apprentissage numérique peut être défini de la manière suivante :

Définition 4.2.2 – *Recommandation de ressources*
Soit A l'ensemble de tous les apprenants (utilisateurs) et I l'ensemble de toutes les ressources d'apprentissage possibles (qui peuvent être recommandées), et soit u une fonction qui mesure l'utilité d'une ressource i pour un apprenant a, c'est-à-dire $u : A \times I \rightarrow \mathbb{R}$. Alors pour chaque apprenant $a \in A$, la ressource recommandée $i' \in I$ est celle qui maximise l'utilité pour l'apprenant : $\forall a \in A, i'_a = argmax_{i \in I} u(a, i)$.

Dans les systèmes de recommandation, l'utilité d'un élément (ici une ressource d'apprentissage) est généralement représentée par un score qui indique l'appréciation de cet élément par un utilisateur particulier (ici un apprenant). La matrice d'utilité est donc $Mu : Apprenants\ (A) \times Ressources\ (I)$. Dans le contexte multi-utilisateurs de l'apprentissage numérique, l'approche la plus appropriée est le filtrage collaboratif. Les systèmes basés sur le filtrage collaboratif produisent des recommandations en calculant la similarité entre les préférences de différents utilisateurs.

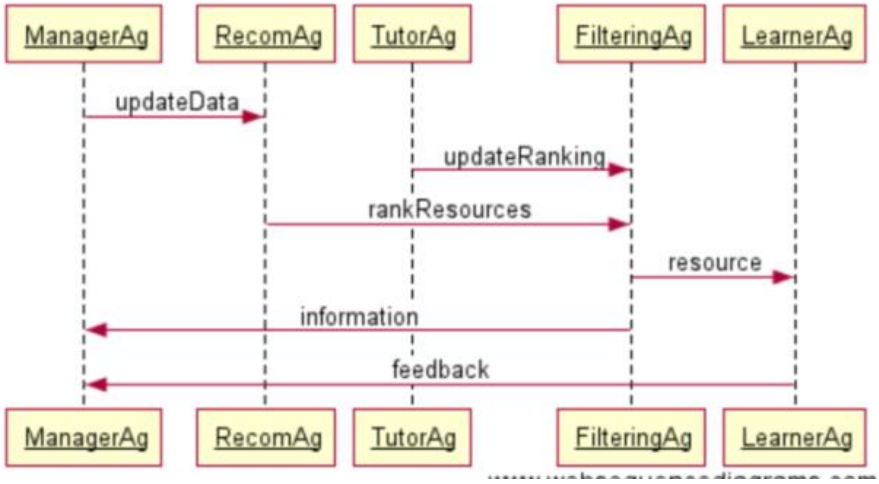

FIGURE 4.4 – Processus de recommandation (UML)

Dans le cadre des plateformes d'apprentissage numérique, la fonction *GenericReco* devient $GenericReco(A, I, \emptyset, Mu, Identité, PretraitementM, Generation, Ordonnancement, Defaut, t, \prec, a_c, \emptyset)$. Cette fonction retourne l'ensemble ordonné des ressources d'apprentissage qui peut être recommandé à l'apprenant courant l_c. Les trois étapes sont :

1. Prétraitement (normalisation et/ou réduction) de la matrice d'utilité Mu,
2. Génération des recommandations candidates (prédictions des scores),
3. Ordonnancement des recommandations.

En raison de son évolutivité, nous proposons d'utiliser une approche « *item-to-item* » en utilisant la décomposition en valeur singulières (SVD) (connue pour être facile à utiliser) pour la prédiction de scores, et les recommandations par défaut sont la ressource d'apprentissage la plus populaire et la plus récente. Ceci est particulièrement adapté à notre contexte (apprentissage numérique) et aux données correspondantes.

Ainsi notre instanciation devient $GenericReco(I, A, Mu, calcAverage, calcSVD, decroissance, popNew, t, \prec_{ressource}, l_c)$.

$CalcAverage(Mu)$ calcule les scores moyens pour chaque ressource (*item*) dans Mu et renvoie une nouvelle matrice $\mathcal{MU}$ en remplissant la matrice Mu (avec les valeurs moyennes obtenues). Notons que Mu peut être très creuse, donc, pour saisir une relation latente significative, nous commençons par enlever la faible densité en utilisant les scores moyens pour une ressource, selon [SKKR00].

$calcSVD(I, A, l_c, \mathcal{MU}, t)$ calcule la décomposition en valeurs singulières de $\mathcal{MU}$ (obtenue à l'étape de prétraitement) et renvoie l'ensemble des ressources non-ordonné $SetCandRessource$ (sous-ensemble de I) qui peuvent être recommandées à l'apprenant courant l_c et la valeur de prédiction correspondante. Ces ressources candidates à la recommandation correspondent aux meilleures prévisions obtenues par les valeurs singulières de la SVD.

$decroissance(SetCandRessource, \prec_{ressource})$ ordonne les ressources candidates de $SetCandRessource$ selon l'ordre décroissant ($\prec_{ressource}$) des valeurs de prédiction.

$popNew(I)$ retourne l'ensemble $\{s_{pop}, s_{new}\}$ où s_{pop} est la ressource la plus populaire, c'est-à-dire celle ayant, en moyenne, le meilleure score parmi tous les apprenants (ensemble A) dans Mu, et s_{new} est la ressource la plus récente dans l'ensemble des ressources I (recommandations par défaut).

	Ressource (I)	AV_1	AV_2	AE_1	AE_2
Apprenant (A)		[Video] Python	[Video] Java	[Ex] Python	[Ex] Java
ST_1		8	7	2	
ST_2		2		8	
ST_3		2			
ST_4			6	8	8

TABLE 4.3 – Evaluations initiales utilisées pour entrainer le système de recommandation.

Finalement, l'agent recommandant renvoie un ensemble ordonné de ressources d'apprentissage qui peut aider un apprenant donné à surmonter ses lacunes.

Agent de filtrage (*Filtering Agent*)
Chaque agent de filtrage est associé à un cours unique et peut être géré par un agent tuteur. En plus du module de communication, un module de décision est mis en œuvre en fonction de la base de connaissances locale mise à jour en temps réel pour le compte de l'apprenant via l'agent gestionnaire. Les règles de décision sont établies par l'agent tuteur afin d'associer les priorités aux ressources qui peuvent être recommandées aux apprenants. Nous considérons comme dans [BC17] que le système traite de plusieurs types de ressources (par exemple ressources d'apprentissage), de supports et de communications.

Agent gestionnaire (*Manager Agent*)
Sur la base de son module de communication, l'agent gestionnaire est lié à tous les agents développés dans la plateforme d'apprentissage. Il a un rôle clé dans le processus de recommandation. Il centralise les informations cruciales à deux niveaux principaux : (i) les informations sur les décisions prises par les agents de filtrage, (ii) les informations sur les commentaires des agents apprenants pour évaluer la pertinence et la qualité des ressources recommandées. Ces informations sont analysées par le module d'agrégation capable de faire une évaluation multi-critère de chaque ressource recommandée et de la transmettre au module de décision pour en déduire les règles et finalement décider si, par conséquent, il devrait y avoir une mise à jour des données d'entrée.

Exemple 4.2.2 – *Nous illustrons notre modèle sur un exemple. Nous décrivons d'abord les données, puis le processus de recommandation de base, les comportements des agents de filtrage et le comportement de l'agent gestionnaire. Nous considérons un exemple de MOOC avec deux cours de Java et deux cours de Python (de même niveau, $C_1, ..., C_4$). Chaque cours est suivi par 100 apprenants ($ST_i, i \in [1, 100]$) et est géré par un tuteur, représenté par son agent $T_1, ...T_4$. Les ressources proposées aux apprenants sont deux vidéos AV_1, AV_2 et deux exercices AE_1, AE_2, dans les deux cas, un pour Python et un pour Java. Une fois fini, les apprenants donnent des commentaires sur leurs activités (score entre 1 et 10). Les évaluations passées sont résumées dans la matrice d'utilité (voir le tableau 4.3) qui est fournie au système de recommandation.*

Le processus peut être décrit comme suit :

- *Le système de recommandation est entrainé sur la matrice Mu. Une fois entrainé, le système peut noter les activités pour tout profil d'apprenant qu'il soit nouveau ou ancien,*
- *Les agents de filtrage gèrent les résultats obtenus par le système de recommandation pour remplir leurs préférences d'enseignement et présentent les propositions finales à l'apprenant,*

– *L'agent gestionnaire reçoit les commentaires des apprenants et gère le mécanisme de mise à jour du système de recommandation.*

Recommandation initiale en utilisant le filtrage collaboratif

Le système de recommandation est entrainé à partir des commentaires passés des apprenants. La réduction de dimensions par décomposition en valeurs singulières (SVD) est effectuée sur la matrice Mu pour créer des profils d'apprenants et de ressources dans l'espace latent de la SVD. Les ressources et les apprenants peuvent être projetés dans l'espace de dimensions k généré par la SVD. Nous considérons ici une simple SVD à deux dimensions (dimensions D_1 et D_2). Les valeurs de l'apprenant sont données dans le tableau 4.4, les valeurs des ressources dans le tableau 4.5 et les valeurs singulières dans le tableau 4.6.

	D_1	D_2
ST_1	0,483	-0,863
ST_2	0,512	0,412
ST_3	0,474	0,21
ST_4	0,526	0,201

TABLE 4.4 – Valeurs de l'espace latent à partir de la matrice d'utilité, pour les apprenants.

	D_1	D_2
AV_1	0,316	-0,731
AV_2	0,514	-0,119
AE_1	0,482	-0,669
AE_2	0,635	-0,047

TABLE 4.5 – Valeurs de l'espace latent à partir de la matrice d'utilité, pour les ressources.

	D_1	D_2
Sigma	25,18	6,64

TABLE 4.6 – Premières valeurs singulières à partir de la matrice d'utilité.

Le score peut être calculé pour n'importe quel apprenant l_c, même s'il ne se trouvait pas dans la matrice originale (il suffit de projeter le nouvel apprenant dans l'espace latent en utilisant la matrice de projection existante). Par exemple, si on considère l'apprenant ST_3 (ou un profil comme ST_3), le score sera le produit scalaire des coordonnées de l'apprenant et des coordonnées de la ressource (pondérée par les valeurs singulières) comme illustré par le tableau 4.7. Ce sont les scores fournis à l'agent de filtrage.

Agents de filtrage et tuteurs

Un agent de filtrage représente un cours géré par un agent tuteur. Nous considérons ici l'agent T_3 (un cours de Java). Il appliquera principalement trois types de filtres :

– *les filtres locaux, en fonction du cours, de l'état de l'apprenant et éventuellement du tuteur,*

	AV_1	AV_2	AE_1	AE_2
Score		5,986	6,694	7,518
Rank		3	2	1

TABLE 4.7 – Résultats initiaux et classement du système de recommandation pour l'apprenant ST_3.

s'adaptent au cours et gèrent la progression du cours. Ici, nous allons nous restreindre à des ressources relatives à Java.

- *les filtres de préférence : le tuteur (humain) peut personnaliser les filtres en fonction de ses préférences et de son expertise en matière d'enseignement. Nous allons ici considérer une simple règle de préférences vidéo.*
- *les filtres globaux (principalement fournis par l'agent gestionnaire), par exemple pour sélectionner les activités que les apprenants peuvent sélectionner. Pour l'instant, ils sont vides.*

L'application de ces filtres sur notre exemple permet de proposer une nouvelle activité à ST_3 dans le cours de Java, comme illustré par le tableau 4.8.

	AV_1	AV_2	AE_1	AE_2
Score	-	8,986	-	7,518
Rank		1		2

TABLE 4.8 – Résultats finaux et classements de l'agent de filtrage pour l'apprenant ST_3.

AV_1 a déjà été utilisé par ST_3, AE_1 a été supprimé par le filtre global, et la forte valeur de AV_2 est biaisée en raison des préférences du tuteur. L'apprenant ST_3 se verra recommander en premier AV_2, puis AE_2. Ainsi, le résultat présenté à l'apprenant suit ses préférences déduites (par le système de recommandation), mais suit également les contraintes du cours et les préférences du tuteur. À la fin de l'activité sélectionnée (par exemple AV_2), l'apprenant donne son avis (ici, 1 car il n'aime pas les vidéos) qui est envoyé par l'agent apprenant à l'agent gestionnaire pour mettre à jour le système de recommandation.

Agent gestionnaire et Gestion du système

Tout d'abord, l'agent gestionnaire actualise périodiquement le système de recommandation (chaque nuit/semaine selon les capacités de calcul, la quantité de commentaires et les biais introduits dans le système de recommandation). Par exemple, les nouveaux commentaires de l'apprenant ST_3 sont inclus dans la matrice d'utilité qui sera utilisée pour le prochain entrainement du système de recommandation (elle changera à la fois la définition de l'espace de la SVD et la matrice d'utilité utilisée pour calculer les préférences des apprenants les plus proches).

Deuxièmement, l'agent gestionnaire peut appliquer des règles globales qui seront envoyées aux agents de filtrage. Nous pouvons également appliquer des règles pour un type spécifique d'apprenants (les apprenants ont accès à certaines ressources en fonction de ce qu'ils ont payé).

Troisièmement, l'agent gestionnaire gère et contrôle les agents de filtrage. Les agents de filtrage sont principalement libres de définir leurs préférences relatives aux règles de filtrage, mais certaines doivent être validées (les plus restrictives), et toutes sont envoyées aux agents gestionnaires. L'agent gestionnaire affiche un aperçu de toutes les politiques et résultats du tuteur. Des règles peuvent être définies pour alerter en cas de problèmes. Par exemple, si la proportion de mauvaises évaluations pour une ressource spécifique est trop élevée. Dans notre exemple, si trop d'apprenants mettent 2 ou

moins à la vidéo AV_2 simplement parce que le tuteur préfère les vidéos, une alerte sera augmentée. L'agent gestionnaire peut réagir en modifiant le biais maximal que ce tuteur peut appliquer. Le tuteur T_3 ne sera pas en mesure d'imposer des vidéos aux apprenants qui ne les aiment pas, même s'il croit que c'est préférable pour eux.

Enfin, l'agent gestionnaire gère les nouveaux arrivants et les nouvelles activités (cas du démarrage à froid). Pour une plateforme comme la nôtre, même les nouveaux arrivants ont des données (évaluations initiales). Mais dans un cas plus général, ils pourraient être complètement inconnus. Dans ce cas, l'agent gestionnaire peut laisser le système de recommandation suggérer les activités les plus généralement aimées (par défaut) et/ou définir certains filtres sur les nouveaux arrivants envoyés aux agents de filtrage pour imposer certaines activités initiales (activités parrainées ou activités d'introduction spécialement développées pour aider les nouveaux arrivants), et/ou laisser l'agent tuteur définir les scores de départ. Le résultat de ces activités initiales aidera le système de recommandation à apprendre les préférences de l'apprenant. De même, les scores des nouvelles activités peuvent être définis avec des règles de filtrage jusqu'à ce que certains commentaires leur permettent d'être intégrés dans le système de recommandation.

Synthèse

Nous avons présenté un système de recommandation à base d'agents qui vise à aider les apprenants en difficulté dans leur processus d'apprentissage en suggérant des ressources d'apprentissage pertinentes en fonction de leurs lacunes détectées ou de leurs défauts. Ainsi, nous avons proposé un système coopératif multi-agents composé d'un ensemble d'agents cognitifs autonomes afin de recommander des ressources utiles aux apprenants. Notre algorithme de recommandation à base d'agents utilise une approche « *item-to-item* » en utilisant la décomposition en valeurs singulières (SVD) pour la prédiction des scores.

Dans cette instanciation de notre modèle générique de recommandation, il apparait que :

- l'apprenant a pour besoin de combler ses lacunes,
- les données d'entrée à notre disposition sont l'ensemble des apprenants (A), l'ensemble des ressources d'apprentissage (I) et la matrice d'utilité $A \times I$,
- les étapes sont :
  1. le prétraitement de la matrice d'utilité (via $calcAverage$),
  2. la génération des ressources candidates (via $calcSVD$),
  3. l'ordonnancement des ressources candidates en fonction de la valeur décroissante de la SVD
- par défaut, la ressource la plus populaire et la ressource la plus récente sont retournées,
- le système retourne un ensemble ordonné de ressources d'apprentissage qui permettront à l'apprenant de combler ses lacunes.

4.2.1.3 Recommander des experts dans les environnements de travail collaboratif

La collaboration peut être définie comme un processus partagé dans lequel deux personnes, ou plus, avec des compétences complémentaires interagissent pour créer une compréhension commune [Sch90]. L'utilisation d'un environnement informatique peut faciliter la collaboration. Utiliser un tel environnement permet également d'exploiter les traces d'interactions. Dans ce cadre, une trace est définie par [ZCEZM11] comme un enregistrement des actions effectuées par un utilisateur sur un système.

Dans [WABN14, WABN15a, WABN15b, WABN15c, WABN16b, Wan16], nous nous sommes intéressés à l'exploitation des traces d'interaction à des fins de recommandation.

Le système de recommandation visé tient compte d'un modèle sémantique [3] (Qui travaille avec qui sur quoi ? Le sujet A est plus proche du sujet B, ...) et des traces d'interactions enregistrées (Qui a partagé un document sur tel sujet avec qui ? Qui échange souvent avec tel expert ? ...).

Ainsi, dans le cadre de nos travaux sur la recommandation dans les environnements de travail collaboratif, nous nous sommes focalisés sur la recommandation d'utilisateurs compétents (experts) à partir de l'analyse sémantique de leurs traces d'(inter)actions au sein de l'environnement de travail collaboratif.

Nous sommes intéressés par les effets des actions ainsi que par les actions elles-mêmes. Un ensemble d'actions, étape par étape, est défini comme une trace [ZCEZM11]. Grâce à la modélisation et l'analyse, les traces, en retour, aident à indiquer la compétence d'un individu [NTdSX13]. Nous proposons donc une approche qui modélise, enregistre et analyse les traces des utilisateurs pour recommander les utilisateurs ayant le plus d'expertise sur un sujet déterminé. Pour atteindre cet objectif, les tâches suivantes sont nécessaires : (i) proposer une structure sémantique pour enregistrer les traces, (ii) proposer un modèle de compétence, (iii) évaluer les traces et (iv) proposer des recommandations.

En instanciant la définition 4.1.1, nous avons défini une telle recommandation comme étant un utilisateur $c \in C$ (ensemble de tous les utilisateurs enregistrés dans l'environnement de travail collaboratif), tel que son utilité pour un sujet $s \in S$ (ensemble de tous les sujets abordés au sein de l'environnement de travail collaboratif) est maximale.

Définition 4.2.3 – *Recommandation d'experts dans les environnements de travail collaboratif*
Soit C l'ensemble de tous les utilisateurs enregistrés dans l'environnement de travail collaboratif et S l'ensemble de tous les sujets abordés au sein de ce même environnement, étant donné un journal/log de traces d'interactions et soit u une fonction qui mesure l'utilité d'un utilisateur c pour un sujet s, c'est-à-dire $u : C \times S \rightarrow \mathbb{R}$. Alors pour chaque sujet $s \in S$, l'utilisateur/expert recommandé $c' \in C$ est celui qui maximise l'utilité pour le sujet : $\forall s \in S, c'_s = argmax_{c \in C} u(c, s)$.

Exemple 4.2.3 – *La matrice d'utilité $C \times S$ peut être illustrée comme suit :*

u(c,s)	*Sujet 1*	*Sujet 2*	*Sujet 3*	*Sujet 4*	*Sujet 5*
Utilisateur/Expert 1	*0.5*	*0.4*			
Utilisateur/Expert 2	*0.9*		*0.3*		
Utilisateur/Expert 3	*0.2*	*0.8*			*0.5*
Utilisateur/Expert 4	*0.5*			*0.1*	*0.9*

Notons qu'une cellule (c, s) de cette matrice d'utilité correspond au score d'utilité assigné à l'utilisateur/expert c pour le sujet s. Ici, les scores sont des valeurs entières entre 0 et 1 (0 étant la plus mauvaise et 1 la meilleure) indiquant l'expertise de l'utilisateur/expert c quant au sujet s.

Dans le cadre des environnements de travail collaboratif, la fonction *GenericReco* devient $GenericReco(C, S, Log, Mu, PretraitementL, Identité, Generation, Ordonnancement, \emptyset, \emptyset, \prec, c, \emptyset)$. Cette fonction retourne l'ensemble ordonné des utilisateurs qui peut être recommandé en tant qu'ensemble d'experts dans un environnement de travail collaboratif.

3. Un modèle sémantique représente les objets/personnes et les concepts manipulés au travers des activités réalisées (indépendamment des moyens mis en œuvre).

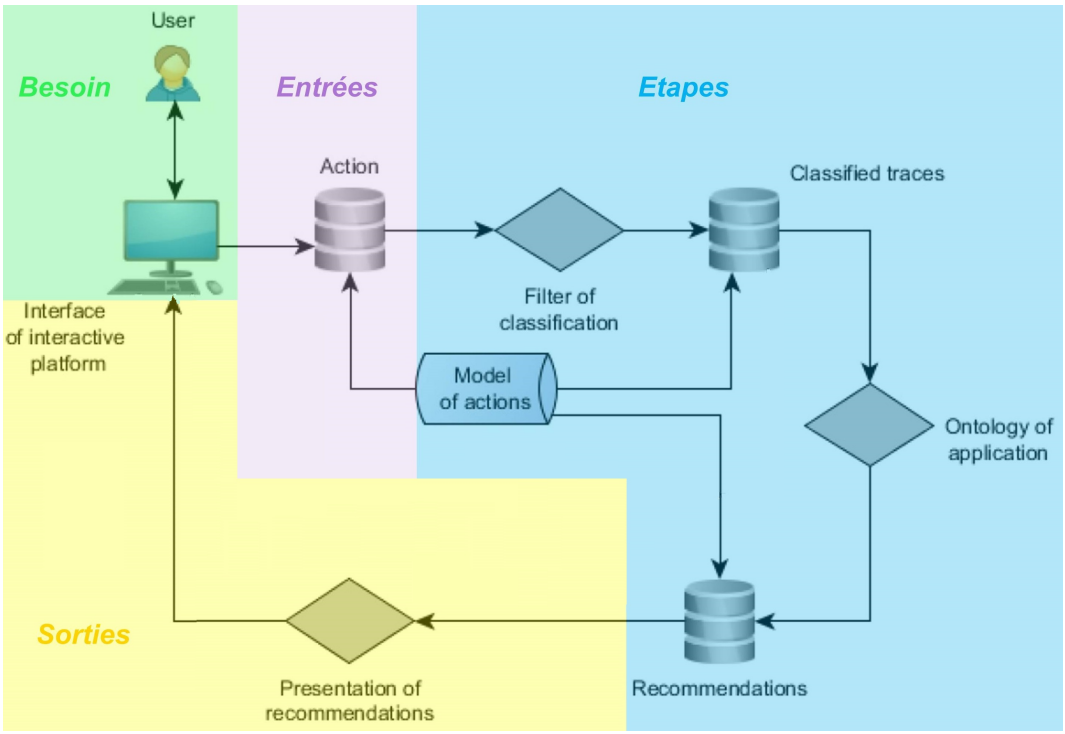

FIGURE 4.5 – Structure de notre système de recommandation dans les environnements de travail collaboratif basé sur l'exploitation sémantique des traces.

Les trois étapes sont :

1. Prétraitement des traces d'interactions (*Log*) en passant par une ontologie d'application (pour avoir le niveau le plus de spécificité),
2. Génération des recommandations candidates soit via le TF-IDF, soit via la classification naïve Bayésienne,
3. Ordonnancement des recommandations par ordre décroissant des valeurs prédites.

Ainsi, nous orchestrons un modèle d'actions (reposant sur le modèle de graphe RDF (*Resource Description Framework*) pour modéliser les actions [AGHH12]), un modèle de compétence (de type « action-connaissance » qui intègre les forces des modèles de compétences existants tout en limitant leurs faiblesses pour permettre d'évaluer les compétences de l'utilisateur) et une ontologie d'application pour formuler des recommandations sur la compétence des utilisateurs. La figure 4.5 montre la structure de notre approche (les quatre parties colorées correspondent à celles de la figure 4.1). Tout d'abord, les actions des utilisateurs sont collectées et modélisées à partir d'une plateforme interactive. Après avoir été filtrées par le filtre de classification, nous obtenons des traces classifiées. Alternativement, nous appliquons un algorithme pour calculer un indice indiquant la corrélation entre les traces classifiées d'un utilisateur donné et d'un sujet donné. Ces valeurs peuvent conduire à des informations utiles qui sont présentées sous forme de recommandations personnalisées, soit à un groupe défini, c'est-à-dire un ensemble d'utilisateurs de la plateforme, soit à un utilisateur individuel.

Notre modèle d'actions présente deux avantages principaux par rapport à une forme traditionnelle de journal/log d'utilisateurs :

- les actions sont présentées dans un graphe multiple, étiqueté et orienté. Cela permet une meilleure structure de stockage et d'utilisation des actions,
- les différents types d'actions ont une importance différente.

Mesure de la compétence

Pour exploiter les fonctionnalités liées aux traces des utilisateurs (à partir de notre modèle d'actions et notre modèle de compétence), nous avons besoin de méthodes mathématiques pour quantifier les compétences avec l'aide de ces fonctionnalités.

1. TF-IDF

   Afin d'évaluer l'importance des différentes traces, dans [WABN14], nous avons adapté le TF-IDF (*Term Frequency - Inverse Document Frequency*).

   La trace d'un utilisateur est composée de toutes les activités qu'il/elle a réalisé. Les traces de tous les utilisateurs forment un ensemble de traces. A partir de ce constat, nous pouvons faire une analogie entre « terme, document, corpus » et « activité, trace, ensemble de traces ». Si un mot apparaît plus souvent dans un document et en même temps moins souvent dans les autres documents du même corpus, il représente mieux ce document. Dans notre cas, nous voulons évaluer la corrélation entre une trace d'un utilisateur donné et un sujet donné. Nous proposons de considérer que si les activités d'un utilisateur sont plus pertinentes concernant un sujet, l'utilisateur a plus de connaissances à ce sujet. Nous sommes donc en mesure de recommander cet utilisateur en tant qu'expert dans ce domaine. Nous étudions donc la relation entre les activités, les traces et l'ensemble des traces dans un groupe d'utilisateurs travaillant dans le même environnement de travail collaboratif. Nous adaptons donc l'équation de TF-IDF :

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{4.2.1}$$

$$idf_i = log \frac{|U|}{|\{u_l : n_{i,l} > 0\}|} \tag{4.2.2}$$

   L'index de TF-IDF, $C_{i,j}$ indiquant la compétence de l'utilisateur j sur le sujet i, est définie comme suit :

$$C_{i,j} = tf_{i,j} \times idf_i \tag{4.2.3}$$

   où :

   - $n_{i,j}$ est le nombre d'activités relatives au sujet i réalisées par l'utilisateur j,
   - $\sum_k n_{i,j}$ est le nombre d'activités relatives aux k sujets réalisées par l'utilisateur j,
   - $|U|$ est le nombre d'utilisateurs dans un groupe,
   - $|\{u_l : n_{i,l}\}|$ est le nombre d'utilisateurs dans un groupe qui ont réalisés au moins une activité sur le sujet i.

   Finalement, l'utilisateur j ayant le plus fort indice de compétence $C_{i,j}$ sera recommandé comme expert pour le sujet i.

2. Classification Bayésienne

   Dans [WABN15c, WABN15b], comme une trace est composée d'activités sur un ensemble

Sujet	Fréquence	Fréquence dans le groupe
Java	10	7
Python	16	5
C++	2	8
PHP	7	3

TABLE 4.9 – Exemple de calcul du TF-IDF pour « activité, trace, ensemble de traces » : Les traces de John dans un groupe de 10 utilisateurs

de concepts, nous avons besoin d'une méthode qui gère mieux les facteurs multidimensionnels (que l'approche TF-IDF). La classification naïve Bayésienne (*Naïve Bayes Classifier*) est basée sur le théorème de Bayes avec une hypothèse d'indépendance forte (naïve) et convient aux cas ayant des dimensions d'entrée élevées [GPB10]. En classification statistique, le classifieur naïf de Bayes minimise la probabilité de mauvaise classification. Nous avons adapté cette méthode à nos objectifs :

$$p(Comp_j\,|Tra_i\,) = \frac{p(Comp_j)p(Tra_i\,|Comp_j\,)}{p(Tra_i)} \tag{4.2.4}$$

où $p(Comp_j)$ est défini comme la probabilité *a priori* qu'un utilisateur aléatoire ait la compétence la plus élevée sur le concept $j \in J$. $p(Comp_j\,|Tra_i\,)$ représente la probabilité qu'un utilisateur $i \in I$ ayant pour traces Tra_i dans la plateforme possède la plus haute compétence sur le concept j. $p(Tra_i)$ est la probabilité de réaliser Tra_i. Comme décrit précédemment, une trace est une combinaison d'activités sur une variété de concepts. Nous définissons $p(Tra_i)$ comme :

$$p(Tra_i) = p(A_{i,1})p(A_{i,2}), ..., p(A_{i,n}) = \prod_k p(A_{i,k}) \tag{4.2.5}$$

où $p(A_{i,k})$ représente la probabilité que les activités de trace i sur le concept k aient lieu. Respectivement, Tra_i est composé des activités sur les n concepts. Ainsi, l'équation 4.2.4 devient :

$$p(Comp_j\,|Tra_i\,) = \frac{p(Comp_j)p(Tra_i\,|Comp_j\,)}{\prod_k p(A_{i,k})} \tag{4.2.6}$$

$p(Comp_j)$ est une constante car sans autre condition, tous les utilisateurs ont la même probabilité d'expertise pour un concept. Sans aucune information préalable, la probabilité d'être le/la plus compétent(e) parmi les $|I|$ utilisateurs équivaut à extraire au hasard des groupes de N utilisateurs. Ainsi, une estimation de $p(Comp_j)$ est :

$$\hat{p}(Comp_j) = \frac{1}{|I|} \tag{4.2.7}$$

Dans notre proposition, la compétence des utilisateurs est mesurée par la fréquence des activités. Nous définissons $p(A_{i,k})$ comme le rang des meilleures fréquences parmi tous les utilisateurs. Ainsi, plus l'utilisateur i agit fréquemment sur le concept k, plus petite sera la probabilité $p(A_{i,k})$.

$p(Tra_i\,|Comp_j\,)$ représente la probabilité que l'utilisateur i réalise la trace Tra_i si l'utilisateur i possède le plus de compétences sur le concept j. Deux facteurs influencent cette valeur. Tout d'abord, si un utilisateur a le plus de compétences sur j, il est fort probable que l'utilisateur i possède beaucoup de compétences sur des concepts sémantiquement liés. Comme Tra_i est composé d'un ensemble d'activités $\{A_{i,k}\,|A_{i,k} \in Tra_i\}$, nous évaluons la distance sémantique entre j et chaque k. Nous utilisons $\omega_{k,j}$ pour représenter le poids du concept k sur j. Deuxièmement, compte tenu du poids entre le concept k et j, plus l'utilisateur i est classé haut pour le concept k, plus grande sera la probabilité $p(Tra_i\,|Comp_j\,)$. Nous définissons :

$$p(Tra_i\,|Comp_j\,) = \frac{1}{Z} \sum_{\{k|A_{i,k} \in Tra_i\}} [1 - p(A_{i,k})] \times \omega_{k,j} \tag{4.2.8}$$

dans laquelle Z est un facteur de normalisation dépendant de $\{A_{i,k}\,|A_{i,k} \in Tra_i\}$, c'est-à-dire une constante si les valeurs des variables caractéristiques sont connues. Nous obtenons :

$$p(Comp_j\,|Tra_i\,) = \frac{\sum_{\{k|A_{i,k} \in Tra_i\}} [1 - p(A_{i,k})] \times \omega_{k,j}}{N \times Z \times \prod_{k=1}^{n} p(A_{i,k})} \tag{4.2.9}$$

Finalement, nous obtenons $p(Comp_j\,|Tra_i\,)$ et en comparant la probabilité de tous les utilisateurs sur le concept, nous pouvons recommander quel est l'utilisateur le plus compétent/expert pour un concept/sujet donné à partir de ses traces.

Synthèse

Nous avons orchestré un modèle d'action, un modèle de compétence et une structure sémantique pour proposer des recommandations basées sur l'évaluation des traces en utilisant le TF-IDF ou la classification naïve Bayésienne. Notons que ces approches ont été implémentées et testées (via la plateforme collaborative MEMORAe [4]) et que les résultats répondent à nos attentes montrant que cette approche prend en charge les relations sémantiques entre les concepts.

Dans cette instanciation de notre modèle générique de recommandation, il apparait que :

- l'utilisateur cherche un expert dans son environnement de travail collaboratif sur un sujet donné,
- les données d'entrée à notre disposition sont l'ensemble des sujets (C), l'ensemble des utilisateurs (I), un log (modèle d'action) contenant les traces d'interactions des utilisateurs de l'environnement et la matrice d'utilité $C \times I$,
- les étapes sont :
  1. le prétraitement du log via l'ontologie d'application,
  2. la génération des candidats compétents via le TF-IDF ou la classification naïve Bayésienne,
  3. l'ordonnancement des candidats en fonction de leur valeur décroissante de compétence
- le système retourne un ensemble ordonné d'experts (ayant des compétences sur un sujet donné) qui permettront à l'utilisateur de trouver l'expert le plus compétent pour le sujet qui l'intéresse.

4. http ://memorae.hds.utc.fr/

4.2.2 Recommandations à visée publique

Dans cette section, nous nous intéressons aux recommandations à visée publique, c'est-à-dire qui s'adressent à un décideur (ou groupe de décideurs) mais qui seront utilisées pour prendre des décisions qui auront un impact sur les populations. Plus précisément, dans le cadre de nos travaux, nous nous focalisons sur des domaines tels que les villes « intelligentes » (*Smart cities*) et les systèmes d'alertes précoces pour la gestion de crises. En particulier, que ce soit pour les villes ou en gestion de crises, recommander des actions à mettre en place, ont été/sont des défis captivants.

4.2.2.1 Recommander des actions à mettre en place dans les villes

Le concept de « ville intelligente » (*Smart City*) n'a pas encore été clairement défini. Cependant, il y a six caractéristiques/catégories pour classer ce genre de villes et les comparer [GFK+07] : l'économie intelligente (*smart economy*), la mobilité intelligente (*smart mobility*), l'environnement intelligent (*smart environment*), des gens intelligents (*smart people*), une manière de vivre intelligente (*smart living*) et la gouvernance intelligente (*smart governance*). Notons que ces six caractéristiques de la ville « intelligente » sont liées aux citoyens de la ville, entre autres. Cependant, être « intelligent » est un défi de plus en plus important pour de nombreuses villes ou communautés. C'est d'un intérêt particulier dans le domaine des technologies de l'information et des communications (TIC) et pour de tels systèmes où il y a des enjeux économiques, sociaux et autres.

Une ville peut devoir décider par exemple, de quelles actions mettre en place pour atteindre un objectif particulier (comme l'une ou plusieurs des six caractéristiques de [GFK+07]) ou quelle(s) personne(s)/expert(s) appelé(s) dans le cadre d'un projet, ... Les décisions de la ville ont un impact sur les citoyens (population) et donc, les recommandations que nous allons pouvoir suggérer sont à visée publique.

Dans le cadre de nos travaux [NRS14, ANRS16, NRS15b], aider une ville à identifier les actions à mettre en œuvre pour atteindre son objectif (au sens des six caractéristiques de [GFK+07]) semble important. Recommander de telles actions est un domaine d'investigation émergent et prometteur.

Dans [NRS14], nous présentons un cadre pour un système de recommandation pour les villes. La portée de ce travail de recherche est de tirer avantage des villes reconnues « intelligentes » et de faire les mêmes actions pour les villes qui veulent devenir « intelligentes ». La méthode suivie est : à partir d'une liste des caractéristiques d'une « ville intelligente », et d'une ville qui veut devenir « intelligente », quelles actions doivent être mises en œuvre par cette ville pour devenir « intelligente » en se basant sur les caractéristiques « d'intelligence » données.

L'idée principale est de recommander à la ville les actions déjà mises en œuvre dans les « villes intelligentes » qui lui sont similaires (la similarité entre deux villes est basée sur des indicateurs tels que la qualité de l'air, la consommation d'eau, ...) ainsi que les actions à mettre en œuvre dans ladite ville.

Ainsi, en instanciant la définition 4.1.1, il est possible de définir une recommandation pour les « villes intelligentes » comme étant une action $a \in A$ (ensemble de toutes les actions possibles) à mettre en œuvre, telle que son utilité pour une ville $v \in V$ (ensemble de toutes les villes possibles) est maximale.

Définition 4.2.4 – *Recommandation pour les « villes intelligentes »*
Soit A l'ensemble de toutes les actions possibles et V l'ensemble de toutes les villes, étant donné un journal/log de villes et une ville qui veut atteindre un objectif (« intelligent »), et soit u une

fonction qui mesure l'utilité[5] *d'une action a pour une ville v, c'est-à-dire* $u : V \times A \rightarrow \mathbb{R}$. *Alors pour chaque ville* $v \in V$, *l'action recommandée* $a' \in A$ *est celle qui maximise l'utilité pour la ville :* $\forall v \in V, a'_v = argmax_{a \in A} u(v, a)$.

Exemple 4.2.4 – *De manière générale, il est possible d'illustrer la matrice d'utilité* $V \times A$ *où un score indique que l'action a été mise en œuvre et si la ville qui l'a mise en œuvre a considéré cette action comme efficace :*

u(v,a)	*Action 1*	*Action 2*	*Action 3*	*Action 4*	*Action 5*
Ville 1	*8*	*7*			
Ville 2	*9*		*3*		
Ville 3	*3*	*5*		*5*	*5*
Ville 4	*5*	*3*		*3*	*3*

Notons qu'une cellule (v, a) *de cette matrice d'utilité correspond au score d'utilité assigné à l'action a pour la ville v. Ici, les scores sont des valeurs entières entre 0 et 10 (0 étant la plus mauvaise et 10 la meilleure) indiquant l'efficacité de l'action a pour la ville v. Notons aussi qu'un exemple d'action peut être « après minuit, éteindre les lumières publiques ». Finalement, notons que les scores sont obtenus en donnant un score global mais pourraient être obtenus en combinant les notions de coût et de temps nécessaires à la mise en œuvre, ...*

Tout d'abord, nous restreignons notre espace de travail en faisant certaines hypothèses :
- la ville, qui a besoin de recommandations, choisit, au départ, une seule catégorie de villes intelligentes (parmi les six) qu'elle souhaite rejoindre,
- les valeurs des indicateurs sont numériques,
- les indicateurs sont les mêmes pour un(e) catégorie/facteur donné(e) pour chaque ville (seules les valeurs changent et peuvent être nulles).

Notre approche utilise à la fois les caractéristiques de la ville qui a besoin de recommandations et le log[6] contenant, pour chaque ville intelligente, sa catégorie, les facteurs, les indicateurs et les actions mises en œuvre. A l'image de notre approche générique (section 4.1 et figure 4.1), elle se décompose en trois étapes, comme illustré sur la figure 4.6 (les quatre parties colorées correspondent à celles de la figure 4.1) :

1. La première étape consiste à prétraiter les indicateurs des villes intelligentes dans le log et à restreindre la matrice d'utilité en fonction de la catégorie de ville intelligente choisie par la ville qui veut a besoin de recommandations (la ville courante),
2. La deuxième étape consiste à comparer les indicateurs de la ville actuelle et ceux des villes intelligentes enregistrées et à extraire les actions correspondantes,
3. La dernière étape consiste à classer les actions candidates à la recommandation.

Chaque étape de ce processus est présentée plus en détail ci-dessous.

5. Les notes/scores sont donnés par une personne autorisée à prendre la décision de mettre en œuvre différentes actions. Ce score indique si l'action est (ou était) pertinente.

6. Le log peut être une base de données ou une autre structure de données. Il sera alimenté par le système de recommandation lors de l'utilisation du système par les villes. Cependant, pour les données initiales (problème du démarrage à froid), nous espérons utiliser les données officielles, publiques et libres, éventuellement enrichies avec la participation de volontaires.

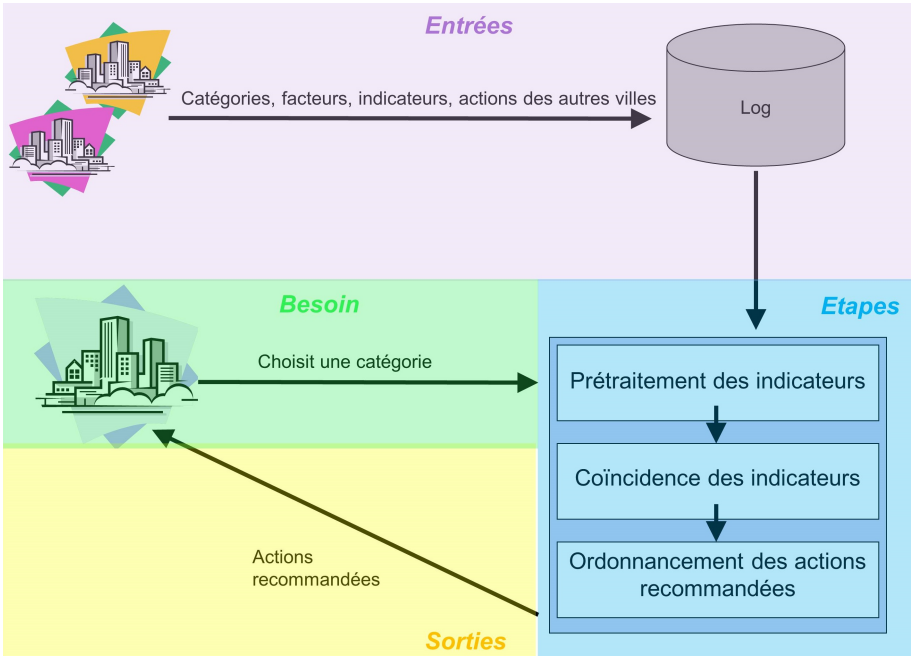

FIGURE 4.6 – Vue d'ensemble de notre approche de recommandation pour les villes intelligentes

Une ville peut être considérée comme un n-uplet contenant la description de la ville et les informations de la ville. L'information de la ville est un ensemble de catégories intelligentes où chaque catégorie est un ensemble de facteurs et chaque facteur est un 3-uplet spécifiant les indicateurs correspondants et leurs valeurs et pour chaque indicateur, l'ensemble des actions mises en œuvre. Notons que l'ensemble des catégories peut être vide si la ville n'est pas « intelligente » pour cette catégorie. Notons également que l'ensemble des actions mises en œuvre peut être vide si aucune action n'a été mise en œuvre.

Chaque ville C_i est ainsi associée à une description ($Description_i$) et à un ensemble de j catégories ($Category_{ij}$)[7]. Chaque catégorie contient un ensemble de k facteurs. Chaque facteur est associé à un ensemble de n triplets contenant le nom de l'indicateur, sa valeur et les actions mises en place dans le cadre de cet indicateur. Nous avons donc, pour chaque ville C_i, $\forall i, j, k, n \in \mathbb{N}^{+*}$:

$$C_i = \langle\, Description_i, \{\, Category_{ij}, \{\, Factor_{ijk},$$
$$\{\, \left\langle Indicator_{ijk1}, Value_{ijk1}, \left\{ Action^1_{ijk1}, ..., Action^{m_1}_{ijk1} \right\} \right\rangle, ...,$$
$$\left\langle Indicator_{ijkn}, Value_{ijkn}, \left\{ Action^1_{ijkn}, ..., Action^{m_n}_{ijkn} \right\} \right\rangle \,\}\,\}\,\}\,\rangle$$

où $\forall l \in \mathbb{N}^{+*}, m_l \in \mathbb{N}^+$, avec i : le nombre de villes, j : le nombre de catégories, k le nombre de facteurs, n le nombre d'indicateurs, m_l le nombre d'actions.

Par exemple, à partir d'un extrait de [GFK+07], la catégorie « Environnement intelligent » contient deux facteurs qui sont : « Développement durable » et « Pollution ». Le facteur « Pollution » est associé à l'indicateur « Qualité de l'air » qui, en fonction de la ville, aura une valeur et correspondra à un ensemble d'actions mises en place.

7. Dans le cas des travaux de [GFK+07], j vaut 6.

Dans le cadre des villes « intelligentes », la fonction $GenericReco$ devient $GenericReco(\emptyset, \emptyset, Log, Mu, PretraitementL, PretraitementM, Generation, Ordonnancement, \emptyset, \emptyset, \prec, c, Cons)$ où l'ensemble des villes est obtenu par le biais du log et l'ensemble des actions recommandables est obtenu via la fonction $Generation$. L'algorithme simplifié 2 représente notre processus. Un tel algorithme a une complexité de $O(mn^2)$ où m est le nombre de villes et n le nombre maximum d'indicateurs et/ou facteurs. Certains choix d'implémentation doivent être faits en partenariat avec les parties prenantes.

Pré-traitement des indicateurs
À partir du log Log des villes intelligentes (avec leurs descriptions, catégories, facteurs, indicateurs, valeurs d'indicateurs et actions) et la catégorie $Cons$ de ville intelligente choisie par la ville courante, cette première étape consiste à prétraiter les indicateurs des villes du log. En fait, nous proposons de calculer les intervalles de valeurs des indicateurs. Notons que notre calcul se limite aux indicateurs correspondant à la catégorie donnée/choisie $Cons$. Dans notre algorithme, nous utilisons la fonction $PretraitementL$. Cette fonction est utilisée pour calculer un ensemble d'intervalles de valeurs d'indicateurs. En fait, cette fonction $PretraitementL$ est utilisée pour chercher parmi les villes intelligentes (du log) ayant des facteurs et indicateurs correspondant à la catégorie donnée/choisie $Cons$, pour chaque indicateur $I_{ijkl}, \forall i, j, k, l \in \mathbb{N}^{+*}$, la valeur d'indicateur minimale V_{ijkl}^{min} et la valeur d'indicateur maximale V_{ijkl}^{max}. Cette fonction retourne un ensemble d'intervalles (un par indicateur I_{ijkl}) où V_{ijkl}^{min} et V_{ijkl}^{max} sont les bornes (inférieure et supérieure) de chaque intervalle $\left[V_{ijkl}^{min}, V_{ijkl}^{max}\right]$. Notons que le calcul de tels intervalles de valeurs d'indicateurs est possible en raison de nos hypothèses, c'est-à-dire que les valeurs d'indicateurs sont numériques et que les indicateurs sont les mêmes pour un(e) catégorie/facteur donné(e).

Coïncidence des indicateurs
La fonction de prétraitement de l'étape précédente retourne un ensemble d'intervalles de valeurs d'indicateurs (un par indicateur). L'objectif, ici, est de vérifier si les valeurs d'indicateurs de la ville courante correspondent aux valeurs des indicateurs des villes intelligentes du log, c'est-à-dire appartiennent aux intervalles précédemment calculés. Ainsi, pour chaque indicateur, si la valeur courante de l'indicateur $V_{current\ jkl}$ n'appartient pas à l'intervalle calculé $\left[V_{ijkl}^{min}, V_{ijkl}^{max}\right]$, alors l'ensemble correspondant d'actions $\{A_{ijkl}\}$ (de l'indicateur donné) est sélectionné (sans doublon). Le but étant que les valeurs d'indicateurs de la ville courante « entrent » dans les intervalles.

L'objectif de la fonction d'extraction, $Extract$, est d'extraire un ensemble d'actions qui serviront de base à la recommandation, c'est-à-dire que les actions extraites peuvent contribuer à aider la ville courante en rejoignant la catégorie choisie $Cons$. L'ensemble d'actions obtenu, $CandAct$, est l'ensemble non ordonné des recommandations qui est renvoyé. Il est à noter que nous sommes conscients qu'un tel ensemble d'actions peut être volumineux. Naturellement, il serait intéressant de prendre en compte les aspects environnementaux, économiques et sociaux lors de notre étape de « matching », comme l'indique [BW10, BLSW10].

Ordonnancement des actions
Dans l'étape précédente, un ensemble de recommandations est obtenu. Le but, ici, est de sélectionner, lorsque l'ensemble retourné $CandAct$ n'est pas vide, les actions les plus pertinentes en fonction d'un critère de satisfaction exprimé par la ville courante. À cette fin, un ordonnancement des actions est nécessaire, qui ordonne les recommandations candidates. Il existe de nombreuses façons d'ordonner les candidates, des plus simples aux plus sophistiquées. Parce qu'il est actuellement difficile d'intégrer

Algorithme 2 Recommandations d'actions pour les villes « intelligentes »

Entrées :

Log : Le log de villes intelligentes,
Mu : la matrice d'utilité,
PretraitementL : Une fonction de prétraitement des villes du log,
PretraitementM : Une fonction de prétraitement de la matrice Mu,
Generation : Une fonction de génération (combinaison de deux fonctions : Match et Extract) des actions candidates à la recommandation,
Ordonnancement : Une fonction d'ordonnancement des actions,
$\prec$: Un ordre sur les actions,
c : La ville courante qui a besoin de recommandations,
Cons : La catégorie de ville intelligente choisie par la ville courante.

Sorties : Un ensemble ordonné de recommandations

$CandCities \leftarrow \emptyset$ // *pour les villes candidates*
$CandAct \leftarrow \emptyset$ // *pour les actions candidates*
$Interval \leftarrow \emptyset$
$\mathcal{MU} \leftarrow PretraitementM(Mu, Cons)$
Pour chaque ville $c_i \in Log$ **Faire**
  Si $c_i.category = Cons$ **Alors**
    $CandCities \leftarrow CandCities \cup c_i$
  Fin Si
Fin Pour
$Interval \leftarrow PretraitementL(CandCities)$
Pour chaque facteur f_j de $Cons$ **Faire**
  Pour $k \in [1..n]$ // *n est le nombre d'indicateurs de* f_j **Faire**
    Si $Match(C_c.Cons.f_j.indicator(k), Interval_{Cons,f_j,k}) = False$ **Alors**
      Pour chaque ville $c_l \in CandCities$ **Faire**
        $CandAct \leftarrow CandAct \cup Extract(c_l.Cons.f_j.indicator(k))$
      Fin Pour
    Fin Si
  Fin Pour
Fin Pour
Si $CandAct \neq \emptyset$ **Alors**
  $Ordonnancement(CandAct, \prec)$
Fin Si
Retourne $CandAct$

les connaissances dans un système informatique afin d'automatiser la prise de décision, cette étape est faite par les parties prenantes.

Exemple 4.2.5 – *Dans cet exemple, nous illustrons plus concrètement la recommandation d'actions pour les villes intelligentes (les données sont artificielles). Supposons qu'il existe un log contenant des informations relatives à deux villes : Smallville et Metropolis. Un extrait du log pourrait être :*

Smallville = ⟨ *Petite ville des États-Unis,* { *environnement intelligent,* { *développement durable,* { ⟨ *consommation d'eau, 300 (litres par an),* { *après minuit stopper les fontaines publiques, ne pas arroser les plantes durant l'été* } ⟩, ⟨ *consommation d'électricité, 3000 (kw par an),* { *après minuit éteindre les éclairages publics, installer des lampadaires solaires* } ⟩ }, *pollution,* { ⟨ *qualité de l'air,* $\frac{10}{10}$, ∅ ⟩ } } } ⟩

Metropolis = ⟨ *Grande ville des États-Unis,* { *environnement intelligent,* { *développement durable,* { ⟨ *consommation d'eau, 100 000 (litres par an),* { *après minuit stopper les fontaines publiques, ne pas laver els voitures en été* } ⟩, ⟨ *consommation d'électricité, 200 000 (kw par an),* { *après minuit éteindre les éclairages publics*} ⟩ }, *pollution,* { ⟨ *qualité de l'air,* $\frac{8}{10}$, { *restreindre l'accès au centre-ville aux véhicules, réduire la vitesse des véhicules sur les grands axes* } ⟩ } } } ⟩

Supposons maintenant qu'une ville, nommée Gotham, veut atteindre un objectif « intelligent » pour la catégorie « environnement intelligent » où :
Gotham = ⟨ *Ville de Batman,* { *environnement intelligent,* { *développement durable,* { ⟨ *consommation d'eau, 150 000 (litres par an),* ∅ ⟩, ⟨ *consommation d'électricité, 100 000 (kw par an),* { *après minuit éteindre les éclairages publics*} ⟩ }, *pollution,* { ⟨ *qualité de l'air,* $\frac{8}{10}$, ∅ ⟩ } } } ⟩.

La matrice d'utilité correspondante pourrait être :

u(v,a)	A_1	A_2	A_3	A_4	A_5	A_6	A_7
Smallville	*9*	*7*		*8*	*8*		
Metropolis	*8*		*6*	*7*		*5*	*5*
Gotham				*8*			

où l'action A_1 est « après minuit stopper les fontaines publiques », A_2 est « ne pas arroser les plantes durant l'été », A_3 est « ne pas laver les voitures en été », A_4 est « après minuit éteindre les éclairages publics », A_5 est « installer des lampadaires solaires », A_6 est « restreindre l'accès au centre-ville aux véhicules » et A_7 est « réduire la vitesse des véhicules sur les grands axes ».

Les informations relatives à la catégorie « Environnement intelligent » sont connues pour les villes Smallville et Metropolis, dans le log. Donc, les villes qui peuvent permettre d'aider Gotham pour la catégorie « Environnement intelligent » sont Smallville et Metropolis.
Nous allons donc considérer que les valeurs de leurs indicateurs sont de bonnes valeurs pour être « intelligent ».

Pour cette catégorie, « Environnement intelligent », nous avons deux facteurs : « développement durable » et « pollution ». Pour le facteur « développement durable », nous avons deux indicateurs : « consommation d'eau » et « consommation d'électricité »[8]*. Les valeurs de « consommation d'eau »*

8. Pour cet exemple, nous nous sommes basés sur la catégorisation de [GFK+07] mais, il est possible d'imaginer que les décideurs de la ville aient élaboré leur propre catégorisation.

	Environnement intelligent		
	Développement durable		Pollution
	Consommation d'eau	Consommation d'électricité	Qualité de l'air
Intervalles	[300; 100 000]	[3 000; 200 000]	$\left[\frac{8}{10}; \frac{10}{10}\right]$
Gotham	150 000	100 000	$\frac{8}{10}$

TABLE 4.10 – Intervalles et valeurs pour Gotham

sont 100 000 *pour Metropolis et* 300 *pour Smallville. Donc, l'intervalle de valeurs correspondant est* [300; 100 000]. *Les valeurs de « consommation d'électricité » sont* 200 000 *pour Metropolis et* 3 000 *pour Smallville. Donc, l'intervalle de valeurs correspondant est* [3 000; 200 000]. *Pour le facteur « pollution », nous avons ici, un seul facteur : « qualité de l'air » et les valeurs sont* $\frac{8}{10}$ *pour Metropolis et* $\frac{10}{10}$ *pour Smallville. Donc, l'intervalle de valeurs correspondant est* $\left[\frac{8}{10}, \frac{10}{10}\right]$.
Le tableau 4.10 résume les intervalles et les valeurs de Gotham.

L'ensemble des actions (mises en œuvre dans les villes du log : Smallville et Metropolis) à proposer à Gotham correspond aux indicateurs pour lesquels les valeurs de Gotham sont en dehors de l'intervalle. Ici (voir tableau 4.10), la valeur de « consommation d'eau » de Gotham est la seule à être en dehors des limites de l'intervalle. Donc, les actions qui peuvent être intéressantes (sans redondance) sont les actions déjà mises en œuvre par Smallville et Metropolis pour l'indicateur « consommation d'eau » telles que « ne pas laver les voitures en été », « ne pas arroser les plantes durant l'été » et « après minuit stopper les fontaines publiques ».

Finalement, ces actions doivent être ordonnées. Par exemple, elles peuvent être ordonnées en fonction de la facilité de mise en œuvre. Il est plus facile et plus rapide de décréter que les fontaines publiques doivent être stoppées après minuit que d'obliger la population à ne pas utiliser d'eau en été. Donc, un ensemble ordonné possible d'actions à mettre en œuvre par Gotham pour améliorer son « intelligence environnementale » est : (1) « après minuit stopper les fontaines publiques », (2) « ne pas arroser les plantes durant l'été » et (3) « ne pas laver les voitures en été ».

Synthèse

Dans le cadre des villes « intelligentes », nous nous focalisons sur la recommandation d'éléments. Plus précisément, nous proposons une approche théorique de recommandations d'actions à mettre en œuvre dans une ville. Cette approche se décompose en trois étapes : prétraitement des indicateurs, coïncidence des indicateurs et ordonnancement des actions. Il est important de noter que les décisions qui seront prises auront un impact sur les citoyens. Aussi, quelle(s) que soi(en)t la (les) recommandation(s) suggérée(s) par notre système de recommandation, la décision finale est prise par les décideurs et doit être prise avec tous les acteurs, en tenant compte des aspects économiques, sociaux, financiers et tous les autres enjeux de ce type de projet.

Dans cette instanciation de notre modèle générique de recommandation, il apparait que :

- la ville cherche à atteindre un objectif particulier,
- les données d'entrée à notre disposition sont le log de villes avec les actions réalisées et la matrice d'utilité $Villes \times Actions$,
- les étapes sont :
  1. le prétraitement de la matrice d'utilité (en la restreignant à l'objectif) et le prétraitement du log (via les intervalles de valeurs),

2. la génération des actions candidates via les fonctions *Match* et *Extract*,
3. l'ordonnancement des actions candidates en fonction des préférences des parties prenantes

– le système retourne un ensemble ordonné d'actions pour permettre à la ville d'atteindre son objectif en les mettant en place.

Nous espérons, dans nos travaux futurs, étudier des cas réels avec des experts. Nous sommes conscients que dans certains cas, il est possible de s'attendre à un biais, parce qu'il s'agit d'actions humaines.

4.2.2.2 Recommander des actions à mettre en place pour la gestion de crises (système d'alertes précoces)

Un système d'alertes précoces peut être défini comme une chaîne de systèmes de communication d'information comprenant des capteurs, de la détection, de la décision et des sous-systèmes transitoires, dans cet ordre, qui travaillent en collaboration, pour prévenir et signaler des perturbations affectant négativement la stabilité du monde physique ; et qui donnent suffisamment de temps au système de réponse pour préparer les ressources et les mesures d'intervention (actions) pour minimiser l'impact sur la stabilité du monde physique [Wai10].

Dans [Neg13b], nous présentons une approche de système de recommandation de gestion de crises. Cette approche utilise les actions déjà mises en œuvre pour gérer les crises antérieures afin d'améliorer la gestion d'une crise donnée. L'idée principale est de recommander les actions déjà mises en œuvre lors de ces précédentes crises qui sont similaires (la similarité entre deux crises est basée sur des indicateurs tels que l'intervalle entre deux crises, le type (ouragan, tsunami, ...)) comme les actions à mettre en œuvre. Enfin, nous proposons d'exploiter les connaissances acquises à partir des expériences passées pour prendre la meilleure décision (à savoir, les meilleures actions à mettre en œuvre) afin de mieux gérer une crise imminente.

En instanciant la définition 4.1.1, une recommandation pour les alertes est définie comme une action $a \in A$ (ensemble de toutes les actions possibles) à mettre en œuvre, telle que son utilité pour une alerte $w \in W$ (ensemble de toutes les alertes possibles) est maximale.

Définition 4.2.5– *Recommandation pour alertes*
Soit A l'ensemble de toutes les actions possibles et W l'ensemble de toutes les alertes, étant donné un journal/log d'alertes, les indicateurs correspondants et une alerte courante déclenchée, et soit u une fonction qui mesure l'utilité d'une action a pour une alerte w, c'est-à-dire $u : W \times A \to \mathbb{R}$. Alors, pour chaque alerte $w \in W$, l'action recommandée $a' \in A$ est celle qui maximise l'utilité pour l'alerte : $\forall w \in W, a'_w = argmax_{a \in A} u(w, a)$.

Exemple 4.2.6– *Dans cet exemple, nous illustrons simplement la matrice obtenue $W \times A$ où un score indique que l'action a été mise en œuvre et si cette action a été considérée comme efficace :*

u(w,a)	*Action 1*	*Action 2*	*Action 3*	*Action 4*	*Action 5*
Alerte 1	*8*	*7*			
Alerte 2	*9*		*3*		
Alerte 3	*3*	*5*		*5*	*5*
Alerte 4	*5*	*3*		*3*	*3*

Notons qu'une cellule (w, a) de cette matrice correspond au score d'utilité assigné à l'action a pour l'alerte w. Ici, les scores sont des valeurs entières entre 0 et 10 (0 étant la plus mauvaise et 10 la meilleure) indiquant l'efficacité de l'action a lors de l'alerte w.

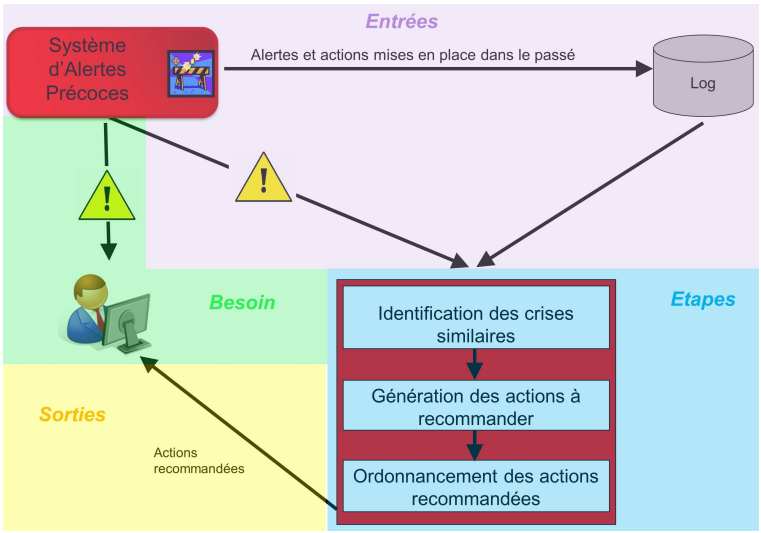

FIGURE 4.7 – Vue d'ensemble de notre approche de recommandation pour la gestion de crises

Notre approche utilise à la fois les caractéristiques des alertes/crises courantes et le log des alertes/crises antérieures, c'est-à-dire les actions mises en œuvre au cours de chaque crise précédente. A l'image de l'approche générique (section 4.1 et figure 4.1), elle se décompose en trois étapes, comme illustré sur la figure 4.7 (les quatre parties colorées correspondent à celles de la figure 4.1) :

1. La première étape consiste à identifier/sélectionner dans le log, des alertes similaires à l'alerte déclenchée en cours et à restreindre la matrice d'utilité aux crises/alertes similaires,
2. La deuxième étape consiste à extraire les actions mises en œuvre pour gérer les anciennes alertes similaires,
3. La dernière étape consiste à classer les actions candidates à la recommandation.

Dans le cadre des systèmes d'alertes précoces pour la gestion de crises, la fonction *GenericReco* devient $GenericReco(\emptyset, \emptyset, Log, Mu, Identité, PretraitementM, Generation, Ordonnancement, Defaut, t, \prec, c, Cons)$ où l'ensemble des alertes est obtenu par le biais du log et l'ensemble des actions recommandables est obtenu via la fonction *Generation*. L'algorithme 4 simplifié représente ce processus.

Sources de données

Pour une crise donnée, chaque fois qu'une alerte est déclenchée par le système d'alertes précoces, les décideurs doivent d'abord décider si cette alerte est suffisante pour être prise en compte. Si cette alerte est majeure, les décideurs doivent prendre des mesures pour faire face au problème signalé par le système d'alertes précoces. Ainsi, l'alerte déclenchée peut générer, ou non, la mise en œuvre d'un certain nombre d'actions. De notre point de vue, dans les deux cas, l'alerte déclenchée est importante et doit être enregistrée (dans un log).

Algorithme 4 Recommandations d'actions pour les systèmes d'alertes précoces

Entrées :
Log : Le log des précédentes alertes déclenchées,
Mu : la matrice d'utilité,
$Generation$: une fonction de génération des recommandations candidates (combinaison de Match, Extract et Clean) $Ordonnancement$: Une fonction d'ordonnancement des actions,
$Defaut$: Une fonction qui retourne une recommandation par défaut,
t : un seuil pour la fonction $Match$,
$\prec$: Un ordre sur les actions,
W_c : L'alerte (déclenchée) courante,
$Cons$: le type de crise.
Sorties : Un ensemble ordonné de recommandations

$SimW \leftarrow \emptyset$ // *pour les alertes similaires*
$CandAct \leftarrow \emptyset$ // *pour les actions candidates*
$\mathcal{MU} \leftarrow PretraitementM(Mu, Cons)$
Pour chaque alerte $w_i \in Log$ **Faire**
  $SimW \leftarrow SimW \cup Match(w_i, W_c, t)$
Fin Pour
Pour chaque alerte $w_j \in SimW$ **Faire**
  $CandAct \leftarrow CandAct \cup Extract(w_j)$
Fin Pour
$Clean(CandAct)$
Si $CandAct \neq \emptyset$ **Alors**
  Retourne $Ordonnancement(CandAct, \prec)$
Sinon
  Retourne $Defaut(L)$
Fin Si

Chaque alerte est associée à un ensemble d'indicateurs. Ces indicateurs doivent également être enregistrés. En fait, ces indicateurs sont des seuils, des intervalles de valeurs, des caractéristiques ...

Enfin, en plus de l'alerte et des indicateurs (et valeurs) correspondants, nous considérons que les actions mises en œuvre pour répondre à l'alerte déclenchée doivent également être enregistrées (dans un log).

Plus formellement, l'alerte déclenchée par le système d'alertes précoces peut être considérée comme un 3-uplet contenant la description de l'alerte, l'ensemble de paires contenant un indicateur et sa valeur et l'ensemble des actions mises en œuvre. Notez que l'ensemble des actions mises en œuvre peut être vide si le décideur ne décide pas de mettre en œuvre certaines actions lorsqu'une alerte est déclenchée par le système d'alertes précoces.

Ainsi, nous avons, pour chaque alerte W_i, $\forall i \in \mathbb{N}^{+*}$:

$$W_i = \langle\ Description_i,\ \{\ \langle Indicator_i^1, Value_i^1 \rangle, ..., \langle Indicator_i^n, Value_i^n \rangle\ \}\ ,\ \{ Action_i^1, ..., Action_i^m \}\ \rangle$$

où $m \in \mathbb{N}^+$ et $n \in \mathbb{N}^{+*}$ avec i : le nombre d'alertes, m : le nombre d'actions, n : le nombre d'indicateurs.

Notez que nous considérons que les indicateurs sont les mêmes pour chaque alerte (seules les valeurs correspondantes sont différentes).

Enfin, nos données sont composées d'un log d'alertes déclenchées et de l'alerte déclenchée en cours.

Identification des crises similaires

A l'aide du log des alertes déclenchées précédemment et de l'alerte déclenchée en cours, cette première étape consiste à identifier des crises/alertes similaires. Nous proposons d'utiliser une fonction de correspondance. Cette fonction est utilisée pour trouver un ensemble d'alertes correspondant à une alerte donnée. En fait, cette fonction de correspondance, $Match$, est utilisée pour rechercher dans l'ensemble des anciennes alertes celles qui correspondent à l'alerte déclenchée en cours. Cette fonction fournit un ensemble d'alertes similaires à l'alerte courante. Notez que cet ensemble retourné peut être vide.

Un premier exemple de cette fonction de correspondance consiste simplement à comparer, pour chaque alerte enregistrée, chaque paire $\langle Indicator_l, Value_l \rangle$ à chaque paire $\langle Indicator_c, Value_c \rangle$ de l'alerte courante où $Indicator_l = Indicator_c$. Cela revient à comparer $Value_l$ et $Value_c$.

Comme les valeurs des indicateurs peuvent être quantitatives ou qualitatives, certaines mesures de similarité peuvent être utilisées et/ou combinées. Une solution facile est d'utiliser la similarité Cosinus.

En fait, pour chaque alerte, le vecteur d'indicateurs $\vec{w_i}$ est composé des valeurs d'indicateurs $(Value_i^k, \forall k \in [1..n])$ tel que (d'après la notation donnée en section 4.2.2.2) :

$$\vec{w_i} = (Value_i^1, Value_i^2, ..., Value_i^n)$$

avec i : le nombre d'alertes et n : le nombre d'indicateurs.

Extraction des actions à recommander

La fonction de correspondance de l'étape précédente fournit un ensemble d'alertes enregistrées similaires, $SimW$, correspondant à l'alerte déclenchée en cours. L'objectif, ici, est d'extraire un ensemble d'actions qui serviront de base à la recommandation, via la fonction $Extract$. Notre but consiste à extraire pour chaque alerte de $SimW$, l'ensemble des actions correspondantes, sans doublon (via la fonction $Clean$). L'ensemble d'actions obtenu, $CandAct$, est l'ensemble des recommandations non ordonnées qui est renvoyé.

Ordonnancement des actions recommandées

Dans l'étape précédente, un ensemble de recommandations est obtenu. Le but, ici, est de sélectionner les actions les plus pertinentes par rapport à un critère de satisfaction exprimé par l'utilisateur. À cette fin, un classement d'actions est nécessaire, qui ordonne les recommandations candidates. Encore une fois, il existe de nombreuses façons d'ordonner ces recommandations candidates, des plus simples aux plus sophistiqués. Voici quelques exemples :

- ordonner les candidates en fonction de leur nombre d'occurrences,
- ordonner les candidates en fonction du profil de l'utilisateur. Par exemple, certaines actions peuvent parfois ne pas être mises en œuvre dans un contexte donné,
- ordonner les candidates en fonction de l'ordre des alertes correspondantes dans le log, c'est-à-dire les actions mises en œuvre le plus récemment seront proposées en premier. Par exemple, les canaux de diffusion évoluent avec le temps.

Recommandation par défaut

Comme indiqué précédemment, l'ensemble des recommandations candidates peut être vide : lorsqu'aucune alerte similaire n'est trouvée. Dans un tel cas, certains systèmes de recommandation sont en mesure de fournir à l'utilisateur une recommandation par défaut. Compte tenu de notre contexte de gestion de crises et des impacts et enjeux d'un tel contexte, et au vu de l'avancement de nos travaux, nous considérons qu'il est préférable de ne rien recommander comme recommandation par défaut. Proposer une recommandation par défaut efficace est un sujet de recherche intéressant.

Synthèse

Dans le cadre de la gestion de crises, nous abordons la recommandation d'actions à mettre en œuvre lorsqu'une alerte est déclenchée par un système d'alertes précoces. Notre approche se décompose en trois étapes : identification des crises similaires, extraction des actions à recommander et ordonnancement des actions.

Dans cette instanciation de notre modèle générique de recommandation, il apparait que :

- les décideurs cherchent à mettre en place des cations pertinentes lors du déclenchement d'une alerte,
- les données d'entrée à notre disposition sont le log contenant les précédentes alertes déclenchées avec les actions correspondantes et la matrice d'utilité $Actions \times Alertes$,
- les étapes sont :

  1. le prétraitement de la matrice d'utilité via la fonction $Match$ pour l'identification des crises similaires,
  2. la génération des actions candidates via la $Extract$,
  3. l'ordonnancement des actions candidates en fonction de leur récence ou de leur nombre d'occurrences

- par défaut, le système ne retourne rien en raison des enjeux importants,
- le système retourne un ensemble ordonné d'actions à mettre en place pour gérer une crise imminente lors du déclenchement d'une alerte.

Ce travail, qui utilise les connaissances acquises par les expériences passées pour prendre la meilleure décision dans le présent, est une étape vers le renforcement des systèmes d'alertes précoces et la gestion de crises.

4.3 Bilan

Afin de pallier la « sur-spécialisation » des systèmes de recommandation existants et de proposer des systèmes de qualité (au sens de l'ingénierie et de la qualité logicielle), dans ce chapitre, nous avons présenté un système générique de recommandation.

Plus formellement, notre système générique de recommandation est représenté sous la forme d'une fonction : $GenericReco(C, I, Log, Mu, PretraitementL, PretraitementM, Generation, Ordonnancement, Defaut, t, \prec, c, Cons)$ qui, à partir de l'ensemble des utilisateurs C, l'ensemble des entités recommandables I, un log/journal relatif aux éléments de I : Log, une matrice d'utilité Mu, une fonction de prétraitement du log : $PretraitementL$, une fonction de prétraitement de la matrice d'utilité : $PretraitementM$, une fonction de génération de recommandations sur l'ensemble I : $Generation$, une fonction d'ordonnancement des recommandations candidates : $Ordonnancement$, une fonction qui retourne une recommandation par défaut (au cas où le système ne trouve pas de recommandations candidates) : $Defaut$, un seuil $t \in \mathbb{R}$ (pour la fonction de génération), un ordre sur I : $\prec$ (pour la fonction d'ordonnancement), l'utilisateur/décideur : $c \in C$ pour qui le système va suggérer des recommandations, et une contrainte de restriction $Cons$ (pour le prétraitement de la matrice d'utilité), retourne un ensemble de recommandations.

Afin de valider notre modèle générique, nous l'avons instancié pour proposer, soit des recommandations individuelles (qui s'adressent à un utilisateur (ou groupe d'utilisateurs) et qui sont utilisées dans un cadre restreint à l'utilisateur (ou au groupe d'utilisateurs)), soit des recommandations à visée publique (qui s'adressent à un décideur (ou groupe de décideurs) mais qui seront utilisées pour prendre des décisions qui auront un impact sur les populations).

Pour les recommandations individuelles, nous nous sommes focalisés sur (i) la recommandation de requêtes OLAP (*On-Line Analytical Process*) pour aider un décideur à naviguer au sein d'un cube de données, (ii) la recommandation de ressources/supports dans les plateformes d'apprentissage numérique pour aider les apprenants à combler certaines lacunes et (iii) la recommandation d'experts dans les environnements de travail collaboratif pour améliorer notamment la collaboration.

Pour les recommandations à visée publique, nous nous sommes focalisés sur (i) la recommandation d'actions à mettre en place dans les villes pour atteindre un objectif donné, et (ii) la recommandation d'actions à mettre en place lorsqu'une alerte précoce est déclenchée pour limiter les effets d'une crise imminente.

Par ailleurs, la distinction entre recommandations individuelles et recommandations à visée publique existe dans le type de recommandations suggérées mais se retrouve aussi dans les paramètres utilisés dans la fonction *GenericReco*. Ainsi, pour les recommandations individuelles, $PretraitementM$ (et $Cons$) ne sont pas obligatoires alors que pour les recommandations à visée publique $PretraitementM$ (et $Cons$) le sont. De même, C et I sont nécessaires dans le cadre de recommandations individuelles alors qu'ils peuvent être induits dans le cadre de recommandations à visée publique.

Le tableau 4.11 récapitule les besoins, entrées, étapes et sorties (c'est-à-dire les quatre parties de notre système générique de recommandation, illustrées par la figure 4.1) et la recommandation retournée par défaut, des différentes instanciations présentées dans ce chapitre.

Enfin, il est à noter que nos différentes instanciations du modèle générique fonctionnent et, à partir des expérimentations réalisées, les résultats obtenus, de notre point de vue, sont convaincants.

		Entrepôts de données	Plateforme d'apprentissage	Environnement de travail collaboratif	Ville « intelligente »	Système d'alertes précoces
Besoin		Explorer un cube de données	Combler des lacunes	Trouver des experts sur un sujet donné	Atteindre un objectif défini	Mettre des actions en place lors du déclenchement d'une alerte
Entrées	Types	• Requête : ensemble de membres, • Session : séquencement de requêtes	• Ressource (hétérogène), • Apprenant	• Utilisateur, • Sujet : texte	• Action : n-uplet, • Ville : n-uplet	• Action : n-uplet, • Alerte : n-uplet (hétérogène)
	Log	Requêtes / sessions précédemment lancées	Agent gestionnaire	Traces d'interaction	Les villes et leurs actions	Les alertes et les actions correspondantes
	Mu	Requêtes × Sessions	Ressources × Apprenants	Utilisateurs × Sujets	Actions × Villes	Actions × Alertes
Etapes	Pré-traitement	Partition du log OU identité	Réduction de Mu	Ontologie d'application	Intervalles de valeurs	Similarité Cosinus
	Génération	Coïncidence et représentant (de session)	Décomposition en valeurs singulières	TF-IDF OU Classification Bayésienne	Appartenance aux intervalles	Similarité entre alertes
	Ordonnancement	Proximité	Ordre décroissant des valeurs prédites	Ordre décroissant des valeurs prédites	Préférences des parties prenantes	Nombre d'occurrences des actions OU Récence des actions
Défaut		Requête faisant autorité	Ressource la plus populaire et ressource la plus récente			
Sorties		Ensemble ordonné de requêtes OLAP	Ensemble ordonné de ressources	Ensemble ordonné d'experts sur un sujet donné	Ensemble ordonné d'actions	Ensemble ordonné d'actions

TABLE 4.11 – Tableau récapitulatif des différentes instanciations

Par conséquent, contrairement aux travaux existants dans la littérature, nous proposons un modèle générique de recommandation qui a été validé pour (au moins) cinq domaines d'applications différents (avec des objectifs et des enjeux différents), à savoir les entrepôts de données (OLAP), les plateformes d'apprentissage en ligne, les environnements de travail collaboratif, les villes intelligentes et les systèmes d'alertes précoces.

Chapitre 5

Evaluer un système de recommandation

Sommaire

5.1 Evaluation subjective du système de recommandation 79
5.1.1 Prise en compte des connaissances 79
5.1.2 Perception du système 80
5.1.2.1 Indicateur de perception pour les systèmes d'alertes précoces . . . 82
5.1.2.2 Expérimentations 84
5.2 Evaluation subjective des recommandations 87
5.2.1 Application à OLAP 88
5.2.1.1 Cas d'étude 88
5.2.1.2 Protocole expérimental 89
5.2.1.3 Utilisateurs et requêtes 90
5.2.1.4 Système d'évaluation sémantique 91
5.2.1.5 Expérimentations et résultats 91
5.3 Bilan 93

Dans les chapitres précédents, nous avons introduit différentes techniques de recommandation ainsi qu'un système générique de recommandation et diverses applications. Les techniques et systèmes évoluent au cours du temps en essayant d'être de plus en plus proches des attentes/besoins des utilisateurs. Pour cela, il est nécessaire d'évaluer les systèmes de recommandation et ainsi vérifier s'ils sont pertinents/performants pour l'utilisateur/décideur en fonction de la situation, des objectifs à atteindre, du temps de réponse, de la prise en compte de certains critères, ...

Lorsque l'on parle de l'évaluation des systèmes de recommandation, il faut avant tout distinguer l'évaluation de la (ou des) recommandation(s) (retournée(s) par le système de recommandation) de l'évaluation du système de recommandation (informatique) [HKTR04, dW08]. Ainsi, il est possible d'évaluer la qualité de la recommandation, c'est-à-dire sa pertinence ou son intérêt pour un utilisateur donné. L'évaluation du système de recommandation quant à elle, se focalise sur la capacité du système informatique à fonctionner correctement et à être performant, par exemple en prédisant correctement des scores ou en ordonnant les recommandations [HKTR04]. Il est à noter que, parfois, l'évaluation de la recommandation est utilisée pour évaluer le système de recommandation.

Par ailleurs, il existe deux grandes familles d'évaluation : l'évaluation objective et l'évaluation subjective [CGT13]. Est objectif ce « qui a rapport, qui répond à un objet, qui représente un objet tel qu'il est en réalité, sans aucune déformation de notre esprit, de nos tendances personnelles » [1].

1. D'après le Dictionnaire de l'Académie française, 8e édition : http ://atilf.atilf.fr/academie.htm.

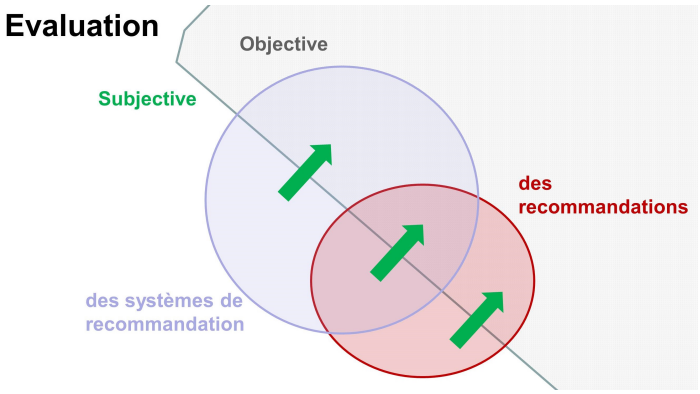

FIGURE 5.1 – Evaluation subjective et objective de la recommandation

A l'inverse, est subjectif ce « qui a rapport au sujet. Il se dit de ce qui se passe dans notre esprit, de ce qui est en nous » [1].

La figure 5.1 illustre notre constat où il est possible d'évaluer soit le système de recommandation, soit la(les) recommandation(s). Cette évaluation peut être objective ou subjective. Et l'évaluation subjective vient compléter/améliorer l'évaluation objective (qui utilise l'évaluation subjective dans ses calculs).

Il existe de nombreuses mesures pour évaluer la qualité des systèmes de recommandation (voir le chapitre 3 et [HKTR04, Neg15b]). Les plus utilisées sont les mesures de justesse (*accuracy*). La plupart de ces mesures supposent que meilleurs sont les algorithmes, meilleures sont les recommandations (d'un point de vue algorithmique) et intègrent dans leurs calculs l'évaluation des recommandations par les utilisateurs sans indiquer comment cette évaluation (subjective) est réalisée. Malheureusement, la justesse (et les autres mesures) ne constitue que partiellement l'expérience d'un utilisateur face à un système de recommandation. De ce fait, ces mesures, de notre point de vue, sont plus proches des mesures d'évaluation objective que de celles d'évaluation subjective. Plusieurs chercheurs ont fait valoir le fait qu'il existe d'autres facteurs qui influencent l'expérience de l'utilisateur, en parlant de l'évaluation subjective de l'utilisateur quant à son interaction avec le système, comme, par exemple, la diversification, la situation, des considérations personnelles, ... [KWG+12]. L'évaluation subjective est peu traitée dans la littérature de par la difficulté de sa mise en place et/ou de celle à impliquer les utilisateurs, or elle permet de positionner l'utilisateur (pour qui les recommandations doivent être pertinentes) au cœur de l'évaluation.

Ainsi, notre objectif, dans ce chapitre, est de proposer des modèles et mesures permettant d'évaluer, de manière subjective (en se concentrant sur l'utilisateur qui reçoit la (les) recommandation(s)) :

- un système de recommandation, via la prise en compte des connaissances des utilisateurs ou via la perception qu'ont les utilisateurs du système de recommandation,
- ou la (les) recommandation(s) retournée(s), via la prise en compte de la satisfaction des utilisateurs.

Nous nous intéressons donc aux recommandations d'éléments et/ou d'utilisateurs, individuelles ou à visée publique, sans amélioration (comme illustré sur les figures 3.4 et 3.5). Ce chapitre est organisé de la manière suivante : dans la section 5.1 nous présentons nos propositions d'évalua-

tion subjective du système de recommandation et dans la section 5.2 celles relatives à l'évaluation subjective des recommandations.

5.1 Evaluation subjective du système de recommandation

5.1.1 Prise en compte des connaissances

Un système de recommandation peut être vu comme un Système d'Information et de Connaissance (SICO). Un SICO [AGRS15] est un ensemble qui repose sur un tissu socio-technique (individus en interaction entre eux, avec des machines, et avec le système lui-même). Il comprend (i) un système d'information, constitué d'individus qui, dans un contexte donné, sont des processeurs de données auxquelles ils donnent un sens sous forme d'informations, et (ii) un système de connaissance, constitué de connaissances tacites incarnées par les individus et de connaissances explicitées formalisées et codifiées sur toute forme de supports. En fait, faisant référence aux travaux de [Tsu93], nous soutenons l'idée que la connaissance dépend du modèle mental de l'individu et du contexte de son action. Une information peut donc être mal comprise dans la mesure où elle prend sens au travers du schéma d'interprétation d'une personne donnée. Ainsi, le concepteur du système de recommandation doit prendre en compte le fait que l'utilisateur reçoit, traite et restitue les informations au travers de ses schémas d'interprétation.

La norme ISO, un des standards pour la gestion de la qualité, inclut la norme ISO 9001 :2008 « Systèmes de management de la qualité - Exigences » définit des exigences portant sur l'organisation d'un système de gestion de la qualité. Ainsi, un système de recommandation de bonne qualité devrait permettre la meilleure gestion (capitalisation, diffusion, création, ...) de la connaissance, celle-ci passant non seulement par la gestion des connaissances explicitées (par exemple via leur traitement et diffusion) mais aussi des connaissances tacites. Les connaissances tacites sont portées par les individus et ne sont pas forcément formalisables, donc pas forcément exploitables à travers le système informatique. Pourtant les connaissances tacites jouent un rôle important dans le fonctionnement du système d'information (SI) [MSS04].

Notre objectif est de proposer un modèle d'évaluation subjective des systèmes de recommandation qui prend en compte les connaissances (explicitées et tacites) des utilisateurs afin d'avoir des systèmes de recommandation de meilleure qualité et toujours plus proches des besoins/attentes des utilisateurs.

Afin de prendre en compte l'éclairage apporté par le SICO à l'évaluation d'un système de recommandation, nous proposons dans [ADG+12], une extension du modèle de [Pal07] en ajoutant un point de vue « prise en compte de la connaissance ». Il s'agit de regarder le système de recommandation en considérant l'humain comme un composant à part entière du système et non plus comme un simple utilisateur.

Ainsi, le point de vue de la prise en compte de la connaissance se décline sous trois aspects :

1. mettre en place des activités pour créer, pérenniser et utiliser les connaissances,
2. être conscient que la connaissance n'est pas un objet, et
3. connaître les conditions et limites pour qu'une connaissance codifiée et formalisée prenne le même sens quel que soit l'utilisateur.

Le premier aspect est induit assez naturellement par la nécessité qu'il y a aujourd'hui de considérer les connaissances comme des ressources. Des ressources qu'il faut créer, entretenir, pérenniser

et utiliser ; des ressources qu'il faut manager.

Le second aspect repose entre autres sur les théories de [Tsu93]. En effet, la connaissance résultant de l'interprétation par un individu d'une information, le sens que prend une information peut diverger d'un individu à l'autre.

Le dernier aspect résulte du constat que la connaissance peut ne pas être considérée comme un objet et que les conditions qui feront qu'un individu sera capable d'interpréter correctement les informations sources de connaissances seront inconnues.

Ainsi notre modèle d'évaluation subjective des systèmes de recommandation (extension du modèle de critères de [Pal07] avec prise en compte des connaissances), illustré sur la figure 5.2, est un modèle d'étude de la maturité de la prise en compte des connaissances (dans les systèmes de recommandation). De notre point de vue, ce niveau de maturité (évalué via les critères/indicateurs de notre modèle d'évaluation subjective) est corrélé avec la qualité du système de recommandation. Notons que les aspects introduits dans cette partie seront confrontés à des études sur le terrain et nous amèneront à concevoir des niveaux de maturité pour chacun en nous inspirant du modèle CMMI [2] (*Capability Maturity Model Integration*) [CKS03].

Synthèse L'évaluation d'un utilisateur de la pertinence/performance d'un système de recommandation est fonction de la situation et de certains critères (entre autres). Parce que nous voyons un système de recommandation comme un système d'information (SI) et, plus particulièrement, comme un système d'information et de connaissance (SICO) (qui considère les individus tout à la fois comme des utilisateurs et des composants du système, porteurs de connaissances), les connaissances de l'utilisateur, de notre point de vue, ont aussi un impact sur l'évaluation du système de recommandation. Ainsi, évaluer un système de recommandation passe par l'évaluation de la prise en compte des connaissances dans un tel système. Parmi les nombreux cadres pour l'évaluation de la qualité des SI, nous avons choisi celui de [Pal07] dont la vision des SI incluant les utilisateurs est la plus proche de celle que nous considérons. Nous avons donc proposé une extension de ce modèle en incluant le point de vue « prise en compte de la connaissance » via notre modèle d'étude de la maturité de la prise en compte des connaissances dans les systèmes de recommandation comme un modèle d'évaluation subjective des systèmes de recommandation. Nous tirons ainsi avantages des performances des modèles de [Pal07] et [CKS03] et nous ajoutons la prise en compte de la connaissance qui, de notre point de vue, permettra une meilleure évaluation du système de recommandation.

5.1.2 Perception du système

La perception est un concept général utilisé dans divers domaines tels que la philosophie, les études sociales, la biologie, l'art et la plupart du temps avec des significations différentes [Hoc56, DSiS91]. De notre point de vue, la perception est le processus de collecte, d'organisation et d'interprétation des stimuli qui peuvent être des informations et des connaissances provenant de différentes sources, médias, discussions avec d'autres personnes, sons ou alertes visuelles. Ils sont interprétés selon les profils personnels, la personnalité, les connaissances, l'expérience, ... Généralement, la perception est évaluée en fonction de sa cohérence avec la réalité et, de ce fait, n'est pas associée à une mesure/valeur quantitative. Néanmoins, des « indicateurs de perception » sont reconnus et utilisés pour améliorer les systèmes informatisés dans des domaines tels que le marketing,

2. Le modèle CMMI définit une échelle de mesure de la maturité à cinq niveaux (*Initial, Managed, Defined, Quantitatively managed, Optimizing*), ainsi que les indicateurs nécessaires pour évaluer les réalisations en fonction de cette échelle.

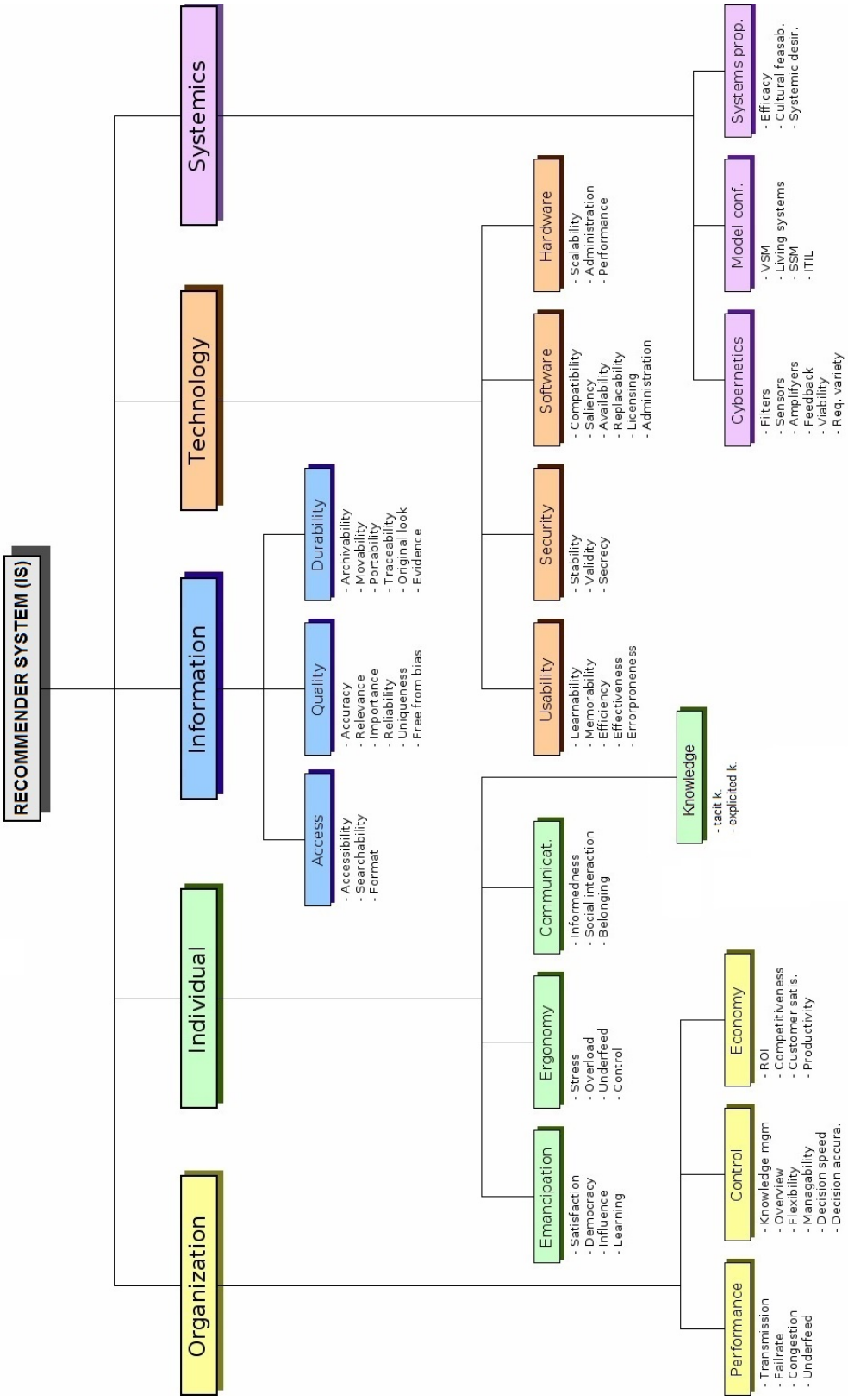

FIGURE 5.2 – Modèle d'évaluation subjective des systèmes de recommandation avec prise en compte des connaissances (adapté de [Pal07])

ce qui atteste que la quantification de la perception peut avoir des implications concrètes dans des systèmes comme les systèmes de recommandation [Lud04, AH09].

En informatique, le concept de perception des personnes/utilisateurs a été initié par le modèle d'acceptation technologique (*Technology Acceptance Model* - TAM) et de la théorie unifiée de l'acceptation et de l'utilisation de la technologie (*Unified Theory of Acceptance and Use of Technology* - UTAUT) [Dav89, GS97, LIC03]. Ainsi, avoir une « bonne perception » du système (de recommandation) va accroitre la « confiance » de l'utilisateur dans le système (de recommandation) et donc, il sera plus enclin à accepter/suivre les recommandations retournées.

Notre objectif est donc de proposer un indicateur de perception comme mesure d'évaluation subjective des systèmes de recommandation. Il est à noter que nous nous intéressons à la perception que les utilisateurs ont du système de recommandation à évaluer. A notre connaissance, il n'existe pas d'indicateur de perception pour évaluer un système de recommandation. Une méthodologie pour mesurer la perception des utilisateurs doit inclure un grand panel de personnes, pour construire un indicateur de perception efficace, facile à mettre en place et global.

Dans [AMN16], nous avons donc proposé, dans le domaine des systèmes d'alertes précoces, un indicateur de perception, que nous avons testé sur un cas réel : le système de sécurité incendie de l'Université Paris-Dauphine (France). Il est à noter qu'ici un système d'alertes précoces est vu comme un système de recommandation qui recommande d'évacuer à la population (les utilisateurs) en fonction de seuils critiques atteints (scores d'utilité).

5.1.2.1 Indicateur de perception pour les systèmes d'alertes précoces

Dans le contexte du risque, les décisions ne sont pas destinées aux individus, mais aux communautés, et une mesure de perception des citoyens peut être considérée comme un indicateur de qualité à prendre en compte par les décideurs. Il peut fournir une mesure de *feedback* pour les experts et les décideurs mettant en évidence les problèmes humains. Un autre intérêt d'un indicateur de perception est d'encourager les gens à se poser des questions sur les risques auxquels ils sont exposés et leur niveau de préparation.

Afin d'agir avec efficacité et rapidité, les décideurs doivent avoir une vision globale et synthétisée de la réalité, il est important de leur fournir une mesure efficace avec une échelle suffisamment précise et facile à interpréter.

Pour rappel, nous souhaitons proposer une mesure d'évaluation subjective des systèmes de recommandation. Notre intérêt se concentre donc sur les facteurs subjectifs de perception. En référence aux facteurs de perception proposés par [OW04], nous conservons uniquement ceux centrés sur les personnes, à savoir : (i) la familiarité avec le danger et la procédure à suivre, en fonction des expériences passées, (ii) la participation volontaire pour agir lorsqu'une alerte est déclenchée, (iii) l'importance des bénéfices et des intérêts associés aux réponses aux catastrophes, (iv) faire confiance aux autorités responsables du système et de la sécurité et (v) le sentiment vis-à-vis des capacités de gestion de risques. Afin d'évaluer ces cinq facteurs subjectifs de perception, nous avons réalisé des enquêtes fondées sur des questionnaires à l'attention des citoyens.

5.1.2.1.1 Introduction d'un questionnaire

Pour créer un questionnaire, nous nous sommes référés à des recherches sur les facteurs de confiance, de perception et de comportement dans le domaine des crises volcaniques et des feux

[HBP08]. Le questionnaire s'inspire également des cinq questions sur les indices de confiance des consommateurs [Lud04]. Pour obtenir un indicateur efficace, facile à mettre en place couvrant un large panel de répondants, nous avons limité l'enquête à six questions à choix multiples et à une question ouverte facultative. Cette dernière permet aux participants de proposer des améliorations.

Ainsi, les questions à choix multiples visent à couvrir le maximum de facteurs de perception dans une liste minimale d'éléments. Les premières et dernières questions évaluent intentionnellement un sentiment général. La deuxième question évalue l'utilité des systèmes d'alertes précoces considérés en fonction du sentiment global évalué lors de la première question. La troisième question porte sur la connaissance des gens sur ce qu'ils doivent faire en cas d'alerte, mais aussi une auto-évaluation volontaire de la capacité d'agir pendant les événements. Elle mesure le niveau de sensibilisation et les efforts d'éducation réalisés. La quatrième question est également une évaluation de la connaissance et du niveau de participation et de la satisfaction des personnes. Enfin, la dernière question vise à évaluer la sensibilité aux fausses alertes.

- Question 1 : Je me sens en sécurité dans la zone à risque par rapport à un risque donné (q1).
- Question 2 : Le système d'alertes (précoces) opérationnel en place est utile pour ma sécurité (q2).
- Question 3 : Je sais ce que je dois faire en cas d'alerte dans la zone à risque (q3).
- Question 4 : Le processus à suivre en cas d'alerte dans la zone à risque pourrait être amélioré (q4).
- Question 5 : J'agis selon le processus de sécurité chaque fois que j'entends l'alerte dans la zone à risque (q5).
- Question 6 : En général, je considère que la sécurité dans la zone à risque est satisfaisante (q6).

Pour faciliter les réponses aux questions et les rendre intuitives, chaque question admet cinq réponses similaires ordonnées selon l'échelle de Likert [Lik32] : « Pas du tout d'accord » (n2), « Pas d'accord » (n1), « Ni en désaccord ni d'accord » (med), « D'accord » (p1), « Tout à fait d'accord » (p2) et une réponse neutre par défaut « Ne se prononce pas ».

5.1.2.1.2 Calcul de l'indicateur

Cette section vise à montrer comment combiner les réponses, obtenues à partir du questionnaire ci-dessus, dans un jugement unique et global interprété comme la perception du système d'alertes précoces. Considérons $N = \{1, ..., n\}$ un ensemble de n répondants aux six questions à choix multiples $\{q1, q2, q3, q4, q5, q6\}$. Nous désignons par r_{ij} la réponse de l'individu j à la question q_i pour $i \in [1, 6], j \in [1, n]$. Par conséquent, l'indicateur de perception est défini comme une agrégation de toutes les r_{ij} réponses. La perception est donnée par la formule $f(r_{ij}, \forall i \in [1, 6], j \in [1, n]) \in [0, 1]$, où f est une fonction d'agrégation. Trouver une fonction d'agrégation appropriée est un problème bien connu dans les domaines de la théorie de la décision ou de la statistique. Ici, nous proposons deux instanciations de la fonction f : f_m, le score moyen de perception et f_s, le score de sentiment.

1. Score moyen de perception : basé sur une moyenne classique [3], il est défini comme suit :

   $$f_m(r_{11}, r_{12}, ..., r_{6n}) = \frac{g_m(1)+g_m(2)+...+g_m(6)}{6}.$$

   où g_m est une fonction d'agrégation qui associe à chaque question à choix multiples i, le nombre

   $$g_m(i) = \frac{(r_{i1}+r_{i2}+...+r_{in})}{n}.$$

3. Ici, chaque réponse a le même poids/impact sur la perception.

Dans ce cas, nous supposons que chaque réponse est enregistrée sur une échelle numérique ordinale 0, 0.25, 0.5, 0.75 et 1 représentant n2 (réponse négative 2), n1 (réponse négative 1), med (réponse médiane), p1 (réponse positive 1) et p2 (réponse positive 2) respectivement. Dans cette formule, les réponses neutres ne sont pas prises en compte.

2. Score de sentiment : il s'inspire des indices mensuels de confiance des consommateurs qui mesurent également un sentiment global. Pour chaque question à choix multiples i, nous comptabilisons le nombre total de réponses positives p1 et p2 (respectivement les réponses négatives n1 et n2) données par les n répondants notés $pos(i)$ (resp. $neg(i)$). Par conséquent, nous avons :

$$f_s(r_{11}, r_{12}, ..., r_{6n}) = \frac{g_s(1)+g_s(2)+...+g_s(6)}{6}$$

où g_s est une fonction d'agrégation qui associe à chaque question i, le nombre

$$g_s(i) = \frac{pos(i)}{pos(i)+neg(i)}.$$

Dans cette formule, les réponses médianes (med) et neutres ne sont pas prises en compte.

Les résultats sont des valeurs normalisées comprises dans l'intervalle $[0; 1]$ qui peuvent être multipliées par les décideurs pour les ramener à l'échelle d'interprétation souhaitée en fonction des contextes et de leurs niveaux de décision.

5.1.2.2 Expérimentations

Dans cette section, nous présentons notre étude où nous avons expérimenté notre questionnaire pour calculer l'indicateur de perception du système de sécurité incendie de notre Université selon nos deux instanciations f_m et f_s.

5.1.2.2.1 Contexte

Notre étude a été construite et testée à l'Université Paris-Dauphine (France) d'octobre à novembre 2015 pour tester la perception de la sécurité incendie du personnel et des étudiants. L'Université Paris-Dauphine est un établissement public qui a la particularité de concentrer plusieurs types de bâtiments et d'activités avec des règles spécifiques : amphithéâtres, salle de sport, bibliothèques, restaurants, parking, librairie et commerces. Il y a cinq bâtiments pouvant brûler en cinq minutes. La majorité des étudiants, des conférenciers et des chercheurs universitaires, du personnel permanent et des invités fréquentent l'université tous les jours. En 2014, l'université comptait 8750 étudiants à plein temps, 2020 professeurs, chercheurs et conférenciers et 460 employés administratifs. Une fois par an, une simulation est organisée sur le site complet de l'Université, avec des alertes et l'évacuation des personnes. Une formation est également proposée chaque année pour le personnel bénévole. En ce qui concerne les étudiants, seules des formations spécifiques de deux heures sont proposées pour des événements associatifs spécifiques qu'ils organisent. Nous précisons qu'à la date de notre sondage, aucune formation ni simulation n'a été réalisée depuis le début de l'année académique.

5.1.2.2.2 Protocole

Nous avons lancé une enquête anonyme en ligne activée du 30 octobre 2015 au 06 novembre 2015, créée avec le logiciel Open Source Limesurvey [4]. Des informations ont été demandées aux

4. https ://www.limesurvey.org

Score	[0,1]	]1,2]	]2,3]	]3,4]	]4,5]
Niveau	1	2	3	4	5

TABLE 5.1 – Correspondance entre les scores et les niveaux de notre indicateur

participants sur leur catégorie d'âge, leur statut, leur sexe et la participation ou pas dans le passé à une formation sur la sécurité incendie. Le corps de l'enquête a été composé de questions à choix multiples avec six réponses possibles de « Pas du tout d'accord » à « Tout à fait d'accord » et la réponse neutre par défaut « Ne se prononce pas », et d'une question facultative ouverte sur les suggestions d'amélioration. Les questions posées ont été adaptées du questionnaire où la zone à risque est l'Université, le risque encouru est l'incendie et le système d'alertes précoces est le système de sécurité incendie de l'Université.

Un lien hypertexte vers l'enquête a été envoyé par courrier électronique à 1187 membres du personnel de l'Université et aux courriels personnels des étudiants de sept classes de 25 à 30 personnes. De plus, le lien a été posté sur l'espace Internet privé des étudiants. Aucune récompense n'a été proposée aux participants.

5.1.2.2.3 Résultats quantitatifs

253 étudiants et membres du personnel ont répondu au questionnaire au cours des dix jours où l'enquête a été ouverte. Plus de 10% des membres du personnel ont répondu au questionnaire, avec 140 réponses sur les 1187 courriels envoyés.

Selon les réponses obtenues, le score moyen de perception (obtenu avec f_m) est de 2.93, ce qui est nettement inférieur au score de sentiment (obtenu avec f_s) qui est de 3.33. Le score de sentiment ignore les réponses « Ni en désaccord ni d'accord » et donne la même importance aux réponses « Pas d'accord » et « D'accord » qu'aux réponses « Pas du tout d'accord » et « Tout à fait d'accord ».

Pour faciliter la lecture des résultats, nous réduisons les scores continus de 0 à 5 en nombres entiers de 1 à 5 avec 5 niveaux correspondant au nombre de réponses possibles dans le questionnaire. Un score de 1 se réfère à une forte inutilité quant à la capacité du personnel et du système à gérer les risques, et un score de 5 met en évidence une forte sécurité.

Le tableau 5.1 indique la correspondance entre les scores et les intervalles où :

- $[0, 1]$: le niveau 1 est considéré comme critique et des efforts de communication et d'éducation sont nécessaires.
- $]1, 2]$: le niveau 2 est plutôt mauvais et les décideurs doivent s'interroger sur les raisons de cette mauvaise perception.
- $]2, 3]$: niveau 3, les gens ont une perception moyenne du système, ils ne se sentent pas en danger mais ils ne considèrent pas que tout est fait pour leur sécurité.
- $]3, 4]$ et $]4, 5]$: niveaux 4 et 5, les gens ont une bonne perception du système et il existe une forte probabilité avec un score de 5 que les gens soient bien informés et se comportent de manière appropriée au moment des événements ou des exercices.

Afin de considérer la rentabilité des systèmes, nous recommandons aux autorités et aux experts d'assurer un minimum de bon niveau de perception, avec un score d'indicateur supérieur à 3 (avec f_m comme avec f_s).

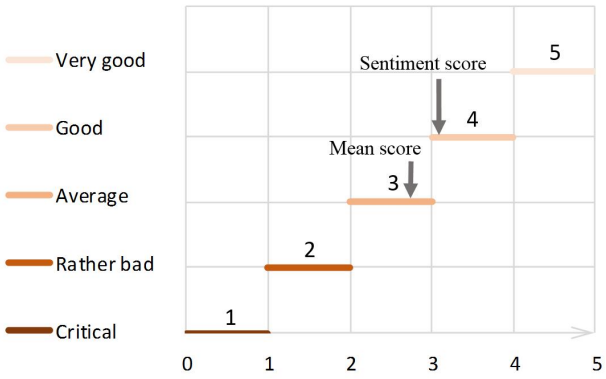

FIGURE 5.3 – Indicateur de perception

Pour le système de sécurité incendie de l'Université, les valeurs d'interprétation sont différentes pour les deux formules (comme montré sur la figure 5.3). Cela pourrait être différent si nous avions choisi une autre échelle avec 3 niveaux par exemple. L'échelle d'interprétation est subjective et le nombre de niveaux de l'échelle doit dépendre du contexte et être discuté par les décideurs.

Si le score est très bas (inférieur à 3), nous considérons que les résultats doivent être interprétés et discutés par les experts et les décideurs pour comprendre les raisons et décider comment (ré)agir. Les raisons peuvent être diverses : problèmes de communication, incohérences de la connaissance des risques ou problèmes de diffusion des alertes. Pour l'interprétation, l'analyse des résultats qualitatifs peut être utile.

5.1.2.2.4 Résultats qualitatifs

Les personnes ayant répondu à la question ouverte ont identifié des questions, des propositions d'amélioration et des demandes. Les 71 réponses obtenues sont spécifiques au contexte, nous les avons classées en quatre catégories :

- Demandes d'information et de formation,
- Propositions d'améliorations techniques,
- Fausses alertes,
- Problèmes divers.

Par exemple, (i) la nécessité de plus de communication et de formation est fortement exprimée, (ii) des propositions claires et précises sont faites par les personnes pour améliorer la sécurité incendie, (iii) les personnes considèrent que les fausses alertes sont dépassées, (iv) les problèmes relatifs à l'évacuation, les alertes et les capacités matérielles sont finalement mises en évidence par les participants, ...

Les résultats qualitatifs doivent être considérés et compris avec prudence, certaines demandes peuvent être considérées comme contradictoires. Par exemple, les gens demandent plus d'exercices et moins de fausses alertes, mais en étant attentifs aux détails, les réponses montrent que les fausses alertes pourraient être mieux acceptées si des informations étaient fournies avant et après les exercices.

Les expériences personnelles peuvent avoir une forte influence sur les réponses quantitatives et qualitatives. Cela peut être considéré comme un biais qui peut être limité avec l'éducation et la proximité des personnes. Les contextes peuvent également avoir une influence sur les réponses et le questionnaire doit être adapté dans des cas spécifiques.

Synthèse
La perception que les utilisateurs ont du système de recommandation a un impact sur l'utilisation qu'ils en font ainsi que sur la confiance qu'ils ont en lui (et envers les recommandations retournées). De ce fait, cette perception, de notre point de vue, doit être prise en compte dans l'évaluation du système de recommandation. Nous proposons donc un indicateur de perception comme mesure d'évaluation subjective des systèmes de recommandation. Cet indicateur mesure la perception globale des utilisateurs à l'égard du système de recommandation. Nous avons présenté deux manières de le calculer (score moyen de perception et score de sentiment). Notre indicateur permet de synthétiser les composantes subjectives de la perception pour indiquer si le système est mal ou très bien perçu.

Nous l'avons mis en place dans le cadre des systèmes d'alertes précoces (vus comme des systèmes de recommandation qui recommandent d'évacuer à la population) où il pourrait être à l'origine d'améliorations opérationnelles, d'événements de communication et d'éducation plus fréquents, ou de l'implication des citoyens dans la définition des indicateurs de risque et de vulnérabilité.

Notre étude sur le système de sécurité incendie de l'Université Paris-Dauphine a permis des discussions avec les experts du système sur leurs besoins pour plus d'exercices d'éducation et de simulation. Notons que l'étude présentée était une approche préliminaire pour tester au niveau local et organisationnel l'intérêt de notre proposition.

5.2 Evaluation subjective des recommandations

Comme indiqué précédemment, certaines mesures utilisées pour l'évaluation d'un système de recommandation impliquent une mesure de l'évaluation des recommandations retournées (par exemple, les mesures de justesse) sans indiquer comment cette évaluation (subjective) est réalisée/obtenue. Cette évaluation subjective des recommandations retournées, appelée satisfaction de l'utilisateur, est rarement traitée dans la littérature [dW08]. Par conséquent, dans cette section, nous tentons de pallier ce manque en proposant une mesure d'évaluation subjective des recommandations retournées en étudiant la satisfaction des utilisateurs.

Pour cela, nous définissons un cadre pour évaluer quantitativement la satisfaction des utilisateurs, en définissant comment les utilisateurs peuvent affirmer si une recommandation est intéressante, et en définissant les paramètres pour conduire des expériences. Nous nous concentrons donc sur la satisfaction des utilisateurs, et plus particulièrement, nous nous intéressons à la façon dont les utilisateurs peuvent déterminer si une recommandation est utile/intéressante.

A partir de [CSMF00], nous avons identifié deux questions principales qui reflètent la satisfaction de l'utilisateur en fonction du comportement du système de recommandation :

Question 1. La recommandation représente-t-elle des informations similaires à celles de mes éléments courants et avec une bonne précision, c'est-à-dire des informations pertinentes ?

Question 2. Le temps de réponse du système de recommandation est-il tolérable ?

Le cadre d'évaluation que nous proposons repose sur ces deux questions principales et doit être réalisé par divers utilisateurs sur différentes recommandations.

La question 1 sert à évaluer si la recommandation est associée à la similarité et à évaluer la performance sémantique de la recommandation. Une manière de quantifier le caractère sémantique de la recommandation consiste à définir n catégories de scores où un score faible est attribué par les

utilisateurs aux recommandations qu'ils considèrent totalement inutiles/inintéressantes, et un score élevé est associé aux recommandations bien adaptées aux besoins/attentes des utilisateurs.

La question 2 (temps de réponse) est importante car les utilisateurs sont rarement patients face à un système informatique. La question 2 est une question booléenne, où 1 indique un temps inférieur à un temps d'attente tolérable et 0 indique le contraire.

Enfin, la mesure d'évaluation globale est : Question 1 * Question 2 (où la réponse à la question 1 $\in [\![0; n]\!]$ et la réponse à la question 2 $\in [\![0; 1]\!]$).

Dans la suite, nous présentons la mise en place de notre cadre d'évaluation subjective des recommandations dans le domaine des entrepôts de données (et plus particulièrement OLAP).

5.2.1 Application à OLAP

Dans le domaine des entrepôts de données et OLAP (*On-Line Analytical Process*), certains travaux traitent de l'évaluation des systèmes de recommandation [CSMF00, GR11] en proposant un ensemble de paramètres quantitatifs tels que le nombre de dimensions/faits et des paramètres de satisfaction axés sur l'utilisateur. Cependant, ces paramètres orientés ne sont pas suffisants pour établir un cadre d'évaluation subjective des recommandations en OLAP pour trois raisons : (i) ils concernent le système d'entreposage de données global sans se concentrer sur le processus d'interrogation ; (ii) ils ne définissent pas de directives pour quantifier l'exactitude de la requête recommandée, et (iii) ils ne proposent pas de paramètres pour la réalisation d'expériences.

Nos contributions dans [BN16] pour l'évaluation subjective des recommandations sous forme de requêtes OLAP sont doubles. Tout d'abord, nous définissons un cadre pour évaluer quantitativement la satisfaction des utilisateurs, en définissant comment les utilisateurs peuvent affirmer si une requête recommandée est intéressante, et en définissant les paramètres pour conduire des expériences (utilisateurs et catégories de requêtes). Deuxièmement, ce travail est la première étude de l'évaluation subjective des recommandations sous forme de requêtes OLAP. En effet, à notre connaissance, bien que plusieurs travaux aient proposé des systèmes de recommandation en OLAP (par exemple, [JRTZ09, Neg09]), ils ne les ont pas évalués en fonction des données et des utilisateurs du monde réel.

5.2.1.1 Cas d'étude

L'entrepôt de données utilisé a été conçu dans le cadre d'un projet national français, *Energetic*. Ce projet vise à intégrer des données énergétiques sur les activités des exploitations agricoles afin de produire des valeurs de référence pour l'évaluation du cycle de vie et de gérer les consommations énergétiques [BPB+13]. La figure 5.4 présente le schéma conceptuel de l'entrepôt de données. Le fait est « Operations ». Il est analysé selon 8 dimensions. Les huit dimensions sont les suivantes : ***Campaign*** - cycles de production exprimés en années (par exemple, blé produit en 2009), ***Time*** - dimension temporelle classique, dans laquelle les jours sont regroupés par mois et par année, ***Products*** - les produits entrants et sortants (les produits sont regroupés de manière récursive en grandes catégories de produits), ***Operators*** - les personnes qui effectuent l'opération, ***Equipment*** - machines et outils utilisés, ***Location*** - la dimension spatiale qui regroupe les parcelles par ferme, département et région, ***Productions*** - type de production (par exemple, blé), et ***Technical Operations*** - les opérations techniques effectuées, regroupées par fonctions. Les mesures utilisées sont la surface travaillée ($surface_{\ w}$), la quantité de produit (entrée représentée par « intrant » et la sortie notée « extrant ») utilisée pendant le travail ou en dehors du travail (notée « w » et' « nw »,

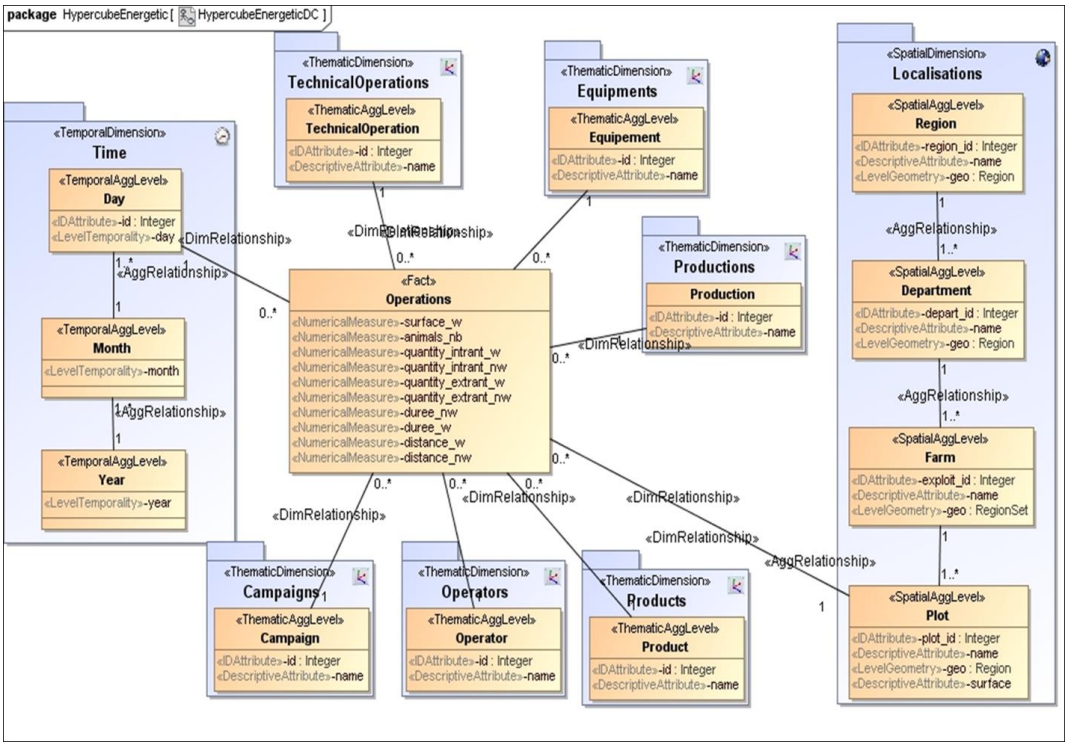

FIGURE 5.4 – Modèle conceptuel de l'entrepôt de données spatiales relatif à la consommation énergétique

respectivement), la durée en heures et la distance parcourue (*distance* $_w$ et *distance* $_n w$). Les mesures sont agrégées en utilisant la somme et la moyenne sur toutes les dimensions. En utilisant ce modèle multidimensionnel, il est possible de représenter et d'agréger des valeurs énergétiques selon différents axes d'analyse, tels que la consommation de carburant par parcelle, par opération technique, par année et par production.

Afin de mener notre étude, nous utilisons le système de recommandation de requêtes OLAP : RecoOLAP [Neg09, GMN09, GMN08, Neg14] (présenté dans la section 4.2.1.1) selon les 4 instanciations déjà validées objectivement (dans [Neg09]) : ClusterH, ClusterSP, EdH et EdSP (voir le tableau 4.1).

5.2.1.2 Protocole expérimental

Appliquées à ce cas d'étude, les deux questions principales de notre cadre d'évaluation subjective des recommandations deviennent :

Question 1. La requête OLAP recommandée représente-t-elle des informations similaires à celles de l'une des sessions d'analyse courantes et avec une bonne précision, c'est-à-dire des informations pertinentes ?

Question 2. Le temps de réponse [5] du système de recommandation est-il tolérable ?

5. La question 2 est importante car les requêtes OLAP doivent être résolues en peu de temps pour permettre une

Ce cadre d'évaluation est réalisé par divers utilisateurs sur différentes requêtes OLAP. Notre cadre d'évaluation pour les recommandations en OLAP est présenté dans le tableau 5.2. Les deux questions principales sont évaluées pour chaque combinaison d'utilisateurs et de types de requêtes OLAP tels que définis dans les sections suivantes. Puis, la mesure globale correspondra à Question 1 * Question 2 (où la réponse à la question 1 $\in [\![0;3]\!]$ et la réponse à la question 2 $\in [\![0;1]\!]$).

5.2.1.3 Utilisateurs et requêtes

L'entrepôt de données relatif à la consommation énergétique peut fournir des réponses aux questions de deux types d'utilisateurs : gestionnaires agricoles et praticiens de l'évaluation du cycle de vie (LFC). Ces deux catégories d'utilisateurs possèdent des compétences en technologie de l'information différentes et ont des besoins décisionnels différents. Étant donné que les recommandations s'adressent à des utilisateurs avec des profils différents [CSMF00], il est obligatoire d'évaluer notre proposition selon différents types de décideurs afin de déterminer si la qualité des requêtes recommandées dépend des compétences des utilisateurs. Ensuite, nous identifions deux types d'utilisateurs : qualifiés et non-qualifiés. Les utilisateurs qualifiés sont des décideurs ayant des connaissances en OLAP et/ou en système d'information. De plus, afin d'évaluer le caractère générique de notre acdre d'évaluation, il doit être testé selon différents types de requêtes.

En outre, l'analyse OLAP sur un entrepôt de données peut être effectuée de différentes façons. Par exemple, du point de vue des gestionnaires agricoles, une requête OLAP utile (c'est-à-dire, des indicateurs (agrégés)) serait le nombre d'interventions par culture. Pour les praticiens LFC, les questions concernant les inventaires de l'évaluation du cycle de vie sont importantes. Par exemple, il peut être intéressant de savoir quelle quantité de produit est nécessaire pour épandre une parcelle de blé et quelle est la consommation moyenne de carburant requise pour épandre l'ensemble de la ferme chaque année. En particulier, les praticiens LFC utilisent l'entrepôt de données pour comparer les valeurs de référence pour certaines opérations techniques (par exemple, labourer) avec des données entreposées. Ces requêtes sont caractérisées en utilisant uniquement un sous-ensemble de dimensions de l'entrepôt de données car les valeurs de référence ne sont pas disponibles, par exemple, par opérateur, équipement, mois et parcelle. Ces requêtes nécessitent des requêtes agrégées selon certaines dimensions. La figure 5.5 illustre un exemple où le praticien LFC s'intéresse aux opérations techniques de fertilisation. Nous appelons ce type de requêtes[6] des requêtes « LFC_q ».

Les deux autres types de requêtes sont définis par les gestionnaires agricoles pour améliorer leurs consommations énergétiques. Le premier type de requêtes concerne l'analyse des énergies utilisées pour les différentes opérations techniques (mesure : « intrant ») et le second type de requêtes est utilisé pour analyser les performances des opérateurs (mesure : « duree »). Nous appelons ces types de requêtes des requêtes « Farm Intrant » et « Farm time », respectivement. Ces requêtes sont définies en utilisant deux mesures particulières et en utilisant toutes les dimensions de l'entrepôt de données car des niveaux détaillés de dimensions sont nécessaires. Notons que les requêtes « LFC_q » et « Farm Intrant » sont définies sur la même mesure (« intrant »), mais les niveaux des dimensions sont différents. En résumé, l'analyse OLAP peut être effectuée de deux façons : *Requêtes agrégées*[7]

analyse interactive. De plus, OLAP est un processus exploratoire, donc, le temps de réponse doit être inférieur à 10 secondes pour une requête OLAP.

6. Un type de requêtes regroupe différentes requêtes ayant les mêmes propriétés.

7. Les requêtes agrégées contiennent des fonctions d'agrégation (min, max, moyenne, ...) et permettent donc de

Requête × Utilisateur	Qualifié	Non-qualifié
Requêtes agrégées		
Requêtes détaillées		

TABLE 5.2 – Protocole d'évaluation

Type de requêtes	Requête	Utilisateur
LFC_q	Agrégée	Qualilfié
Farm Time	Détaillée	Non-qualifié
		Qualifié
Farm Intrant	Détaillée	Non-qualifié
	Agrégée	

TABLE 5.3 – Sessions d'analyses

			Localisation
			+All Localisations
			Produits
Measures	Operation_technique	Temps	+energie indirecte
intrant_quantite_w	+Fertilisation	+All Tempss	

FIGURE 5.5 – Requête agrégée : exemple d'une requête « LFC_q »

(par exemple, les requêtes LFC_q) et *Requêtes détaillées* (par exemple, les requêtes « Farm Intrant » et « Farm Time »).

5.2.1.4 Système d'évaluation sémantique

Pour quantifier le caractère sémantique de la requête recommandée [8], nous avons défini quatre catégories de scores de 0 à 3. Un score de 0 est attribué par les utilisateurs aux requêtes recommandées qu'ils considèrent comme totalement inutiles, et un score de 3 est associé à des requêtes bien adaptées aux besoins des décideurs. Parce que les utilisateurs fournissent leur évaluation sans paramètre quantitatif, pour que le système d'évaluation soit utilisable par les décideurs, nous avons décidé d'utiliser uniquement 4 classes, car de cette façon, la tâche d'évaluation peut être effectuée facilement par nos utilisateurs. De plus, un nombre pair de choix possibles empêche les utilisateurs de donner une réponse « neutre ».

5.2.1.5 Expérimentations et résultats

Le système de recommandation utilisé, RecoOLAP, nécessite un cube de données et un ensemble de sessions sur ce cube comme entrées. Le cube utilisé est celui correspondant à l'entrepôt de données relatif à la consommation énergétique présenté précédemment. Le log/journal contenant un ensemble de sessions (précédentes) a été obtenu en enregistrant les sessions de requêtes lancées sur notre cube par différents utilisateurs (ces utilisateurs sont présentés dans la section 5.2.1.3). Notre log/journal est composé de 7 sessions, où chaque session contient entre 7 et 23 requêtes. Chaque session de test est lancée par un utilisateur qui a la même expertise que les utilisateurs présentés dans la section 5.2.1.3. Nous avons 10 sessions de test (2 sessions pour chacune des 5 possibilités) qui explorent différentes parties du cube, où chaque session contient entre 1 et 3 requêtes. Le tableau 5.3 répertorie les 5 sessions d'analyse que nous avons évaluées [9]. Le log/journal et les sessions de test sont lancés, enregistrés et traités à l'aide de RecoOLAP.

regrouper certaines données/informations. Les requêtes détaillées, quant à elles, ont un niveau de détail très fin sur l'ensemble des dimensions de l'entrepôt de données.

8. Pour chaque session lancée par un utilisateur donné, le système de recommandation renvoie une requête recommandée.

9. Les 5 sessions correspondent aux 5 lignes du tableau 5.3, c'est-à-dire au croisement d'un type de requêtes avec une requête et un utilisateur : soit 1 session avec LFC_q, 2 avec $FarmTime$ et 2 avec $FarmIntrant$.

Dans cette section, nous nous concentrons uniquement sur la Question 1 de notre cadre d'évaluation subjective des recommandations, car la Question 2 (temps de réponse) a déjà été évaluée comme étant bonne dans [Neg09] en utilisant une variable quantitative mesurée en secondes. Pour rappel, *RecoOLAP* [Neg09] est le système présenté dans la section 4.2.1.1. Il a été instancié et testé selon 4 combinaisons : ClusterH, ClusterSP, EdH et EdSP. Selon [Neg09], sur données synthétiques, EdH et EdSP ont de meilleures performances que ClusterH et ClusterSP (pertinence et temps).

Ainsi, à partir du cube de données et du log/journal présenté précédemment, RecoOLAP recommande une requête par session courante lancée par l'un des utilisateurs. Ensuite, l'utilisateur donne un score entre 0 et 3 à la requête recommandée délivrée par le système. Nous testons chaque session de test avec chaque instanciation présentée précédemment : ClusterH, ClusterSP, EdH et EdSP, avec 3 valeurs possibles de γ ($\gamma = 0$, $\gamma = 0.5$ et $\gamma = 1$) couplée avec la distance entre requêtes ($d^{\gamma}_{queries}$) [Neg09]. L'analyse a été effectuée pour chaque type de requêtes et pour chaque type d'utilisateurs (voir le tableau 5.3). Pour les scores de satisfaction donnés par les utilisateurs LFC pour leurs deux sessions correspondantes [10], pour la session 1, la meilleure combinaison semble être ClusterH avec $\gamma = 1$. Pour la session 2, les meilleures combinaisons semblent être EdH et EdSP.

En analysant les valeurs moyennes et les écarts-type pour les scores de satisfaction donnés par les utilisateurs qualifiés LFC pour leurs deux sessions correspondantes avec des requêtes agrégées, EdH et EdSP semblent fournir des résultats plus pertinents. Notons que pour d'autres types d'utilisateurs et de requêtes, les résultats obtenus sont assez similaires.

En observant les valeurs moyennes et les écarts-type des scores donnés à l'ensemble des sessions courantes pour chaque instanciation et pour chaque γ, nous remarquons que pour les instanciations utilisant un algorithme de clustering (ClusterH et ClusterSP), la moyenne est meilleure lorsque $\gamma = 0$. De même, pour les instanciations utilisant la distance de Levenshtein (EdH et EdSP), la moyenne est pire lorsque $\gamma = 1$ et équivalente lorsque $\gamma = 0.5$. Par conséquent, nous pouvons conclure que, en moyenne, les résultats les plus pertinents sont obtenus lorsque $\gamma = 0$ et que la distance de Hausdorff permet, à elle seule, d'obtenir la meilleure pertinence (la distance basée sur les dimensions contribue négativement à la distance entre requêtes).

Nous remarquons également que pour les instanciations utilisant un algorithme de clustering (ClusterH et ClusterSP), en moyenne, les résultats de ClusterH sont assez semblables à ceux de ClusterSP. Cela est également vrai pour les instanciations utilisant la distance de Levenshtein : les résultats d'EdH sont assez semblables à ceux d'EdSP.

Par conséquent, nous pouvons conclure que l'utilisation d'une distance entre les membres basée sur le plus court chemin ou sur la distance de Hamming ne semble pas avoir une incidence sur la pertinence des recommandations. Cela s'explique par la spécificité des hiérarchies de notre entrepôt de données qui sont de type parent-enfant. en effet, la distance de Hamming et celle basée sur le plus chemin retournent le même résultat dans le cas de relations parent-enfant (ce n'est pas le cas pour des hiérarchies plus complexes).

Enfin, avec toutes les sessions de test fusionnées, la meilleure moyenne est obtenue pour EdH ($\gamma = 0$ ou $\gamma = 0.5$), puis pour EdSP ($\gamma = 0$ ou $\gamma = 0.5$). En outre, le plus faible écart-type est obtenu pour EdH ($\gamma = 0$ ou $\gamma = 0.5$), puis pour EdSP ($\gamma = 0$ ou $\gamma = 0.5$). Nous considérons que nous avons de bons résultats lorsque la moyenne est élevée et lorsque l'écart-type est faible. Par conséquent, nous pouvons conclure que les instanciations utilisant la distance de Levenshtein (EdH

10. Pour rappel, nous avons 2 sessions par utilisateur possible et donc 2 sessions pour les utilisateurs LFC qui sont des utilisateurs qualifiés avec des requêtes agrégées.

et EdSP) proposent des recommandations plus pertinentes. Cette conclusion fondée sur des données réelles est conforme à celle de [Neg09] sur des données synthétiques.

Nous avons remarqué que la plupart du temps, le score le plus bas, 0, est attribué lorsque la requête recommandée ne contient pas la mesure recherchée. Cela signifie que la requête recommandée n'est pas du tout similaire à la session d'analyse courante. Un score de 3 est donné lorsque des mesures et des membres de dimensions sont présents dans la requête recommandée, en particulier, lorsque la requête recommandée atteint une « bonne précision » ou un niveau de détail. Un score de 1 est attribué lorsque la mesure est présente mais qu'aucune dimension intéressante n'est proposée. Enfin, un score de 2 est attribué lorsque la mesure est présente et lorsque des niveaux intéressants se trouvent également dans la requête recommandée. Il est important de noter que la précision de la requête recommandée est définie par les dimensions, les niveaux et les membres concernés. Par conséquent, nous pouvons conclure qu'une bonne requête recommandée est une requête très similaire aux précédentes. À la lumière de ces conclusions, nous pouvons affirmer que EdH avec $\gamma = 0$ semble être l'instanciation qui renvoie les recommandations les plus pertinentes.

Synthèse L'évaluation subjective des recommandations en étudiant la satisfaction des utilisateurs n'a été que très rarement traitée dans la littérature. Or, l'évaluation des systèmes de recommandation passe souvent par l'évaluation des recommandations elles-mêmes sans être détaillée. Nous proposons donc un cadre d'évaluation subjective des recommandations qui repose sur deux questions : l'une relative à la qualité des recommandations retournées et l'autre au temps de traitement.

Nous avons mis en place notre cadre dans le domaine des entrepôts de données. A notre connaissance, il s'agit du premier travail de recherche sur l'évaluation subjective de recommandations en OLAP (satisfaction des utilisateurs). Nos expérimentations ont été réalisées sur un entrepôt de données concernant la consommation énergétique agricole, *Energetic*, avec différents types d'utilisateurs et de requêtes, via le système de recommandation de requêtes OLAP, *RecoOLAP*. Les résultats obtenus montrent que les performances de *RecoOLAP* sur données synthétiques sont similaires à celles sur données réelles. Par conséquent, nous appuyons le fait que la recommandation de requêtes OLAP est utile dans une étude de cas réel et confirmons l'importance de ces outils académiques.

5.3 Bilan

Les systèmes et techniques évoluent au cours du temps en essayant d'être toujours plus proches des attentes/besoins des utilisateurs. Pour cela, il est nécessaire d'évaluer les systèmes de recommandation et ainsi vérifier s'ils sont pertinents/performants pour l'utilisateur. La majorité des mesures utilisées pour les systèmes de recommandation sont des mesures d'évaluation objective. Malheureusement, la plupart de ces mesures intègrent dans leurs calculs l'évaluation des recommandations par les utilisateurs sans indiquer comment cette évaluation subjective est obtenue. L'évaluation subjective soulève des défis quant à la difficulté de sa mise en place et/ou de celle à impliquer les utilisateurs.

Nous proposons donc des évaluations subjectives car, bien qu'il soit difficile d'obtenir des retours/*feedbacks* des utilisateurs, l'objectif d'un système de recommandation reste avant tout de les satisfaire en étant au plus proche de leurs attentes/besoins.

Dans ce chapitre, nous avons présenté différentes propositions liées à l'évaluation des systèmes de recommandation. Plus particulièrement, nous avons proposé deux évaluations subjectives des systèmes de recommandation : (i) en prenant en compte les connaissances des utilisateurs, via un

modèle d'étude de la maturité et de la connaissance et (ii) en mesurant la perception qu'ont les utilisateurs du système de recommandation via un indicateur de perception. Nous avons également proposé un cadre d'évaluation subjective des recommandations en mesurant la satisfaction des utilisateurs.

Nos propositions ont pu être validées sur des données et des utilisateurs réels. Ainsi, les expérimentations que nous avons pu mener ont montré l'intérêt de nos propositions ainsi que leur applicabilité.

Finalement, les systèmes de recommandation doivent donc être de plus en plus centrés sur les utilisateurs mais leur évaluation aussi.

Chapitre 6

Améliorer la recommandation

Sommaire

6.1 Améliorer les données d'entrée : Structuration de logs/journaux 97
6.1.1 Sous forme de graphe 98
6.1.1.1 Construction du graphe de requêtes 99
6.1.1.2 Construction de l'arbre de recouvrement 100
6.1.1.3 Expérimentations 102
6.1.2 Sous forme de résumé 104
6.1.2.1 QSL 106
6.1.2.2 Mesure de qualité d'un résumé 106
6.1.2.3 Résumé d'un log 106
6.1.2.4 Implémentation et tests 106
6.2 Démarrage à froid 108
6.2.1 Obtenir les squelettes des requêtes du log 109
6.2.2 Prédire les opérations candidates 110
6.2.3 Calculer les recommandations candidates 111
6.2.4 Ordonner les recommandations candidates 112
6.3 Ajouter des données/sources externes : Prise en compte du contexte . 113
6.3.1 Contexte 113
6.3.2 Les facteurs de contexte de l'utilisateur 114
6.3.3 Dans la pratique... 117
6.3.3.1 En OLAP 117
6.3.3.2 Dans un environnement commercial 121
6.3.3.3 En gestion de crises 124
6.4 Bilan 129

Nous avons vu dans le chapitre précédent qu'il est possible d'évaluer les systèmes de recommandation ainsi que les recommandations qu'ils retournent. Cependant, malgré de bonnes performances, les systèmes de recommandation et/ou les recommandations retournées peuvent être améliorés pour tenter d'être encore et toujours au plus près des attentes/besoins des utilisateurs. Il est à noter qu'en aucun cas, dans ce chapitre, nous ne nous intéressons à l'optimisation algorithmique liée à une exécution plus rapide ou à une consommation de ressources moins grande du système de recommandation. Au lieu de cela, nous souhaitons améliorer la qualité des recommandations retournées (c'est-à-dire les sorties du système de recommandation).

Pour rappel, notre système générique de recommandation, présenté dans le chapitre 4, est composé de 4 grandes parties (comme illustré par la figure 4.1) :

- le besoin de l'utilisateur,
- les entrées : un ensemble d'utilisateurs, un ensemble d'entités recommandables, une matrice d'utilité et un log/journal liant les entités recommandables et les utilisateurs,
- les trois étapes de recommandation : prétraitement, génération et ordonnancement des recommandations candidates,
- les sorties : les recommandations.

Comme indiqué au début de ce mémoire, il existe une quantité diluvienne de données/informations hétérogènes. Ce volume et cette variété sont, bien entendu, un plus dans l'amélioration des systèmes de recommandation mais ces deux propriétés ne sont rien sans la véracité/qualité des données d'entrées [TDG+16]. Ainsi, une amélioration de la qualité des données d'entrée du système de recommandation serait une étape vers l'amélioration de la recommandation. Notre système générique de recommandation nécessite quatre entrées différentes. De nombreux travaux se sont déjà intéressés à l'amélioration des profils utilisateurs ou des caractéristiques des éléments [CEG12, LZ13, Nad16] ainsi qu'à celle de la matrice d'utilité [KBV09, KPC16]. Mais, à notre connaissance, aucun travail de recherche ne s'est intéressé à l'amélioration des logs/journaux pour une utilisation dans un système de recommandation.

Par ailleurs, le problème/challenge le plus difficile dans les systèmes de recommandation est le démarrage à froid (abordé dans le chapitre 3), c'est-à-dire lorsque le système de recommandation ne peut pas prédire de score d'utilité pour les utilisateurs ou les éléments sur lesquels il n'y a pas encore suffisamment d'information [SPUP02]. Les travaux de [Kło08] ont permis d'identifier plusieurs types de problèmes de démarrage à froid : nouvel utilisateur [RKR08, GKL11], nouvel élément [SPUP01], association non transitive et apprentissage. Le démarrage à froid a fait l'objet de nombreux travaux dans des domaines divers et variés. Cependant, ces travaux se sont limités à un seul type de démarrage à froid [ZTZX14, NB14]. Or, comment gérer le problème lorsque le système de recommandation se retrouve en présence à la fois de nouveaux éléments et de nouveaux utilisateurs. Ainsi, proposer une approche de démarrage à froid capable de pallier une double problématique liée à de nouveaux éléments couplés à de nouveaux utilisateurs serait aussi une étape vers l'amélioration de la recommandation.

Enfin, un certain nombre d'algorithmes ont été proposés pour améliorer à la fois la qualité de la recommandation et les problèmes d'évolutivité (passage à l'échelle) [BHK98, SKKR01, MZLK11]. Cependant, en dépit de leur succès, les méthodes existantes souffrent du problème de faible densité des données. Par exemple, la densité des scores disponibles dans les systèmes de recommandation commerciaux est souvent inférieure à 1% voire moins [SKKR01]. Ainsi, il apparait que l'amélioration de la qualité de la recommandation passe par la résolution du problème de la faible densité des données. Grâce aux applications web et à l'Internet des objets, entre autres, les systèmes de recommandation peuvent être associés à différents types d'informations contextuelles comme la position géographique ou l'agenda de l'utilisateur. Ces informations contextuelles contiennent de nombreuses informations complémentaires sur les intérêts des utilisateurs ou les propriétés des éléments [ADE+13, MHT+17], ce qui leur permettra d'améliorer la qualité de la recommandation [CCST11].

Par conséquent, dans ce chapitre, nous nous intéressons aux recommandations d'éléments, individuelles ou à visée publique, avec amélioration (comme illustré sur les figures 3.5 et 3.4) et, plus particulièrement à trois approches différentes d'amélioration de la recommandation, à savoir : (i) améliorer la qualité des données d'entrée, en particulier des logs/journaux (dans la section 6.1) (ii) gérer le problème du démarrage à froid pour les cas où il y a simultanément de nouveaux utilisateurs et de nouveaux éléments (dans la section 6.2) ou encore (iii) intégrer des données/sources externes (données contextuelles) (dans la section 6.3). Nous présentons nos différentes propositions en ce sens.

6.1 Améliorer les données d'entrée : Structuration de logs/journaux

Dans cette section, nous nous concentrons sur l'amélioration de la qualité des données d'entrée d'un système de recommandation. Comme spécifié précédemment, à notre connaissance, aucun travail ne s'est intéressé à l'amélioration des logs/journaux pour une utilisation par un système de recommandation. Un log/journal est par définition, un fichier, ou tout autre dispositif, permettant de stocker un historique des évènements attachés à un processus[1]. Or, un log/journal peut être mal organisé, c'est-à-dire pas organisé pour le traitement voulu dans le cadre des systèmes de recommandation. En effet, c'est un fichier plat, non structuré, dans lequel il est difficile/fastidieux de trouver des informations. De ce fait, il faut le nettoyer, organiser/ranger/structurer au mieux pour faciliter/améliorer le traitement par le système de recommandation.

Des travaux existent sur l'analyse de logs (et les algorithmes d'analyse) [HP00, HWN+08, FA14] et sur le prétraitement des logs [RL14]. Cependant, l'analyse de logs permet de faire émerger des informations sans pour autant structurer/organiser les logs et les travaux sur le prétraitement se concentrent sur le nettoyage des données du log sans aller jusqu'à la réorganisation [RL14]. Seul [CFT07] va plus loin en proposant une approche de résumé de données.

Les évènements du log ont une relation temporelle et contiennent des informations sur les corrélations/liens entre les entités recommandables et les utilisateurs. Or, en informatique, un moyen de modéliser des données et les « liens » entre ces données est d'utiliser un graphe. Un graphe permettrait de structurer un log, faisant ainsi apparaitre des données/informations complémentaires qui n'étaient pas visibles dans la version non structurée du log. Les données/informations ainsi obtenues viendront renforcer les données d'entrée du système de recommandation.

Par ailleurs, les logs sont des fichiers plats très volumineux dans lesquels il est très difficile voire impossible de se retrouver. Une simple visualisation du contenu est fastidieuse. Avoir une représentation concise, c'est-à-dire un résumé, de ce que contient le log permettrait de mieux appréhender son contenu et de le structurer. Une telle représentation, qui condense les données du log, permettrait également de faire apparaitre des informations complémentaires mais aussi d'améliorer le temps de (pré)traitement du log et de ce fait, les performances du système de recommandation.

Ainsi, nous proposons deux approches de structuration de logs afin d'améliorer les données d'entrée d'un système de recommandation, à savoir une structuration du log soit sous forme de graphe, soit sous la forme d'un résumé. Notons que depuis nos travaux, d'autres chercheurs ont proposé des approches de traitement de logs à base de graphes [XCC14].

1. Généralement, ces évènements sont horodatés et ordonnés en fonction du temps.

6.1.1 Sous forme de graphe

Dans cette section, nous illustrons notre proposition de structuration de logs sous forme de graphes en l'appliquant au domaine des entrepôts de données et en particulier de l'OLAP [CMN10]. Pour rappel, OLAP permet l'exploration et l'analyse de gros volumes de données dans un environnement multi-utilisateurs [DBB07]. Dans ce domaine, les logs sur lesquels s'appuient le système de recommandation contiennent des requêtes d'un serveur OLAP. Organiser/structurer les logs permettrait de mettre rapidement en évidence les analyses déjà conduites, de manière à aider les explorations ultérieures [KBG+09].

Ainsi, nous proposons d'organiser les logs d'un serveur OLAP sous forme de graphes.

Nous adaptons une technique qui a fait ses preuves pour la réorganisation de sites web, celle de [CMS08]. Cette adaptation repose sur une similarité entre requêtes basée sur le TF-IDF. Plus précisément, le log est d'abord vu comme un graphe pondéré de requêtes, les poids reflétant la similarité entre requêtes. Cette similarité est calculée en tenant compte des faits de l'entrepôt interrogés et de l'appartenance des requêtes à une même session. Ces poids sont ensuite ajustés pour trouver un équilibre entre ces deux facteurs. Enfin le log est présenté sous la forme d'un graphe dont la structure est l'arbre couvrant minimal de ce graphe.

En résumé, nos contributions sont les suivantes :

- une méthode pour l'organisation de requêtes OLAP,
- l'implémentation de cette méthode pour organiser des logs,
- des expériences conduites sur des logs synthétiques pour tester la qualité des solutions obtenues, selon des critères utilisés dans le domaine du web et des interfaces homme-machine.

Considérons un log contenant des sessions/séquences de requêtes OLAP. Ce log est le résultat de toutes les sessions conduites par tous les différents utilisateurs pour analyser un cube de données. Supposons maintenant que d'autres utilisateurs aient les besoins suivants :

- rendre compte de ce qui a été fait avec le cube au cours des sessions antérieures,
- trouver une analyse sur un sujet particulier, cette analyse pouvant avoir été conduite sur plusieurs sessions,
- savoir si une analyse particulière n'a pas déjà été conduite,
- savoir si des sessions peuvent être connectées entre elles parce qu'elles portent sur des sujets proches.

Le but de notre approche est de satisfaire ces besoins.

En adaptant les travaux de [CMS08], étant donné un log, notre méthode repose sur les étapes suivantes :

1. Construction du graphe de requêtes,
2. Construction de l'arbre de recouvrement.

Nous illustrons ci-dessous chacune de ces étapes. Avant cela, nous notons que si le log est trop volumineux, sa taille pourra être réduite en effectuant un clustering, comme cela est, par exemple, fait dans [Neg09], et en construisant un graphe à partir uniquement des requêtes représentant les clusters.

6.1.1.1 Construction du graphe de requêtes

A partir du log, un graphe pondéré de requêtes, éventuellement non-connexe, est construit. Plusieurs méthodes de construction sont envisagées selon la prise en compte des sessions et du séquencement des requêtes. L'ensemble des requêtes du log peut être représenté sous la forme d'un graphe où chaque nœud représente une requête et où les arêtes du graphe sont pondérées [2] selon la distance entre les nœuds, c'est-à-dire les requêtes.

Adjacence
Afin de construire le graphe représentatif du log, trois moyens pour relier les nœuds sont possibles :

(a) Non prise en compte des sessions de requêtes
Une première approche consiste à voir le log comme un ensemble de requêtes indépendantes les unes des autres et donc à ne pas prendre en compte le séquencement des requêtes dans les sessions, c'est-à-dire à considérer que toute requête peut être rapprochée de toute autre. Le graphe représentatif du log sera alors complet. Une telle représentation est donnée via le graphe de la Figure 6.1 (a).

(b) Prise en compte des sessions mais pas du séquencement de requêtes
Une seconde approche est de considérer que l'ordre dans lequel les requêtes ont été lancées dans une session n'est pas important. Dans ce cas, le log est vu comme un ensemble de sessions contenant des requêtes non ordonnées, les nœuds représentants des requêtes issues d'une même session sont reliés entre eux. Cela revient à réaliser des graphes complets sur les requêtes d'une même session. Une telle représentation est donnée via le graphe de la Figure 6.1 (b).

(c) Prise en compte du séquencement de requêtes et des sessions
Une dernière approche est de considérer que l'ordre dans lequel les requêtes ont été lancées au sein d'une session est important. Dans ce cas, une succession ou une précédence existe entre deux requêtes dans au moins une session du log et les nœuds représentants les deux requêtes sont liés. Ainsi, si une requête q_i précède ou succède à une requête q_j dans le log, une arête est placée entre les nœuds q_i et q_j dans le graphe. Chaque sous-graphe correspondant à une session est connexe. Une telle représentation est donnée via le graphe de la Figure 6.1 (c).

Selon [Sar00, SBFS00], le séquencement des requêtes est important dans les sessions d'analyse. Par conséquent, l'option (c) devrait être la meilleure. Afin de vérifier cela, dans la suite nous gardons les trois options et les départagerons durant les expérimentations.

Similarité
En utilisant une des trois techniques décrites précédemment, nous disposons d'un graphe représentatif du log où chaque nœud représente une requête. Les arêtes du graphe sont pondérées en fonction de la distance entre les nœuds, c'est-à-dire les requêtes. Nous proposons d'utiliser le TF-IDF utilisé efficacement pour évaluer l'importance d'un mot dans un document. Dans notre cas, une requête OLAP est vue comme un ensemble de membres, nous pouvons donc évaluer l'importance d'un membre dans une requête. Pour chaque requête i, nous calculons un vecteur $tfidf_i$, où chaque composante est notée $tfidf_{i,j}$ avec j un membre de la requête i. La taille du vecteur est fonction de l'ensemble des membres présents dans le log de requêtes.

Finalement, les arêtes du graphe sont pondérées avec la valeur de la distance, obtenue avec le TF-IDF, entre les deux nœuds, c'est-à-dire requêtes, qu'elles relient.

2. En OLAP, une session d'analyse est très souvent constituée de requêtes « proches » [Sar00].

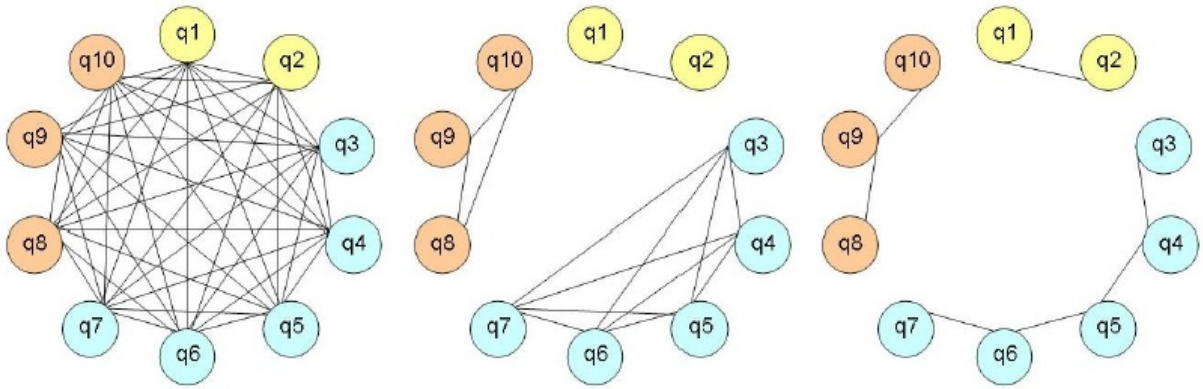

FIGURE 6.1 – Graphes représentatifs du log selon différents types d'adjacence.

Racine
Hormis le graphe représentant le log sans prendre en compte les sessions (section 6.1.1.1.a), les graphes générés, à savoir celui représentant le log en prenant en compte les sessions mais pas le séquencement (section 6.1.1.1.b) et celui prenant en compte le séquencement (section 6.1.1.1.c), ne sont pas forcément connexes. De ce fait, certains sommets (requêtes) ne peuvent pas être atteints à partir des autres sommets. Or, nous voulons représenter le contenu du log de manière exhaustive où chaque requête a un lien plus ou moins « fort » avec les autres requêtes du log. Nous devons donc rendre le graphe connexe (c'est-à-dire obtenir un graphe à n sommets ayant au moins $n-1$ arêtes). Obtenir un graphe connexe n'est pas une tâche aisée, aussi nous nous sommes orientés vers un graphe connexe « simple », à savoir un arbre (c'est-à-dire un graphe à n sommets ayant exactement $n-1$ arêtes). Pour cela, il faut un nœud racine. Nous considérons que le nœud racine est une requête existante dans le log. Notre choix s'est donc porté vers la requête médoïde. Il s'agit de la requête pour laquelle la distance moyenne à toutes les autres requêtes du log est minimale. Si plusieurs requêtes peuvent être le médoïde, c'est la première requête trouvée qui devient la racine de l'arbre. Deux possibilités sont envisagées pour rendre connexe le graphe de départ :

- relier le médoïde à la première requête de chaque session (dans le cas où le séquencement de requêtes est pris en compte (section 6.1.1.1.c)),
- relier le médoïde à la requête la plus proche de chaque session.

Ces deux possibilités seront départagées par la suite à partir des expérimentations.

La figure 6.2 propose une modification du graphe de la figure 6.1 (c) où le médoïde $q1$ est relié à la première requête de chaque session.

6.1.1.2 Construction de l'arbre de recouvrement

Une heuristique à base de fourmis artificielles est alors utilisée pour ajuster ces poids afin de rapprocher les requêtes [CMS08]. Parce que la proximité entre requêtes est importante en OLAP [Sar00], nous souhaitons construire un arbre en minimisant son coût (représenté par les poids sur les arêtes). En théorie des graphes, un tel arbre est obtenu en calculant l'arbre de recouvrement minimal. La racine de cet arbre est la requête en moyenne la plus proche des autres requêtes du log.

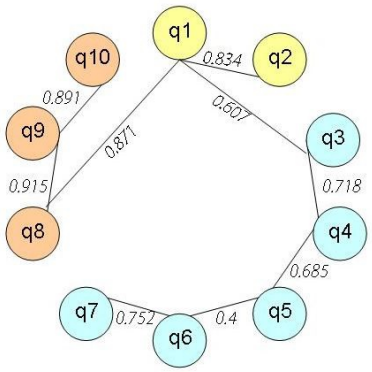

FIGURE 6.2 – Graphe pondéré représentatif du log prenant en compte le séquencement et les sessions où le médoïde est relié à la première requête de chaque session.

Prim

A partir du graphe du log, le problème consistant à obtenir un arbre couvrant de poids minimum (AM) peut être résolu par l'algorithme de Prim [Pri57]. Cet algorithme consiste à faire croître un arbre à partir d'un nœud racine (le premier sommet marqué). Dès lors, à chaque itération, il connecte un sommet non marqué à un sommet marqué (c'est-à-dire déjà placé dans l'arbre en construction). L'arbre va ainsi grossir jusqu'à ce qu'il couvre tous les sommets du graphe. A chaque étape, le nouveau sommet y à marquer est tel que y est adjacent à un sommet x marqué et que l'arête (x, y) est celle de plus faible poids. Chaque augmentation se fait donc de la manière la plus économique possible. Cependant, avec l'algorithme de Prim, seuls les liens du graphe serviront dans la construction de l'AM. Ainsi pour les graphes prenant en compte les sessions (sections 6.1.1.1.b et 6.1.1.1.c), aucun rapprochement entre requêtes de sessions différentes ne sera établi par le simple fait d'un coût économique. Afin de permettre l'établissement de nouveaux liens non initialement présents mais permettant l'obtention d'un AM de coût inférieur, nous allons ajouter une pondération supplémentaire à chaque chemin permettant de représenter la probabilité d'utilisation pour la construction de l'AM. Pour ce faire nous avons utilisé un algorithme de fourmis artificielles [CMS08].

Ajustement des chemins par les fourmis artificielles

Dans leur milieu naturel les fourmis sont inconsciemment capables de trouver le plus court chemin entre le nid et une source de nourriture. Pour cela, elles déposent des phéromones (substances volatiles) sur le terrain, ce qui incite les autres fourmis à suivre le même chemin (et renforcent à leur tour le chemin). L'heuristique ACO (Ant Colony Optimization) regroupe toutes les variantes issues de cette rencontre fourmis-phéromones-optimisation. Le principe général de l'algorithme ACO utilisé pour la génération de plan par les fourmis artificielles est donné par l'algorithme 5 [CMS08].

Dans cet algorithme, les fourmis réalisent des circuits, représentant des regroupements de sources de nourriture (c'est-à-dire les requêtes) proches les unes des autres, sur lesquels l'algorithme de Prim est utilisé. Le nombre de fourmis dans cet algorithme est égal au nombre de requêtes. La fourmi doit opérer des choix concernant le chemin à emprunter, choix influencés par la distance séparant les sources de nourriture (la distance entre requêtes) et par la quantité de phéromones sur les chemins. Dans le but de favoriser les liens mis en evidence dans le graphe de départ, la matrice de phéromones est initialisée avec la matrice d'adjacence du graphe initial, et, pour ne pas

exclure les chemins inexistants au départ, une quantité faible mais non nulle de phéromones leur est attribué. Lors de la mise à jour des phéromones, la quantité de phéromones augmente pour les liens appartenant à la meilleure solution et diminue dans le cas contraire. Les fourmis étant attirées vers les chemins contenant la plus grande quantité de phéromones, un écart se creuse au niveau de la quantité de phéromones sur les liens, selon qu'ils appartiennent à la solution ou non. Le test d'arrêt est vérifié lorsque la quantité de phéromones sur les chemins appartenant à la meilleure solution est classiquement supérieure à 0.9, et inférieure à 0.1 pour ceux n'appartenant pas à la meilleure solution.

Algorithme 5 Génération de plan avec des fourmis artificielles

```
Initialiser les valeurs des phéromones
Répéter
  Pour tout les fourmis Faire
    Générer les circuits
    Pour chaque circuit Faire
      Générer un AM avec l'algorithme de Prim
    Fin Pour
  Fin Pour
  Mettre à jour les phéromones
Jusqu'à validation du test d'arrêt
Retourne la meilleure solution trouvée depuis le début
```

6.1.1.3 Expérimentations

Nous avons évalué les résultats produits. Les logs utilisés sont des données synthétiques produites grâce à notre propre générateur [Neg09].

Les tests mis en place permettent d'extraire des indicateurs concernant les coûts (coût de l'arbre, c'est-à-dire la somme des poids des arêtes de l'arbre, coût des liens ajoutés par la méthode des fourmis par rapport au graphe de requêtes), et la forme de l'arbre obtenu (hauteur, largeur, séparation des sous-arbres).

Le graphe G1 (Figure 6.1 (a)) correspond au graphe ne prenant pas en compte le séquencement et les sessions ; G2 et G3 (Figure 6.1 (c)) correspondent aux graphes prenant en compte le séquencement et les sessions avec le médoïde relié respectivement aux premières requêtes de chaque session (G2 - Figure 6.2) et à la plus proche requête de chaque session (G3) ; et G4 (Figure 6.1 (b)) correspond au graphe prenant en compte les sessions mais pas le séquencement (avec le médoïde relié à la plus proche requête de chaque session). Il apparaît que le graphe G1 donne de moins bons résultats en terme de séparation de sous-arbres que les graphes prenant en compte les sessions. Les meilleurs résultats sont obtenus avec le graphe G2 en utilisant l'approche à base de fourmis artificielles et avec le graphe G3 en utilisant l'algorithme Prim. Nous observons en général une meilleure séparation des sous-arbres avec la méthode des fourmis artificielles. En moyenne, nous observons une meilleure séparation avec la méthode des fourmis par rapport à l'algorithme de Prim pour les logs de faible densité, et une distance plus faible pour les logs de forte densité (requêtes en moyenne plus proches les unes des autres).

Nous observons que la quantité moyenne de phéromones sur tous les chemins diminue au fur et à mesure des itérations. Cette évaporation moyenne des phéromones illustre la suppression d'arêtes entre le graphe de requêtes et l'arbre de recouvrement, mais aussi l'échange entre une arête existante et une arête de plus faible coût choisie par les fourmis.

La valeur moyenne du coût des solutions décroissante montre qu'au fur et à mesure des itérations

les fourmis sont incitées à choisir des chemins (grâce aux phéromones) qui mèneront à une solution de faible coût. Notons que le coût de l'arbre est moindre avec un graphe de requêtes prenant en compte les sessions.

Finalement, les fourmis artificielles sont capables de choisir les arêtes qui permettront d'obtenir une solution de coût réduit. Elles ont bien pondéré les chemins apportant une amélioration à la solution finale, puisque les coûts des arbres se trouvent inférieurs à ceux des arbres générés uniquement par Prim. De plus, elles apportent une amélioration en terme de séparation des sous-arbres pour les logs de faible densité.

Au vu des résultats obtenus, le graphe ayant les meilleures performances est G2 [3] (avec l'approche à base de fourmis artificielles). Finalement, pour obtenir la meilleure (en terme de performances) représentation/structuration d'un log sous forme de graphe dans le domaine OLAP afin d'améliorer les données d'entrée d'un système de recommandation, nous préconisons de prendre en compte le séquencement des requêtes et les sessions de requêtes, ainsi que de relier le médoïde (des requêtes) aux premières requêtes de chaque session.

Il est à noter que, outre son intérêt pour les systèmes de recommandation, notre proposition pourrait permettre à un utilisateur d'appréhender le contenu d'un log de requêtes. Cela lui permettrait d'avoir une idée de ce qui a été fait avec le cube au cours des sessions antérieures, de trouver une analyse sur un sujet particulier, ... Ainsi, afin que notre proposition soit conviviale pour l'utilisateur, nous avons proposé une visualisation de l'arbre obtenu sous forme de site web. L'arbre de recouvrement donne la structure du site, à savoir un ensemble de pages reliées entre elles à raison d'une requête par page (nœuds de l'arbre). Les hyperliens (arêtes) entre requêtes adjacentes q et q' sont ajoutés de manière à représenter les membres à ajouter et à supprimer pour passer de la requête q à la requête q'. A chaque nœud est également associé une description du sous-arbre dont ce nœud est la racine contenant la liste des membres apparaissant dans le sous-arbre.

Ainsi, le log est présenté comme un site web de requêtes dont la figure 6.4 présente la structure et où la figure 6.3 montre la page web correspondant à la requête q_1. Ainsi, l'utilisateur peut naviguer ce site et voir par exemple que :

- suivre un des chemins correspond à une analyse des ventes en Californie via les requêtes q_2 ou q_6,
- une analyse portant sur l'année 1998 n'a pas déjà été conduite, par exemple, la suite de requêtes q_3, q_9, q_{10} qui traite des ventes de boissons en fonction du genre en 1997, pourrait être réutilisée pour l'année 1998.

Synthèse Un log est un fichier plat, non structuré, dans lequel il est difficile de trouver des informations. De plus, il n'est pas organisé pour le traitement voulu dans le cadre des systèmes de recommandation. Il est donc nécessaire de le structurer au mieux pour faciliter son traitement. Parce qu'il existe des corrélations entre les entités recommandables et les utilisateurs, nous proposons une technique pour transformer un log en un graphe. Elle repose sur deux étapes : (i) construction du graphe de requêtes, et (ii) construction de l'arbre de recouvrement. Cette technique est implémentée et a été testée en OLAP. La meilleure solution consiste à construire un graphe de requêtes prenant en compte le séquencement et les sessions avec le médoïde relié aux premières requêtes de chaque session, couplée avec l'algorithme à base de fourmis artificielles. Par souci de convivialité, la meilleure

3. Le graphe G2 prend en compte le séquencement et les sessions et le médoïde est relié aux premières requêtes de chaque session (Figure 6.2).

FIGURE 6.3 – Exemple de page web pour la requête q_1

solution obtenue pourrait être affichée sous forme de site web à un utilisateur. Un tel outil permettra d'améliorer la structuration du log/journal utilisé par le système de recommandation faisant ainsi apparaitre des données/informations complémentaires qui viendront renforcer les données d'entrée du système de recommandation. Cet outil pourra également aider l'utilisateur dans la consultation d'un log.

6.1.2 Sous forme de résumé

Dans cette section, nous illustrons notre proposition de structuration de logs sous forme de résumés. Ainsi structurées, les informations contenues dans le log seront plus facilement exploitables par le système de recommandation. En effet, une représentation condensée du contenu du log facilitera le traitement du log (et donc améliorera son temps de traitement), fera apparaitre les informations essentielles du log, voire des informations nouvelles et permettra ainsi d'améliorer la recommandation. Nous appliquons notre proposition au domaine des entrepôts de données et en particulier de l'OLAP [AMN10a, AMN10b, AMN11a, AMN11b] où les logs sur lesquels s'appuient le système de recommandation contiennent des requêtes d'un serveur OLAP. En OLAP, où l'accès à l'information se fait par l'intermédiaire de requêtes, un résumé permettrait un gain de temps et d'efficacité dans l'accès à l'information. Par exemple, il aurait comme avantage de donner une première idée du genre de requêtes lancées par d'autres utilisateurs [KBG+09].

Dans [AMN10a, AMN11a, AMN11b], nous développons nos travaux sur le résumé de log de requêtes OLAP, où nous proposons un langage de manipulation de requêtes (appelé QSL) pour

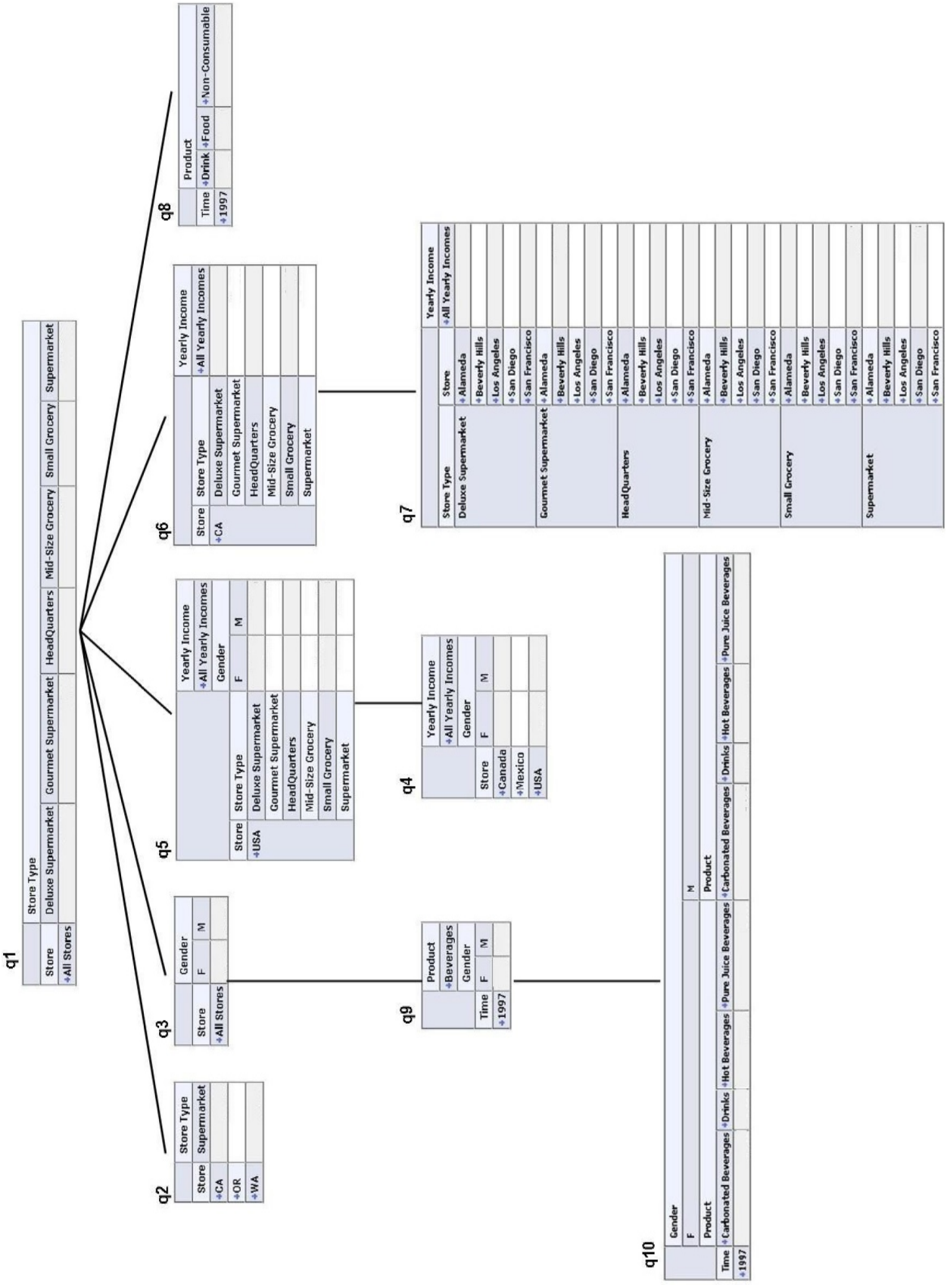

FIGURE 6.4 – Arbre couvrant de poids minimal fournissant la structure du site

résumer des requêtes OLAP, une mesure de la qualité d'un résumé et un algorithme glouton de construction automatique de résumés de log de requêtes utilisant QSL.

6.1.2.1 QSL

QSL est un langage algébrique de manipulation de requêtes à des fins de résumer. Il est composé d'opérateurs unaires et binaires qui manipulent des requêtes et retournent une requête, appelée la requête résumée (ou résumé par abus). Ces opérateurs construisent une nouvelle requête à partir de celle(s) en paramètres en traitant chaque dimension indépendamment les unes des autres.

6.1.2.2 Mesure de qualité d'un résumé

Nous cherchons à évaluer si une requête (resp., un log), qui correspond à un ensemble de références [4] (resp., requêtes), résume fidèlement une autre requête (resp., un autre log). Les opérateurs de QSL résument en conservant, ajoutant ou retirant des références aux requêtes à résumer pour constituer le résumé. Une manière d'évaluer cette fidélité est de mesurer la proportion de ce qui est ajouté ou enlevé par l'opérateur. Nous nous sommes donc tournés vers les mesures de rappel et précision. Dans notre cas, ces mesures sont étendues pour prendre en compte la relation de couverture exploitée par les opérateurs. Pour être le plus général possible, nous choisissons la relation de couverture définie sur les références puisque tant les requêtes que les logs peuvent être assimilés à des ensembles de références [5]. Ainsi définie, cette mesure présente l'intérêt de pouvoir être appliquée sur deux requêtes ou deux logs, ou n'importe quel couple d'ensembles de références.

6.1.2.3 Résumé d'un log

Notre objectif est de résumer un log (qui est un ensemble de requêtes et donc un ensemble de références). L'algorithme (6) *SummarizeLog* est un algorithme glouton qui résume un log en résumant successivement des requêtes au moyen des opérateurs de QSL, jusqu'à atteindre une taille donnée [6]. L'expression QSL employée est celle qui maximise la mesure de qualité [7] tout en faisant varier le log. Le choix de l'expression utilise deux stratégies. La première teste pour chaque requête ou couple de requêtes consécutives quelle opération de QSL maximise la qualité, l'expression QSL est alors l'application de cette opération (stratégie 1). La seconde teste pour chaque couple de requêtes consécutives quelle opération binaire de QSL maximise la qualité, applique cette opération, puis teste sur le résultat quel opérateur unaire maximise la qualité. Un résumé est donc également un log.

6.1.2.4 Implémentation et tests

L'implémentation a été faite en considérant que les dimensions concernées résident en mémoire. Celles-ci sont représentées par des arbres stockant pour chaque membre la cardinalité de sa couverture au niveau le plus fin. Les logs utilisés proviennent de logs synthétiques sur la base de données

4. Une référence de cellule d'un cube de données C à N dimensions : $C = \langle D_1, ..., D_N, Fait \rangle$ (ou référence par abus) est un N-uplet $\langle r_1, ..., r_N \rangle$ où $r_i \in adom(D_i)$ (domaine actif : ensemble des membres de la dimension D_i) pour tout $i \in [1, N]$.

5. Une requête est un ensemble de références par définition. Le log, en tant qu'ensemble de requêtes (qui sont des ensembles de références) peut aussi être assimilé à un ensemble de références.

6. La f-measure est la moyenne harmonique du rappel et de la précision telle que $f-measure = 2 * \frac{Précision \times Rappel}{Précision + Rappel}$

7. La qualité testée à chaque itération de *SummarizeLog* est une qualité « locale » entre la ou les requêtes intervenant dans l'expression QSL et le résultat de l'expression.

Algorithme 6 SummarizeLog (stratégie 1)

Entrées : Log :un log
U : un ensemble d'opérateurs unaires
B : un ensemble d'opérateurs binaires
D : un ensemble de dimensions
α : un entier positif
Sorties : un résumé de Log
Variables : une requête q

Tant que $|Log| > \alpha$ **Faire**
$q = argmax_1(\{f - measure(q', q'', D) \mid q'' = op(q'), op \in U, q' \in Log\}$
$\cup \{\{f - measure(q' \cup q'', q''', D) \mid q''' = op(q', q''), op \in B, q', q'' \in Log\}\})$
Remplacer dans Log l'ensemble des requêtes impliquées dans l'expression QSL de q par q
Fin Tant que
Retourne Log

exemple *FoodMart* fournie avec le moteur OLAP Mondrian [Pen17]. La génération des logs, simulant une analyse OLAP, est expliquée dans [Neg09]. Les logs générés ont une densité assez élevée (5 dimensions parmi les 13 possibles sont utilisées pour naviguer). Ils sont répartis en 4 fichiers de respectivement 119, 241, 437 et 904 requêtes.

Nous avons testé l'efficacité de l'algorithme *SummarizeLog* dans ses deux stratégies[8] présentées précédemment. Le temps mis par *SummarizeLog* évolue de manière polynomiale avec la taille du log à résumer et s'avère assez couteux pour des logs importants. Ce temps reste néanmoins acceptable puisqu'il suffit de 10 minutes environ avec la deuxième stratégie pour réduire le plus gros fichier log de 80%. De manière générale, la stratégie 2 reste plus efficace, demandant moins de tests de qualité (partie la plus couteuse de l'algorithme *SummarizeLog*). Le nombre de requêtes inintéressantes augmente tout d'abord, parce qu'elles bénéficient d'un bon « rappel », jusqu'à ce que leur présence dégrade trop la « précision », elles sont alors « absorbées » par les autres requêtes. Nous constatons finalement que la stratégie 1 donne de meilleurs résultats en terme de qualité globale.

Synthèse Un log peut être très volumineux. Bien que le volume de données à disposition soit intéressant dans le cadre des systèmes de recommandation (notamment dans les approches collaboratives), la qualité des données d'entrée, quant à elle, est primordiale. L'amélioration de la qualité du log, de notre point de vue, serait une étape vers l'amélioration des recommandations. A cet effet, dans le domaine de l'OLAP, nous proposons de résumer un log de requêtes. Le format du résumé est le même que celui des données résumées : une requête résume une autre requête et un log résume un autre log. Nos contributions s'appuient sur un langage de résumé de requêtes qui permet de spécifier un résumé, une mesure pour évaluer la qualité d'un résumé et un algorithme pour calculer automatiquement un résumé de log de requêtes de bonne qualité. Notre approche a été testée pour résumer des logs de requêtes OLAP et les résultats obtenus sont prometteurs. Elle permettra ainsi d'améliorer la recommandation.

8. Comme indiqué dans la section 6.1.2.3, la stratégie 1 consiste à tester pour chaque requête ou couple de requêtes consécutives quelle opération de QSL maximise la qualité, l'expression QSL est alors l'application de cette opération. La stratégie 2 consiste à tester pour chaque couple de requêtes consécutives quelle opération binaire de QSL maximise la qualité, appliquer cette opération, puis tester sur le résultat quel opérateur unaire maximise la qualité.

6.2 Démarrage à froid

Dans cette section, nous nous concentrons sur le problème du démarrage à froid des systèmes de recommandation. Comme spécifié précédemment, bien que de nombreux travaux se soient intéressés à ce problème, ils se sont limités à un seul type de démarrage à froid (à la fois). Mais, à notre connaissance, aucun travail ne s'est intéressé au démarrage à froid lorsque le système de recommandation cumule de nouveaux éléments et de nouveaux utilisateurs.

Ainsi, nous souhaitons proposer une approche de démarrage à froid capable de pallier cette double nouveauté afin d'améliorer la recommandation. Nous illustrons cette proposition en l'appliquant au domaine des entrepôts de données et OLAP [NRTT13a, NRTT13b].

Bien que les approches de recommandation existantes dans le domaine OLAP permettent de recommander des requêtes pertinentes à un utilisateur donné, elles s'avèrent inopérantes lorsque l'entrepôt de données est mis à jour ou lorsque les décideurs sont différents ou nouveaux. Ce problème du démarrage à froid est rencontré lorsque les recommandations sont nécessaires pour des objets et/ou des utilisateurs pour lesquels nous n'avons aucune information explicite ou implicite [SPUP01]. Dans le cadre des entrepôts de données, nous sommes confrontés au problème du démarrage à froid lorsque le système souhaite formuler des recommandations pour un nouveau cube de données (à un nouvel utilisateur). Par exemple, prenons le cas d'un entrepôt de données contenant (1) des données jusqu'à l'année 2010, (2) un cube de données C1 portant sur les années 2009 et 2010 et (3) le journal des requêtes permettant de construire C1. Supposons que cet entrepôt de données a été mis à jour pour intégrer les données de 2011 et qu'un nouveau cube C2 portant sur les années 2010 et 2011 a été créé, le système ne pourra faire que très peu de recommandations aux décideurs sur C2 voire aucune.

Dans le cadre des entrepôts de données, le problème du démarrage à froid est plus complexe que dans le contexte du web car nous sommes confrontés à un nouveau système : nouvel élément (un nouveau cube) et nouveaux utilisateurs (nouveaux décideurs ou décideurs connus avec une nouvelle fonction). Par conséquent, nous devons aller au-delà de la simple recommandation en suggérant des requêtes à un décideur qui n'ont pas encore été exécutées sur le cube considéré.

Notre processus de démarrage à froid dans le cadre de notre système de recommandation de requêtes OLAP, [NRTT13a, NRTT13b], utilise la séquence de requêtes de la session courante ainsi que le log de requêtes du serveur OLAP. Ce log contient les séquences de requêtes précédemment lancées sur un autre cube qui est similaire au cube sur lequel sont lancées les requêtes de la session courante. Peu de travaux se sont intéressés à la similarité entre cubes. Cependant, nous proposons d'utiliser les travaux de [SSB10] sur l'interopérabilité des cubes afin de détecter des cubes similaires (qui sont les plus aboutis et les plus proches de notre problématique).

Notre processus (voir la figure 6.5 où les quatre parties colorées correspondent à celles de la figure 4.1) inclut les quatre étapes suivantes :

1. Obtenir les squelettes[9] des requêtes du log,
2. Prédire les opérations candidates,

9. La probabilité de trouver deux requêtes syntaxiquement identiques dans un log est très faible. Par contre, celle de trouver des requêtes sémantiquement proches est beaucoup plus grande. Prétraiter les requêtes sous forme de squelettes, contenant les principales informations des requêtes (mesures et niveaux hiérarchiques interrogés) permet de se libérer des considérations syntaxiques.

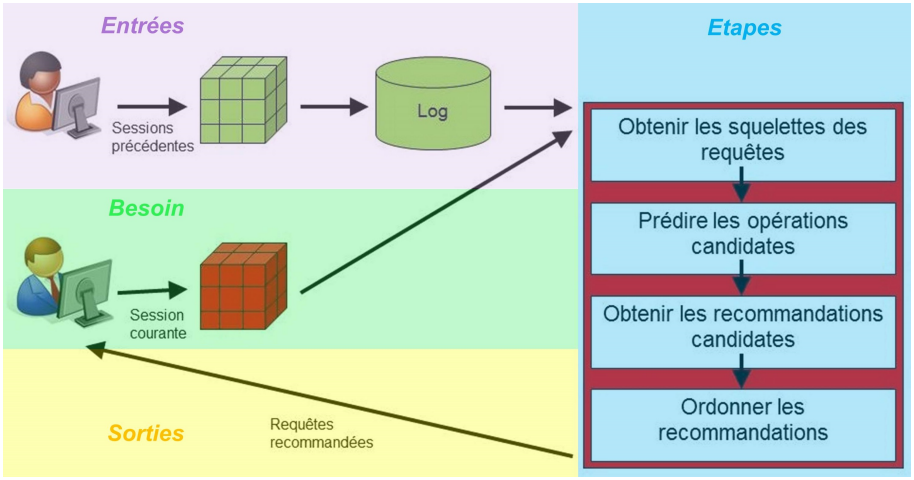

FIGURE 6.5 – Vue d'ensemble du processus de démarrage à froid.

3. Calculer les recommandations candidates en combinant une requête de la session courante et les opérations candidates,
4. Ordonner les recommandations candidates.

Chaque étape est détaillée par la suite.

Il est à noter que, la nouveauté, par rapport à notre approche générique de recommandation (Figure 4.1) ou son instanciation en OLAP (Figure 4.2), réside dans la partie *Étapes* (en bleu), et notamment dans les étapes de prétraitement du log (obtenir les squelettes) et de génération des recommandations candidates.

6.2.1 Obtenir les squelettes des requêtes du log

Parce que le log de requêtes à la disposition du système de recommandation contient des requêtes qui n'ont pas été lancées sur le cube de la session courante, la probabilité de trouver deux requêtes syntaxiquement identiques dans un log est quasiment nulle. Par contre, celle de trouver des requêtes sémantiquement proches est beaucoup plus grande. Pour pallier ce manque de similarité entre requêtes, il est nécessaire de prétraiter le log. Pour cela, nous souhaitons extraire le squelette de chaque requête contenant la structure, les mesures interrogées ainsi que les niveaux hiérarchiques (sans les valeurs). Ainsi, grâce à une fonction permettant d'obtenir le squelette de chaque requête (voir l'algorithme 7), chaque session du log est transformée en squelettes de requêtes. Cette fonction retourne un ensemble d'ensembles de squelettes de requêtes, que nous appelons un ensemble de squelettes de sessions. Le principe est le suivant : Pour chaque session du log, il s'agit de remplacer dans la session, chaque requête par son squelette en utilisant la fonction $extractReqSquel$ et de détailler les opérations qui permettent de passer d'une requête à sa suivante dans la session (fonction $extractOp$). Le squelette de la session courante est obtenu de la même manière.

L'obtention des squelettes de requêtes ($extractReqSquel(q, C)$) consiste à extraire les références (ensemble des valeurs des attributs et des mesures) d'une requête donnée q (lancée sur le cube C) et de retourner, pour chaque dimension, le nom du niveau correspondant.

L'extraction des opérations ($extractOp(q_1, q_2, C)$) consiste à extraire les références de deux requêtes données q_1 et q_2 (lancées sur le cube C), les numéros de niveaux hiérarchiques correspondants

Algorithme 7 LogSquel(L, C_L)

Entrées :
L : Le log de requêtes précédemment lancées, c'est-à-dire un ensemble de sessions,
C_L : Le cube sur lequel les requêtes du log ont été lancées.
Sorties : un ensemble de squelettes de sessions (SP)

$V \leftarrow \emptyset$ // *pour les opérations*
$E \leftarrow \emptyset$ // *pour les squelettes de requêtes*
$SP \leftarrow \langle E, V \rangle$
Pour chaque session $s_i \in L$ **Faire**
  Pour chaque couple de requêtes $\langle q_j, q_{j+1} \rangle \in s_i$ **Faire**
    $SP.E \leftarrow SP.E \cup extractReqSquel(q_j, C_L)$
    // *extractReqSquel : une fonction qui extrait le squelette d'une requête donnée.*
    $SP.E \leftarrow SP.E \cup extractReqSquel(q_{j+1}, C_L)$
    $SP.V \leftarrow SP.V \cup extractOp(q_j, q_{j+1}, C_L)$
    // *extractOp : Une fonction qui extrait l'opération permettant de passer d'une requête à l'autre dans une session donnée.*
  Fin Pour
Fin Pour
Retourne SP

sur chaque dimension et à trouver l'opération de manipulation OLAP qui permet de naviguer d'une requête à l'autre. Nous nous intéressons aux opérations de manipulation OLAP classiques : *slice-and-dice*, *drill-down*, *roll-up*, *switch* et *pivot*[10] [Tho02]. Il est à noter que les opérations *slice-and-dice* et *switch* n'ont pas d'influence sur notre approche puisque ces opérations impactent uniquement les valeurs. Notre approche intervient à un niveau plus élevé, elle manipule la structure du cube et non ses valeurs.

Plus précisément, pour deux requêtes données (q_1, q_2) et pour chaque dimension du cube C :
- si les références (ensembles d'attributs) de chaque requête sont localisées sur le même niveau hiérarchique de la dimension, l'opération pour passer d'une requête à l'autre est *l'identité*,
- si les références d'une seule des requêtes sont localisées au niveau le plus élevé de la hiérarchie (niveau $'ALL'$), l'opération est un *pivot*,
- si les références de la première requête lancée sont localisées à un niveau plus élevé que les références de la seconde requête lancée, l'opération est un forage vers le bas, $DrillDown$,
- dans tous les autres cas, l'opération est un forage vers le haut, $RollUp$.

Lorsque le système détecte un *pivot*, il s'agit de déterminer sur quelle dimension le *pivot* a lieu.

6.2.2 Prédire les opérations candidates

L'étape précédente produit l'ensemble des squelettes des sessions. Maintenant, en utilisant cet ensemble et la séquence de requêtes de la session courante, l'algorithme 8 permet de construire un ensemble d'opérations (de manipulation OLAP) candidates : en plus de l'ensemble des squelettes des sessions et de la session courante, cet algorithme utilise une fonction de concordance (*matching* :

10. http ://www.olapcouncil.org/research/glossaryly.htm

Match). Cette fonction est utilisée pour trouver l'ensemble des squelettes des sessions du log qui coïncident avec le squelette de la session courante. L'algorithme procède de la manière suivante : Dans un premier temps, le squelette de la session courante est construit (la fonction LogSquel est utilisée sur la session courante qui peut être vue comme un log contenant une seule session). Puis, la fonction Match [11] est utilisée pour chercher, parmi l'ensemble des squelettes de sessions, celui qui coïncide avec le squelette de la session courante. Cette fonction retourne un ensemble de paires de sessions indiquant quels squelettes de sessions coïncident avec le squelette de la session courante ainsi que la position de la concordance. A partir de ces paires, l'algorithme extrait un ensemble d'opérations candidates qui sont les opérations qui permettent de passer du squelette de requête, situé à la position retournée dans le squelette de session qui coïncide, au squelette de requête suivant dans le squelette de la session candidate. Cet ensemble d'opérations est retourné comme résultat. Il va servir à construire les recommandations candidates dans l'étape suivante. Il faut noter que notre objectif est d'éviter que cet ensemble soit vide et qu'un tel algorithme a une complexité de $O(n[(w+1)^2+1]+w)$ où n est le nombre de sessions du log et w est la longueur maximale (nombre de requêtes) d'une session donnée.

Algorithme 8 PredictOpCandidates(s_c, C, SP, Match)

Entrées :
s_c : La session courante,
C : Le cube sur lequel la session s_c a été lancée,
SP : l'ensemble des squelettes de sessions, c'est-à-dire le résultat de l'étape précédente,
Match : Une fonction qui retourne les squelettes de sessions candidates.
Sorties : un ensemble d'opérations candidates ($Cand$)

$M, Cand \leftarrow \emptyset$
$P_{s_c} \leftarrow$ le squelette de la session courante s_c
$M \leftarrow Match(P_{s_c}, SP)$
Si $M \neq \emptyset$ **Alors**
  Pour chaque $m = \langle p, pos \rangle \in M$ **Faire**
    //où $p = \langle e, v \rangle$ est un squelette de session
    $Cand \leftarrow Cand \cup p.v(p.e[pos], p.e[pos+1])$
  Fin Pour
Fin Si
Retourne $Cand$

6.2.3 Calculer les recommandations candidates

L'étape précédente produit un ensemble d'opérations (de manipulation OLAP) candidates. Suite à cela, pour chaque opération candidate, une nouvelle requête est construite en appliquant ces opérations candidates à la dernière requête de la session courante.

Dans [Neg09], cinq possibilités ont été considérées sans être départagées : (i) la dernière requête de la session courante, (ii) le successeur d'une requête donnée de la session courante, (iii) l'union des requêtes de la session courante, (iv) l'intersection des requêtes de la session courante, et (v) le médoïde [12] des requêtes de la session courante. La possibilité (v) est chronophage lorsque les sessions

11. La fonction *Match* a été étudiée dans [Neg09] et présentée dans le chapitre 4. Ici, nous utilisons EdH (distance de Levenshtein combinée avec la distance de Hamming) car dans le cas du démarrage à froid, le système proposé doit être le plus rapide et le plus efficace possible.

12. Le médoïde d'un ensemble de requêtes appartient à cet ensemble et est la requête qui minimise les distances

sont longues (à cause du calcul du médoïde pour un grand nombre de requêtes). La possibilité (iv) peut aboutir à un ensemble vide de requêtes et la possibilité (iii) peut créer une requête dont le résultat peut correspondre à l'intégralité du cube de données. Par analogie avec le web [WBC07] et vis-à-vis de notre contexte de démarrage à froid, nous supposons que des étapes intermédiaires sont inutiles et qu'une session d'analyse se concentre sur le but de la session. Ainsi, pour nous, la meilleure possibilité est d'appliquer les opérations (de manipulation OLAP candidates) à la dernière requête de la session courante afin d'obtenir les recommandations candidates.

6.2.4 Ordonner les recommandations candidates

A partir de l'ensemble de recommandations obtenu précédemment, l'idée est de sélectionner les recommandations les plus pertinentes. Dans notre contexte de démarrage à froid, il n'y a pas de critère de satisfaction exprimé par l'utilisateur. Par conséquent, un ordonnancement de requêtes, qui ordonne les recommandations candidates, est difficile à proposer.

A l'heure actuelle, les solutions que nous pouvons considérées sont : (i) ordonner les candidats en fonction de leur proximité avec la dernière requête de la session courante, (ii) ordonner les candidats en fonction de leur nombre d'occurrences dans les logs, (iii) ordonner les candidats en fonction de la position de l'opération candidate dans le squelette du log, c'est-à-dire les opérations les plus récentes lancées dans le log seront utilisées pour construire les premières requêtes de l'ensemble des requêtes recommandées.

Tout d'abord, nous ne connaissons ni la densité du log ni son contenu. Ainsi, ordonner les candidats en fonction de leur nombre d'occurrences dans les logs peut être difficile, si, par exemple, chaque candidat apparait le même nombre de fois. Toujours pour des raisons de rapidité, ordonner les candidats en fonction de leur proximité avec la dernière requête de la session courante peut être chronophage. Pour ces raisons, les candidats sont donc ordonnés en fonction de la position de l'opération candidate dans le squelette du log.

Synthèse Un challenge bien connu des systèmes de recommandation est le démarrage à froid. Il en existe plusieurs types. Souvent traités séparément, il arrive que le système de recommandation se retrouve confronté à deux problématiques en même temps : nouvel utilisateur et nouvel élément. Afin de relever ce double défi, nous avons proposé un système de recommandation lors de l'exploration de nouveaux cubes de données OLAP. Notre processus de prédiction de requêtes est basé sur l'analyse du comportement du décideur durant ses sessions d'analyse (séquences de requêtes) actuelles, et, sur le comportement de l'utilisateur au cours de sessions d'analyses passées sur un autre système.

Cette approche prédictive répond à quatre problématiques : (1) extraire le comportement des utilisateurs en définissant le squelette de requêtes en cours et en analysant les journaux de requêtes sur d'anciens cubes de données via notre algorithme *LogSquel* (une originalité par rapport à un système de recommandation standard/classique), (2) prévoir des opérations candidates de manipulation OLAP via la correspondance de squelettes de requêtes avec notre algorithme *PredictOpCandidates*, (3) sélectionner les opérations pertinentes parmi l'ensemble précédemment défini et construire les recommandations candidates en combinant la dernière requête de la session courante et les opérations candidates (une autre originalité par rapport à un système de recommandation standard/classique), (4) classer ces recommandations candidates en fonction de la position de l'opération candidate dans le squelette du log. Ce système de recommandation, basé sur le principe du démarrage à froid, sera utilisé jusqu'à ce que le système recueille suffisamment de données sur les analyses décisionnelles des utilisateurs pour ensuite pouvoir utiliser un système de recommandation standard.

avec les autres requêtes de l'ensemble.

6.3 Ajouter des données/sources externes : Prise en compte du contexte

Dans cette section, nous nous concentrons sur l'intégration de données/sources externes dans le système de recommandation, et en particulier, d'informations contextuelles [CFCG10, CCST11]. Comme spécifié précédemment, les informations contextuelles contiennent de nombreuses informations complémentaires sur l'utilisateur et les éléments, ce qui permettra d'améliorer la qualité de la recommandation.

6.3.1 Contexte

Malgré de bonnes performances des systèmes de recommandation, les recommandations ne sont parfois « pas assez pertinentes ». Par exemple, supposons que recommander un film à un jeune homme de 20 ans consiste à lui proposer des films de guerre ou d'action et supposons que lorsqu'il est avec une amie, il préfére qu'on lui recommande des films romantiques. Ici le contexte/la situation [13] influence les préférences, les envies et les intérêts des utilisateurs et de ce fait, leurs décisions. Nous souhaitons donc intégrer les données/informations (qu'elles soient qualitatives et/ou quantitatives) contextuelles au système de recommandation et proposer un système de recommandation contextuel.

La notion de « contexte » a été étudiée dans divers domaines mais il est difficile d'établir une définition standard (unique) en raison de la nature multiforme du contexte [BB05]. La définition la plus acceptée est celle de [DSA01] qui définit le contexte comme toute information qui peut être utilisée pour caractériser la situation d'une entité. Une entité peut être une personne, un lieu ou un objet qui est considéré pertinent dans l'interaction entre un utilisateur et une application, tout en incluant ces deux derniers. Afin de connaître le contexte, il faut collecter les informations contextuelles. Le contexte peut être capturé, collecté explicitement ou implicitement [MPRB04].

Les systèmes de recommandation traditionnels se basent seulement sur les utilisateurs et les éléments pour faire leurs recommandations, sans prendre en compte le contexte actuel de l'utilisateur. Cependant celui-ci peut influencer les intérêts/besoins de l'utilisateur, c'est pourquoi il est important de prendre en compte ce type d'informations complémentaires afin d'obtenir des recommandations plus appropriées [BLPR12]. Ainsi, les systèmes de recommandation contextuels existants sont plus efficaces/pertinents (c'est-à-dire plus en adéquation avec les besoins de l'utilisateur) que les systèmes de recommandation traditionnels [AT08]. Cependant, leur performance est impactée par le contexte, qui est flou et omniprésent [BB05].

Avec l'intégration du contexte, le schéma de notre système générique de recommandation, illustré sur la figure 4.1, devient celui de la figure 6.6, où le contexte agit à tous les niveaux (besoin, entrées, étapes, sorties) du système de recommandation.

Notre objectif est d'aller vers une modélisation du contexte pour une meilleure prise en compte de celui-ci. Pour atteindre cette modélisation, dans [FNC17], à partir d'une étude bibliographique, nous identifions dans un premier temps tous les facteurs de contexte qui ont été étudiés.

Il est à noter que dans cette section, nous abordons la modélisation du contexte mais pas son intégration au sein du système de recommandation. Cependant, plusieurs techniques existent pour l'intégration du contexte dans le processus de recommandation : (i) le pré-filtrage, où seules les données qui correspondent au contexte actuel sont sélectionnées (comme entrées), puis une méthode de recommandation traditionnelle est appliquée sur celles-ci, (ii) le post-filtrage, où une méthode

13. En science de l'information, les notions de contexte et situation sont utilisées de manière interchangeable. [Coo01] suggère que les contextes sont les grandes lignes, c'est-à-dire l'environnement statique, et les situations sont les environnements dynamiques dans lesquels les processus d'interprétation se déroulent.

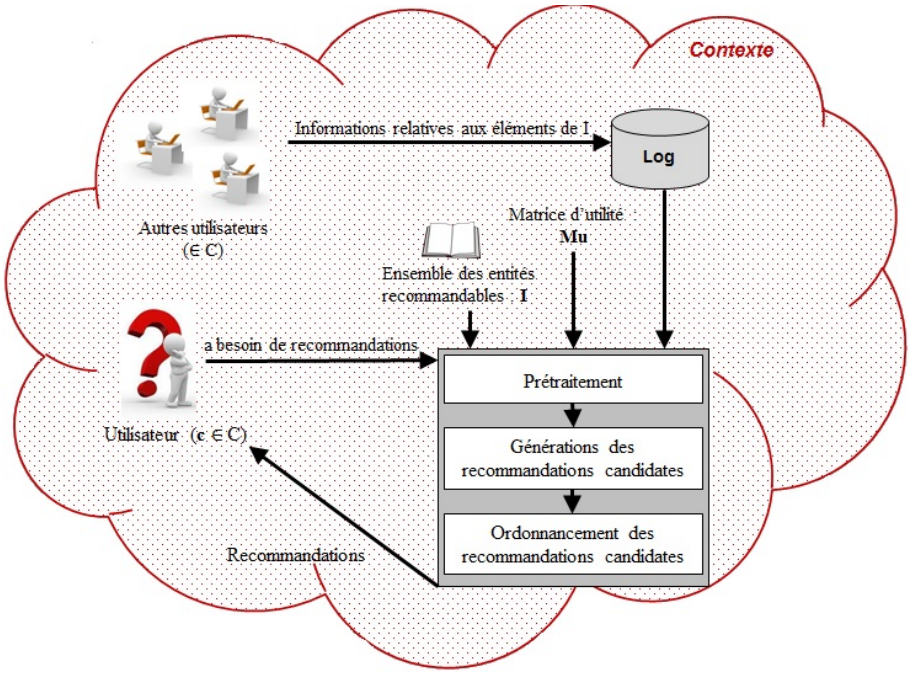

FIGURE 6.6 – Vue d'ensemble de notre système générique de recommandation avec prise en compte du contexte.

de recommandation traditionnelle est appliquée sur la totalité de l'ensemble de données, puis les résultats de la recommandation (sorties) sont ajustés en fonction du contexte, et (iii) le *context modeling* dans lequel les données et informations contextuelles sont directement prises en compte dans le processus de recommandation (les étapes) comme une dimension supplémentaire [AT08].

6.3.2 Les facteurs de contexte de l'utilisateur

Afin de décrire plus concrètement en quoi consiste le contexte d'un utilisateur, de multiples catégorisations de facteurs de contexte ont été proposées, certaines pour des domaines spécifiques comme la recherche d'information contextuelle [TLBD10], et d'autres, plus généralement pour les applications contextuelles. On retrouve plus d'une quinzaine de catégories à travers ces propositions.

A partir d'une étude bibliographique et des facteurs de contexte présenté par [AT11], nous avons regroupé/recoupé et structuré les travaux existants. Puis nous avons proposé une catégorisation hiérarchique, comme illustrée sur la figure 6.7. Cette catégorisation hiérarchique est un arbre, c'est-à-dire un graphe non orienté, acyclique et connexe, vu comme la généralisation de listes (puisque les listes peuvent être représentées par des arbres) favorisant ainsi son utilisation. Nos objectifs pour cette nouvelle proposition de catégorisation des facteurs de contexte sont de répondre aux besoins des systèmes de recommandation contextuels tout en (i) satisfaisant [14] la définition de [DSA01], (ii) compensant les lacunes (non-satisfaction de la définition de [DSA01] ou caractérisation non optimale)

14. Les caractéristiques des lieux, des personnes et des objets qui jouent un rôle dans l'interaction entre l'utilisateur et l'application sont respectivement représentés par les unités du contexte physique, par l'unité sociale du contexte utilisateur et par l'unité équipement du contexte physique. Les caractéristiques de l'utilisateur et de l'application apparaissent dans les contextes personnel et technique.

des précédentes propositions, (iii) permettant de travailler le contexte sur plusieurs niveaux [15] et (iv) permettant de l'appliquer sur un spectre assez large de cas d'applications [16].

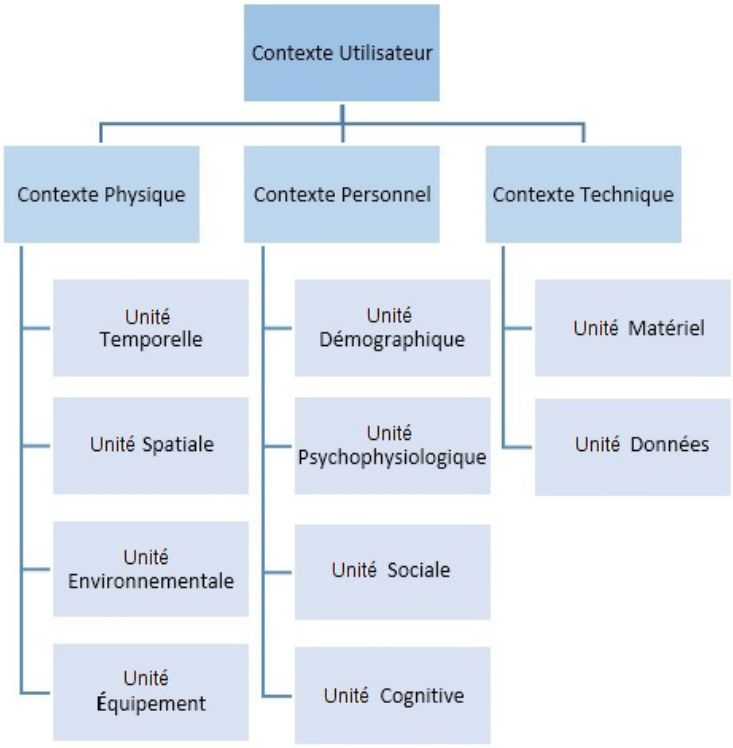

FIGURE 6.7 – Catégorisation des facteurs de contexte

Notre catégorisation a comme principales familles de contexte : le contexte physique, le contexte personnel et le contexte technique [FNC17]. Le contexte utilisateur est alors l'union de toutes ces familles de contexte, chacune composée de plusieurs unités [17] :

1. Le contexte physique regroupe tous les aspects sur lesquels la position géographique de l'utilisateur va avoir une forte influence. Nous avons regroupé plusieurs unités dans cette catégorie :
   (a) L'unité temporelle : comme le moment de la journée, semaine/weekend, la saison, les évènements (anniversaire, nouvel an, etc),
   (b) L'unité spatiale : qui selon le cas d'application peut être représentée par la position géographique exacte (coordonnées GPS, longitude/latitude) ou par des classes nominales (pour dire si l'utilisateur est chez lui, au travail, en voyage, etc).
   (c) L'unité environnementale : en fonction du cas d'application, cette unité peut représenter des caractéristiques environnementales comme la température, la météo, le niveau de lu-

15. Notre catégorisation à plusieurs niveaux peut s'adapter à différents cas d'applications et à leurs besoins, et permet de travailler le contexte de façon plus générale ou plus fine.

16. Nous espérons tendre vers une représentation, rassemblant tous les facteurs de contexte possible, pour une application au plus grand nombre de domaines.

17. Nous sommes conscients que notre catégorisation en trois grandes familles est discutable, mais comme toute catégorisation hiérarchique, il est possible de grouper/dissocier certaines familles/unités afin d'avoir un plus grand/petit nombre de familles/unités.

minosité ou le niveau sonore du lieu où l'utilisateur se trouve, et/ou la situation régionale de ce lieu, comme une guerre, une catastrophe naturelle, une crise économique, ...

(d) L'unité équipement : tout ce (non humain : objet ou espace) qui entoure l'utilisateur comme : barbecue, ustensiles, électroménager (friteuse, four, etc), imprimante, jardin/terrasse, ...

2. Le contexte personnel représente les informations plus spécifiques de l'utilisateur qui sont regroupées en quatre unités :
   (a) L'unité démographique regroupe notamment les informations sur l'identité (le nom, l'âge, le sexe, la nationalité, etc) de l'utilisateur.
   (b) L'unité sociale fait référence à la présence et au rôle des autres personnes autour de l'utilisateur. Selon le cas d'application, on peut prendre en compte seulement les personnes présentes pendant l'utilisation de l'application par l'utilisateur, les personnes avec lesquelles on veut partager l'activité en question, ou bien aller plus loin en prenant en compte les relations plus fines, comme les amis, la famille, les collègues, les voisins, ...
   (c) L'unité psychophysiologique représente l'aspect psychophysiologique de l'utilisateur, comme son état d'esprit, son humeur, son degré de fatigue, ...
   (d) L'unité cognitive fait référence aux expériences de l'utilisateur, ses objectifs, ses contraintes, son activité, ...
3. Le contexte technique représente les caractéristiques des dispositifs utilisés par l'utilisateur. Cela peut être :
   (a) Le matériel utilisé par l'utilisateur pour accéder au système de recommandation contextuel, comme le dispositif utilisé, les processeurs, la capacité du réseau, ...
   (b) Les données qui sont manipulées par l'application, leur format (texte, film, audio, image, etc), leur sources de provenance, la qualité de ces données, leur période de validité, leur exactitude, ...

Il est à noter que chaque unité sera instanciée en fonction du cas d'application, de la pertinence et de la disponibilité des indicateurs associés.

Synthèse Bien que ces dernières années beaucoup de progrès aient été accomplis dans le domaine des systèmes de recommandation contextuels, certaines difficultés subsistent. Au niveau de la définition du contexte nous avons pu constater qu'il n'existe toujours pas de définition unique et globale, ce qui gène l'identification de ses composants et sa modélisation. Nous avons donc proposé nos différents facteurs de contexte, à savoir le contexte physique et ses unités temporelle, spatiale, environnementale et équipement, le contexte personnel et ses unités démographique, psychophysiologique, sociale et cognitive, et le contexte technique avec ses unités matériel et données. Notre représentation du contexte utilisateur est à notre sens la plus complète, lisible et facile d'utilisation, par rapport à celles présentes dans la littérature, et permet ainsi de répondre aux besoins d'un spectre assez large de cas d'applications dans le cadre des systèmes de recommandation. Cette représentation devra ensuite être instanciée selon le cas d'application pour identifier et garder les indicateurs pertinents disponibles.

6.3.3 Dans la pratique...

Nous précisons dans cette section comment instancier notre catégorisation du contexte dans trois domaines d'applications, deux concernent les recommandations individuelles (d'éléments) : OLAP et un environnement commercial, et le troisième concerne les recommandations à visée publique (d'éléments) : la gestion de crise. Nous les présentons en fonction de la densité des informations contextuelles dans notre catégorisation, c'est-à-dire de la quantité de familles/unités de contexte prises en compte.

6.3.3.1 En OLAP

Recommander une requête OLAP à un décideur d'une grande entreprise française consistera à retourner des requêtes relatives au chiffre d'affaire annuel de la société. Mais si la société s'ouvre à l'international (États-Unis par exemple) ou si certaines données (anciennes) ont été archivées alors le décideur préfèrerait que le système lui recommande des requêtes « récentes » ou en rapport avec les États-Unis. En OLAP, les systèmes de recommandation prennent souvent un log parmi leurs données d'entrée. Or, les analyses OLAP (stockées dans le log) sont lancées régulièrement mais cette régularité peut être annuelle. Donc, le log peut contenir des données très anciennes sans rapport avec les analyses réalisées au moment de l'utilisation par le système de recommandation. De sorte que le système de recommandation retourne des recommandations non pertinentes. Prendre en compte des informations contextuelles permettrait de pallier ce manque de pertinence.

6.3.3.1.1 Familles et unités de contexte

Nous commençons par déterminer quelles informations contextuelles sont utiles pour les systèmes de recommandation en OLAP [Neg17b] (nous les présentons selon la catégorisation des facteurs de contexte présentée précédemment). Il est à noter que les familles/unités de contexte, qui n'apparaissent pas, ont été considérées comme inutiles en fonction des spécificités d'OLAP.

Contexte Physique

- Unité temporelle : Par définition, un entrepôt de données est une collection de données orientées sujet, intégrées, non volatiles et historisées, organisées pour le support d'un processus d'aide à la décision. De plus, les données de l'entrepôt varient au cours du temps. L'unité temporelle est donc importante.

Contexte Personnel

- Unité démographique : Les informations sur l'utilisateur sont la base des systèmes de recommandation pour aider l'utilisateur à trouver des informations pertinentes. De plus, d'après la règle 8 de [CCS93], les systèmes OLAP sont multi-utilisateurs, il faut donc pouvoir dissocier un utilisateur d'un autre. L'unité démographique contenant, entre autres, les préférences de l'utilisateur, doit donc être prise en compte.

- Unité sociale : D'après la règle 8 de [CCS93], les systèmes OLAP sont multi-utilisateurs et selon la définition de [CP06], il existe des relations entre les utilisateurs. Par conséquent, cette unité doit être prise en compte dans le cadre d'un système de recommandation par filtrage

collaboratif (similarité entre utilisateurs) ; mais n'est pas utile pour les systèmes de recommandation basés sur le contenu qui se focalisent sur un utilisateur donné.

- Unité cognitive : L'activité de l'utilisateur, courante ou passée, peut nous aider à avoir ses préférences ou encore connaître les caractéristiques des requêtes lancées. De plus, d'après la règle 9 de [CCS93], les utilisateurs OLAP ont peu de restrictions quant à leurs activités (croisements entre dimensions dont le nombre est illimité - règle 12).

Contexte Technique

- Unité Matériel : Connaître le dispositif matériel utilisé pour lancer une requête OLAP est intéressant. En effet, les ordinateurs portables, les tablettes et les smartphones, sont différents au niveau matériel, logiciel et communication, nécessitant des solutions d'interopérabilité. [Ngu10] et la règle 11 de [CCS93] appuient l'utilité de cette unité.

Dans la suite, nous proposons une modélisation de ces 5 unités pour leur prise en compte dans un système de recommandation de requêtes OLAP.

6.3.3.1.2 Modélisation

La modélisation du contexte reste encore compliquée compte tenu de la nature des données et/ou informations contextuelles : en effet le modèle doit pouvoir gérer les sources de données variées, l'hétérogénéité au niveau de leur qualité et de leur durée de vie et la nature imparfaite de celles-ci [HI06]. Il existe différents modèles de contexte [BBH+10] : attribut/valeur, mots clés, graphes, logique, ... D'après [SA15], qui a fait une comparaison entre les différents modèles de contexte, seul le modèle ontologique permet une bonne validation partielle des données et une bonne formalisation du modèle.

Unité temporelle Selon [SA15], une norme internationale (ISO 8601) [18] a été définie par l'Organisation Internationale de Normalisation pour la description des instants et des intervalles de temps, avec l'objectif d'éviter tout risque de confusion dans les communications internationales en raison du grand nombre de notations nationales différentes. L'ontologie OWL-Time [19] est aujourd'hui une reférence pour la représentation du temps [Pan07]. Au vu de son adéquation avec notre cas d'application, nous proposons d'utiliser l'ontologie OWL-Time [Pan07], voire sa version simplifiée [SA15], pour modéliser l'unité temporelle dans notre cadre de système de recommandation de requêtes OLAP.

Unité démographique Selon [SA15], la modélisation du profil repose sur des techniques permettant non seulement de représenter et construire le profil de l'utilisateur, mais aussi de gérer son évolution de manière dynamique au cours du temps. Une représentation sémantique du profil utilisateur, généralement basée sur une ontologie, permet de mieux représenter les différents acteurs du système en décrivant les relations sémantiques entre ces derniers, mais aussi avec les données collectées sur eux [HSM+07]. Chaque utilisateur du système est caractérisé par un ensemble de données. [SA15] propose de classer ces données en deux catégories : la première catégorie *General*

18. http ://www.iso.org/iso/catalogue detail ?csnumber=40874
19. http ://www.w3.org/TR/owl-time/

Information, regroupe des données générales sur l'utilisateur (nom, prénom, date de naissance, nationalité, ...) et la deuxième catégorie *Spécifique Information*, varie en fonction des différents types d'utilisateurs (par exemple, les préférences, l'employeur, les objectifs, ...). Au vu de son adéquation avec notre cas d'application, nous proposons d'utiliser l'ontologie relative au profil utilisateur de [SA15] pour modéliser l'unité démographique dans notre cadre de système de recommandation de requêtes OLAP.

Unité cognitive [AHC+14] ont proposé un modèle de conception générale d'ontologie pour modéliser l'activité et pouvoir l'utiliser dans des domaines différents. L'activité se compose de plusieurs activités qui peuvent également être associées à d'autres activités. Des relations de dépendance existent comme l'enchainement des activités, les participants à l'activité ainsi que le moment de début et de fin et la durée de l'activité. Au vu de son adéquation avec notre cas d'application, nous proposons d'utiliser l'ontologie relative à l'activité de [AHC+14] pour modéliser l'unité cognitive dans notre cadre de système de recommandation de requêtes OLAP.

Unité sociale Nous avons adapté/simplifié l'ontologie proposée par [Lou12] dans le cadre du réseau social Facebook. En effet, les relations existantes dans un réseau social comme Facebook sont beaucoup plus complexes que celles que nous souhaitons modéliser dans le cadre d'un système de recommandation de requêtes OLAP. A partir des définitions de [ZLO07] et [FMG15], où les relations sociales sont une caractéristique intrinsèque de l'être humain, et du modèle de [Lou12], nous proposons une ontologie pour l'unité sociale où :

- *Inbox* correspond à la boite aux lettres électronique de l'utilisateur où sont reçus et stockés les différents mails échangés via Internet ou des réseaux internes à l'entreprise ;
- *Outbox* correspond à la boite d'envoi électronique où sont stockés les mails envoyés par l'utilisateur ;
- *Organigramme* représente schématiquement les liens et les relations fonctionnelles, organisationnelles et hiérarchiques qui existent entre les éléments et les individus d'une entreprise ;
- *Work on the same project* correspond aux relations entre le décideur et les personnes qui ont travaillé sur le même projet que lui.

Unité matériel Les dispositifs matériels sont caractérisés par différentes propriétés telles que le degré de portabilité du dispositif, la résolution de l'écran, la puissance du processeur, la connectivité, ... et ils ont pour objectif de proposer une interface d'interaction entre le système et l'utilisateur. De nombreux travaux de recherche se réfèrent aujourd'hui à des spécifications standards pour la description des dispositifs dans un système, tels que WURFL [20] (sérialisation en XML), CC/PP [21] (sérialisation en RDF) et FIPA [22] DeviceOntology (sérialisation en OWL). Afin de modéliser l'unité matériel, nous proposons d'exploiter la FIPA DeviceOntology qui est la plus en adéquation avec notre cas d'application. Par ailleurs, [SA15] propose de décrire un dispositif matériel selon trois principales catégories d'informations : des informations générales sur le dispositif , des informations sur la partie matérielle (*hardware*) et des informations sur la partie logicielle (*software*). Au vu de son adéquation avec notre cas d'application, nous proposons d'utiliser l'ontologie relative au contexte matériel de [SA15] pour modéliser l'unité matériel dans notre cadre de système de recommandation de requêtes OLAP.

20. http ://wurfl.sourceforge.net/
21. http ://www.w3.org/TR/2007/WD-CCPP-struct-vocab2-20070430/
22. http ://www.fipa.org/specs/fipa00091/PC00091A.html

Ontologie générale Nous avons une ontologie [23] pour chacune des cinq unités de contexte détectées utiles dans le cadre d'un système de recommandation de requêtes OLAP (Unité temporelle, Unité démographique, Unité cognitive, Unité sociale, Unité matériel). Selon [FMG15], la meilleure façon pour regrouper ces cinq ontologies est de les regrouper dans une seule ontologie afin de ne pas perdre les relations sémantiques. Par conséquent, nous proposons de regrouper nos cinq ontologies en une seule plus générale comme illustré sur la figure 6.8 (par souci de lisibilité, nous nous limitons à afficher le premier niveau de chacune des cinq ontologies - (sous Protégé [24])).

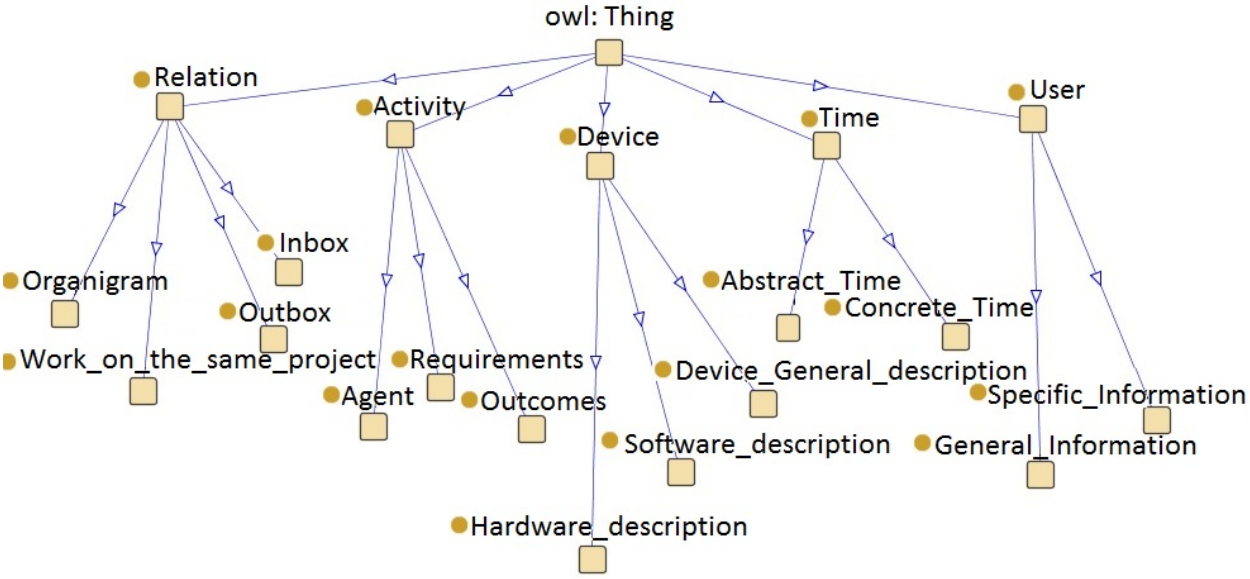

FIGURE 6.8 – Ontologie générale de notre modélisation de contexte (cinq unités) dans le cadre d'un système de recommandation de requêtes OLAP

Synthèse Nous nous intéressons à la notion de contexte qui influence les préférences, les envies, les intérêts et les décisions des utilisateurs, dans le cadre des systèmes de recommandation de requêtes OLAP. En effet, intégrer les informations contextuelles dans de tels systèmes permettra d'avoir des recommandations plus en adéquation avec les attentes des utilisateurs. Nous avons détecté cinq unités utiles dans le cadre d'un système de recommandation de requêtes OLAP puisqu'elles respectent les spécificités d'OLAP. Puis nous avons proposé une modélisation de ces cinq unités sous la forme d'une ontologie (actuellement implémentée sous Protégé) à intégrer dans le système de recommandation (qui devient contextuel). Finalement, afin d'améliorer la recommandation en prenant en compte le contexte, cette ontologie générale devra être intégrée dans le système de recommandation de requêtes OLAP en tant que données/sources externes. Elles auront un impact sur la pertinence des recommandations retournées à l'utilisateur.

23. Pour rappel, [SA15] indique que le modèle ontologique est le plus approprié pour les données contextuelles.
24. http ://protege.stanford.edu/

6.3.3.2 Dans un environnement commercial

Nous nous sommes intéressés aux données/informations contextuelles dans un environnement commercial particulier : celui de la vente de véhicules d'occasion [ANRS13, AMNR14]. Il est à noter que nous avons commencé à partir d'observations « terrain » et que l'approche proposée est spécifique au cas rencontré.

Acheter un véhicule d'occasion En France, en général, pour vendre un véhicule à un distributeur, ce véhicule doit être en bon état et pas trop vieux. Le distributeur achète votre véhicule uniquement si vous achetez un nouveau dans son entreprise. Il peut se permettre d'offrir un prix d'achat plus élevé car il fera une marge bénéficiaire significative avec le véhicule que vous achèterez. La promesse de la société sur laquelle porte notre étude, est d'acheter tous les véhicules sans condition, quel que soit leur état et sans obligation d'acheter un autre véhicule.

Vendre un véhicule d'occasion Lorsqu'une personne (privée), Elsa, veut vendre son véhicule, elle demande une évaluation préalable du véhicule d'occasion sur le site de la société. Elle donne des informations sur les principales caractéristiques du véhicule telles que la marque, le modèle et l'âge. À la suite de cette requête, elle obtient une évaluation du prix d'achat du véhicule. Le système ne prend pas en compte l'usure du véhicule et il estime que le véhicule est en bon état. Lors de l'interrogation pour obtenir une première évaluation sur le site web, Elsa renseigne son adresse et son numéro de téléphone, elle sera contactée par la société pour prendre rendez-vous dans une agence. Quand Elsa va à l'agence, elle est reçue par un expert automobile, John. Il examine le véhicule et évalue les coûts pour reconditionner le véhicule. Cette étape est très importante car elle contrôle l'état de fonctionnement du véhicule, John doit faire un essai routier et identifier d'éventuelles anomalies. Les données sont intégrées et stockées via une plateforme informatique afin de demander aux experts en cotation une cotation du véhicule. La valeur que John obtiendra du service des achats à travers cette plateforme sera la meilleure proposition et John n'aura aucune marge de manœuvre sur cette proposition[25].

6.3.3.2.1 Familles et unités de contexte

Grâce à des entretiens et des échanges avec des experts et des vendeurs, il apparait que la prise en compte du contexte améliorerait l'évaluation du prix du véhicule ainsi que le succès de la transaction. Ainsi, les facteurs de contexte utiles (présentés selon notre catégorisation des facteurs de contexte) sont :

Contexte Physique

- Unité spatiale : *La localisation du véhicule.* Par exemple, les experts savent que certains véhicules seront vendus différemment en fonction de leur situation géographique. La localisation influence John et la décision finale de la société. Nous avons trois cas : le véhicule est situé

25. La façon dont les véhicules sont évalués n'est pas discutée ici. Cependant, cette évaluation est imprévisible parce qu'elle dépend de chaque expert en cotation. Pour mieux estimer le marché, ces experts utilisent différents sites publicitaires vendant des véhicules sur le web et examinent les prix. Ils évaluent le véhicule de sorte qu'il soit bien placé parmi les annonces similaires. Cette approche conduit parfois à ce que le prix d'achat offert au particulier ne soit pas conforme au prix réel de revente.

dans une zone géographique où les véhicules sont vendus à bas prix (L1), à prix moyen (L2), à des prix élevés (L3). Meilleure sera la localisation, meilleure sera la note. Ici, L3 est meilleure que L2 et L1.

- Unité environnementale : *Le marché*. Par exemple, si un véhicule est très populaire, la société le gardera en stock moins de temps, donc ce sera moins cher. Cependant, si un véhicule n'est pas populaire, il se passera plus de temps entre l'achat du véhicule et sa revente, il y a donc un coût de stockage supplémentaire pour la société. Le marché influence John et la décision finale de la société. Nous avons quatre cas : le véhicule n'est pas populaire (M1), le véhicule est modérément populaire (M2), le véhicule est populaire (M3) ou le véhicule est très populaire (M4). Meilleur sera le marché, meilleure sera la note. Ici, M4 est meilleure que M3, M2 et M1.

- Unité Équipement : *Les coûts de réparation*. Par exemple, une simple égratignure peut provoquer le remplacement d'une partie importante du véhicule et seul un expert peut le savoir. Les coûts de réparation peuvent être difficiles à évaluer notamment lorsque le véhicule ne peut pas être étudié en détail. Notre proposition donne à John l'opportunité de dire rapidement si, selon lui, les coûts de réparation auront un impact sur le prix final du véhicule, même si le véhicule n'est pas expertisé en détail. John peut pouvoir dire, par exemple, si une simple égratignure entraînera le remplacement d'une partie importante de la carrosserie du véhicule. Ces coûts de réparation influencent John et la décision finale de la société. Nous avons quatre cas : peu de réparations (juste polie) (R1), quelques réparations (égratignures uniquement) (R2), un certain nombre de réparations (frais requis) (R3) ou de nombreuses réparations (R4). Moins il y aura de réparations, meilleure sera la note. Ici, R1 est meilleur que R2, R3 et R4.

Contexte Personnel

- Unités démographique, psychophysiologique, sociale et cognitive : *Le contexte personnel d'une vente particulière*. Par exemple, les problèmes financiers ou familiaux peuvent engendrer un besoin urgent de vendre le véhicule quel que soit le prix ou bien, cela peut entraîner une tentative de négociation du prix aussi élevé que possible. Le contexte personnel au moment de la vente pourrait avoir une forte influence sur le prix final et c'est la raison pour laquelle il doit être considéré par notre proposition (toute information/donnée personnelle est intéressante). Ce contexte est important pour le vendeur et influence sa décision finale. Nous avons trois cas : le propriétaire n'est pas pressé de vendre sa voiture (T1), le propriétaire est moyennement pressé de vendre sa voiture (T2) ou le propriétaire est pressé de vendre sa voiture (T3). Meilleur sera le contexte personnel au moment de la vente, meilleure sera la note. Ici, T1 est meilleur que T2 et T3.

6.3.3.2.2 Modélisation

Afin d'intégrer ces informations contextuelles pour améliorer la recommandation, il faut les modéliser. Pour cela, nous proposons de modéliser ce contexte en combinant une approche d'aide à la décision multicritère et une approche coût-bénéfice. Les « meilleures » décisions possibles obtenues avec notre modèle pourront ensuite être utilisées par le système de recommandation comme des données/sources externes complémentaires.

Bien que, [DB09] semble interpréter que les résultats d'une approche d'aide à la décision multicritère [FMR05, Roy96] soient plus mauvais que ceux d'une approche coût-bénéfice [Nas96], ces

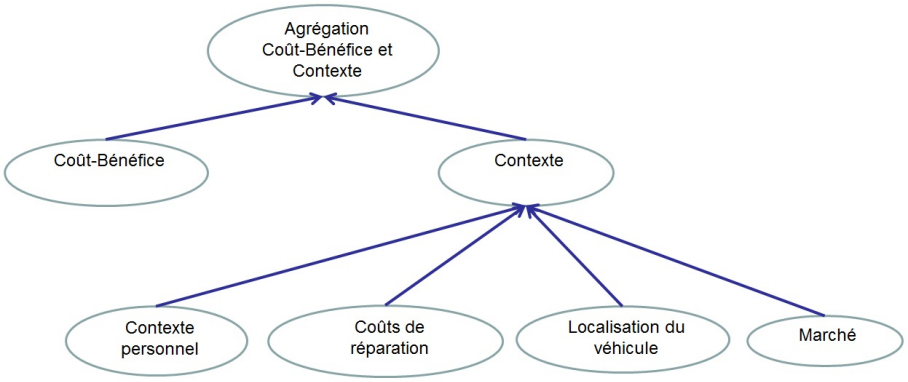

FIGURE 6.9 – Un modèle hiérarchique d'aide à la décision multicritère vu comme une agrégation des coûts-bénéfices et du contexte

deux types d'approches partagent des similarités :

- les deux sont des cadres pour évaluer les options auxquelles sont confrontés les décideurs,
- les deux essayent de construire une métrique pour comparer des options,
- les deux sont sensibles aux hypothèses, mais ces hypothèses sont différentes.

Notons que l'analyse coût-bénéfice est déjà mise en œuvre et utilisée par la société et l'analyse multicritère est efficace pour des coûts et des bénéfices non mesurables. La société ne veut qu'un seul système. Donc, nous devons procéder en deux étapes/analyses puis les combiner. Nous tirons avantage de chaque type d'analyse. L'analyse coût-bénéfice est pertinente et facile à utiliser pour des coûts mesurables et prend en compte les attentes de la société (« *business is business* »). L'analyse multicritère est pertinente pour prendre une décision et permet de mettre en jeu des coûts non mesurables. Ainsi, notre proposition consiste à utiliser l'analyse coût-bénéfice pour les coûts mesurables (quantitatifs) et à utiliser l'analyse multicritère pour les coûts non mesurables (qualitatifs).

La combinaison de ces deux approches peut se faire en utilisant une approche d'aide à la décision multicritère avec deux critères :

- le premier critère concerne le coût-bénéfice,
- le deuxième critère sera une agrégation des facteurs de contexte, en utilisant la méthode ELECTRE TRI [FMR05], où l'évaluation d'un véhicule sur ce critère sera l'une des six catégories prédéfinies C1 à C6 [26].

Cette approche combinée est illustrée sur la figure 6.9 selon une hiérarchie de critères. Étant donné que les données relatives au coût-bénéfice n'étaient pas fournies par la société, nous avons modélisé uniquement, dans cette hiérarchie, le nœud associé aux facteurs de contexte.

26. Soit un véhicule évalué en fonction du contexte personnel, des coûts de réparation, de la localisation du véhicule et du marché, l'objectif est d'offrir à John une marge de manœuvre sur le prix du véhicule : aucune augmentation (C1), augmenter d'au plus 5% l'estimation faite par les experts en cotation (C2), augmenter d'au plus 10% (C3), augmenter d'au plus 15% (C4), augmenter d'au plus 20% (C5) et augmenter d'au plus 25% (C6). Ces paramètres, ainsi que la plupart des paramètres introduits dans cette étude, ont été définis par la société considérée en accord avec sa politique financière.

Synthèse Nous soulignons les atouts de certaines données/informations contextuelles dans le processus décisionnel. Ainsi, nous avons élaboré un modèle d'aide à la décision multicritère en fonction de quatre critères/informations contextuelles (contexte personnel, coûts de réparation, localisation du véhicule, marché). Afin de respecter la politique financière de la société, nous avons utilisé l'analyse coût-bénéfice déjà mise en œuvre [27]. Notre modèle joue un rôle important dans l'amélioration du prix final du véhicule, de sorte que non seulement les attentes financières de la société sont prises en compte, mais aussi le prix final du véhicule satisfait le vendeur et assure que la transaction sera effectuée. Finalement, les « meilleures » décisions obtenues via notre modèle d'aide à la décision multicritère seront intégrées comme des informations contextuelles, c'est-à-dire des données/sources externes complémentaires, à un système de recommandation, comme illustré par la figure 6.6.

6.3.3.3 En gestion de crises

Dans le cadre de la gestion de crises, les informations contextuelles sont nombreuses. Celles relatives à l'individu peuvent être obtenues via son comportement, c'est-à-dire qu'elles sont collectées implicitement. Nous nous sommes donc intéressés au comportement des populations. Ainsi, nous avons mis en adéquation les concepts de comportement et de contexte.

6.3.3.3.1 Comportement

Le comportement est un concept qui nécessite d'être précisé et bien défini, il s'agit d'un concept « nomade » qui peut prendre plusieurs sens selon les disciplines [Ton09]. Nous reprenons ici la définition proposée par [Sil83] pour qui le comportement correspond aux « réactions d'un individu, considéré dans un milieu et dans une unité de temps donnée à une excitation ou un ensemble de stimulations ». Cette définition permet de situer clairement le comportement dans un espace spatio-temporel et comme une réponse à un ensemble d'excitations ou stimulations. Notons que nous limitons l'étude des comportements aux réactions observables par une entité extérieure. Il s'agit de travailler à établir des tendances, des règles qui permettent de déterminer des facteurs qui orientent des comportements particuliers. Pour cela, il nous semble nécessaire de décrire les comportements individuels en décomposant le plus finement possible les différents facteurs à l'aide d'éléments mesurables : les indicateurs.

La figure 6.10 permet de représenter notre cheminement des indicateurs vers le comportement : à partir des indicateurs, en passant par les facteurs (constitués d'indicateurs), nous obtenons les comportements.

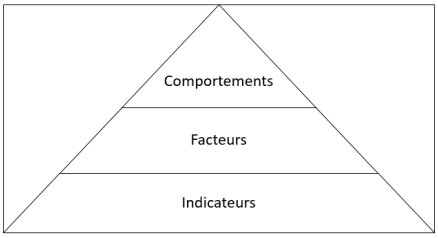

FIGURE 6.10 – Des indicateurs vers les comportements

27. Nous sommes conscients que l'analyse multicritère peut être utilisée seule (sans combinaison avec l'analyse coût-bénéfice), mais parce que l'analyse coût-bénéfice est déjà mise en œuvre et utilisée par la société, nous proposons une approche combinant l'analyse multicritères et l'analyse coût-bénéfice.

6.3.3.3.2 Les facteurs de comportement

Les chercheurs en sciences humaines, sociologues, psychologues sont capables d'analyser des comportements à partir d'observations directes ou rapportées et de produire des résultats de recherche à partir de ces observations. Malheureusement, en informatique nous n'avons pas les moyens de produire des résultats à partir de ce niveau des comportements, ni même à partir de celui des facteurs du comportement. Les comportements sont dérivés de connaissances difficilement explicitables, et donc par définition ils ne sont pas manipulables par des moyens informatiques. Pour produire des analyses et obtenir des résultats en informatique, nous nous situons au niveau des données qui, elles, peuvent être manipulées et agrégées de façon automatique pour produire des informations, voire des connaissances. Les vingt facteurs (et les indicateurs associés) du comportement que nous présentons ci-après ont été identifiés à partir de recherches dans la littérature comme ayant une influence sur les réactions humaines en situation de crise [AN17, ANRS17a, ANRS17b].

6.3.3.3.3 Familles et unités de contexte

Nous présentons les facteurs de comportement selon la catégorisation des facteurs/unités de contexte présentée précédemment.

Contexte Physique

- Unité temporelle
  - Caractéristiques de la période pendant laquelle survient la crise : les comportements des personnes sont différents selon que la crise les surprend de jour ou de nuit (F15a) et selon le nombre d'heures qui les séparent du pic de la crise (F15b) [PDPV$^+$15].
  - Phase temporelle de la crise : nous nous intéressons aux comportements des personnes avant la crise (F16a), à son commencement (F16b) jusqu'à son pic (F16c) et sa phase descendante (F16d) ainsi que dans l'après crise (F16e) qui correspond à une phase de retour à des comportements habituels [PDPV$^+$15].

- Unités spatiale et environnementale
  - Caractéristiques de la zone géographique : les premiers éléments pris en considération dans les études sur la vulnérabilité des personnes dans les situations de crise sont les caractéristiques des zones géographiques concernées [CMN14, CMN15]. Ils comprennent notamment l'étendue de la zone (F12a), la densité de la population (F12b), le niveau de pauvreté (F12c), le statut économique (F12d), le niveau d'urbanisme (F12e), la pyramide des âges (F12f) ou encore les caractéristiques culturelles : culture de l'individualité ou de la collectivité (F12g), confiance dans le gouvernement (F12h).

- Unité Équipement
  - Capacité d'interaction et de mobilité : sur la capacité d'interaction nous prenons en compte à la fois les interactions physiques qui dépendent de la fréquentation de la zone (F13a) et à la fois les interactions virtuelles qui dépendent principalement du nombre de smartphones par habitant (F13b). La capacité de mobilité est représentée par l'accès aux transports (F13c).

Contexte Personnel

- Unité démographique :
  - Etat civil : les caractéristiques individuelles ont une influence statistique évidente dans les réactions que les individus peuvent avoir face à une situation donnée. Nous retenons les indicateurs suivants comme éléments principaux, qu'on retrouve communément dans les études sociologiques [28] : l'âge (F1a), le sexe (F1b), la nationalité (F1c), le lieu de résidence (F1d), le niveau d'étude (F1e), le métier (F1f).

- Unité psychophysiologique
  - Personnalité : nous nous sommes inspirés d'une reprise du modèle de personnalité FFM [ZAV10] pour définir nos cinq composants de la personnalité : les désirs (F2a), les principes moraux (F2b), la sociabilité (F2c), les croyances, qu'elles soient religieuses ou non (F2d) et la capacité à décider (F2e), auxquels nous avons ajouté la réactivité mimétique (F2f).
  - Motivation à évacuer/à défendre : évacuer la zone de danger (F3a) et se défendre en luttant contre le danger (F3b) sont deux grandes réactions attendues en situation de crise [AG16].
  - Emotions : nous nous limitons aux émotions qui peuvent se lire sur les visages [EF71] : joie (F5a), tristesse (F5b), colère (F5c), dégoût (F5d), peur (F5e), surprise (F5f).
  - Signaux physiologiques : les quatre indicateurs physiologiques que nous retenons peuvent être mesurés à l'aide de capteurs, le rythme cardiaque (F11a), la tension, (F11b), le niveau de transpiration (F11c) et le niveau de contraction musculaire (F11d) qui peuvent être mesurés lors de simulations ou d'exercices grâce à des tensiomètres ou des montres intelligentes par exemple.

- Unité sociale
  - Responsabilité : en cas de crise dans une organisation, on attend naturellement un contrôle et une prise en charge de la situation par les meneurs, les personnes responsables d'un groupe de par leur statut ou de par leurs compétences [WJ08]. Le niveau de responsabilité d'une personne (F4) permet de mesurer ce facteur.
  - Caractéristiques de l'entourage : un des premiers réflexes en situation de crise est de rechercher des amis, d'échanger avec des proches [Dup03]. Nous retenons comme caractéristiques de l'entourage la densité des personnes à proximité de l'individu (F18a), la présence de représentants de l'autorité, policiers... (F18b), le niveau de sécurité de la zone (F18c) et la présence de relations proches (F18d).
  - Comportement des personnes les plus proches : les études sur la contagion sociale tendent à démontrer que l'influence des personnes les plus proches géographiquement ont une influence plus importante que l'entourage global sur les actions des individus [GJ05]. Les indicateurs que nous choisissons pour caractériser ce facteur sont les deux comportements dominants dans l'entourage proche F19a et F19b, qui dominent par leurs niveaux de contagion.
  - Comportement global de l'entourage : il s'agit du comportement dominant global (F20) qui est adopté par la population dans un périmètre étendu [Dup03].

- Unité cognitive
  - Expérience et capacités : le vécu d'une personne peut avoir une grande influence sur ses réactions, selon son expérience d'une autre crise dans le passé (F6a), ses capacités objectives

28. https ://www.insee.fr/

(physiques et intellectuelles) à évacuer et à se défendre (F6b) et ses capacités subjectives à évacuer et à se défendre (F6c) [AG16].
- Connaissances explicitées : les connaissances explicitées des personnes peuvent jouer un rôle primordial dans certaines situations, notamment dans l'évaluation du risque, et sauver de nombreuses vies [ANRSG16]. Ces connaissances peuvent être répertoriées en trois grandes catégories, les connaissances communes, partagées par l'ensemble d'une population à une échelle régionale ou nationale (F7a), les connaissances acquises lors de formations de préparation au risque (F7b), les connaissances fournies par l'accès à des documents relatifs au danger (F7c) et par l'accès à des outils de partage des connaissances (F7d).
- Evaluation du risque : la perception subjective du risque peut concorder avec le risque objectif ; ou pas [ST71]. Nous considérons que l'évaluation du risque est composée à la fois d'une partie objective (F8a) qui dépend beaucoup des signaux perceptibles de la crise et d'une partie subjective (F8b) qui, elle, dépendra principalement de la personnalité de l'individu et des émotions ressenties.
- Perception du système d'alertes : nous définissons la perception comme le processus de collecte, d'organisation et d'interprétation de stimuli qui peuvent être des informations ou des connaissances provenant de différentes sources. L'indicateur correspondant à ce facteur (F9) peut être mesuré à travers des réponses à un questionnaire ([AMN16] et section 5.1.2).
- Actions courantes : certaines composantes des actions humaines peuvent être déterminés automatiquement à partir de vidéos [NWFF08]. Nous différentions ici les types d'actions qu'une personne réalise au moment où une crise survient ou est annoncée selon leurs degrés d'interaction (F10a), de concentration (F10b) et de mouvement (F10c).

Contexte Technique

- Unité Données
  - Signaux perceptibles de la crise : nous définissons trois indicateurs pour ce facteur, les signaux visuels (F14a), les signaux sonores (F14b) et les signaux olfactifs (F14c).
  - Alertes/Informations transmises : nous avons choisi de caractériser les alertes et informations transmises selon leur quantité (F17a) qui ne doit être ni trop faible ni trop importante, leur qualité (F17b) évaluée selon les informations et recommandations fournies, et selon le nombre de canaux de diffusion utilisé (F17c) [DGS13].

6.3.3.3.4 Modélisation

Dans le cadre de la gestion de crise, modéliser le contexte en intégrant nos facteurs donne la possibilité de prendre en compte les réactions humaines. Avec un tel modèle, il est possible de tester des hypothèses sur la participation des différents facteurs aux comportements de crise pour une population donnée. Il est également possible de distinguer l'importance des différents facteurs impliqués pour différents types de crise. Les résultats obtenus à partir de ces analyses pourraient aider à la prévention et à la préparation des programmes de gestion des crises et aider à apporter des modèles qui fournissent des prédictions en temps réel. Une combinaison de différents modèles semble nécessaire à la fois pour représenter les définitions et la complexité de l'interaction, et pour pouvoir agréger les données pour offrir une vision synthétique des hypothèses validées. Beaucoup d'approches proposent déjà des combinaisons de plusieurs modèles [LL91, DPP06, SGW16]. Nous avons retenu trois choix qui peuvent intégrer nos facteurs dans des objectifs d'analyse mathématique ou sémantique : les modèles attribut-valeur, les ontologies et les modèles prédictifs :

- modèles attribut-valeur pour une analyse statistique : un modèle attribut-valeur peut être directement extrait de nos indicateurs. Selon l'analyse choisie au préalable, nous pouvons sélectionner les indicateurs et corrélations à tester. La méthode ACM (Analyse des Correspondances Multiples) pourrait être particulièrement adaptée pour tester des hypothèses à partir de ce type de modèle, elle permet de traiter les liens de corrélation entre n'importe quel nombre de variables. Elle est très souvent utilisée dans l'analyse de réponses à des questionnaires (qui est, comme indiqué dans le chapitre précédent, une méthode très utilisée en gestion de crise). En pratique, nous pouvons utiliser cette méthode pour expliquer un ou plusieurs indicateurs à partir d'un autre ensemble d'indicateurs. Malheureusement, cette méthode ne permet pas de reconnaître les relations déjà connues entre indicateurs.

- ontologies et règles de décision : des ontologies ont déjà été proposées pour représenter des stratégies comportementales [Sil01, SJCO06] ainsi que des informations contextuelles [SA15]. Il est possible d'intégrer nos facteurs de comportement en implémentant une ontologie, et en y ajoutant un module de mémoire de travail qui permettrait de tirer avantage de règles de décisions pour découvrir de l'information à partir de données collectées. En informatique, on définit les ontologies comme des « spécifications formelles et explicites des termes d'un domaine et les relations entre ces termes » [Gru93]. Issue de l'ingénierie des connaissances, elles permettent de représenter des relations sémantiques avec des liens de composition ou d'héritage par exemple. Leur objectif est de donner un formalisme sémantique consensuel, normatif, cohérent, partageable et réutilisable de l'information tel qu'il puisse être utilisé par un ordinateur.

- modèles prédictifs et analyses de corrélation : en partant d'une information complète sur des situations passées ou des études statistiques, nous pourrions construire un réseau Bayésien ou des modèles de processus markoviens qui permettraient de réaliser des prédictions à partir de données existantes dans les systèmes de gestion de crise et de tables de probabilités. Aujourd'hui, nous n'avons pas cette possibilité. Un réseau Bayésien est, en informatique et en statistiques, un modèle graphique probabiliste représentant des variables aléatoires sous la forme d'un graphe orienté acyclique. Les réseaux Bayésiens sont à la fois (i) des modèles de représentation des connaissances (ii) des calculateurs de probabilités conditionnelles, et (iii) une base pour l'aide à la décision. Ici, nous pourrions décrire les relations causales entre variables à partir de ce type de graphe. Les relations causales entre les variables ne sont pas déterministes mais bien probabilistes. Par conséquent, l'observation d'une cause n'implique pas systématiquement les effet(s) qui dépendent d'elle, mais elle modifie la probabilité de les observer. L'intérêt d'utiliser des réseaux Bayésiens est de tenir compte simultanément des connaissances a priori (représentées par le graphe), et de l'expérience apportée par les données.

Ces choix de modélisation devront être affinés pour enrichir une représentation au moins en partie probabiliste, ou reposant sur la logique floue qui sont les deux bases théoriques pour raisonner et prendre des décisions dans des situations incertaines [Gai78]. En effet, le comportement ne se réduit pas à un état ou une série d'états telle une succession de points qui définirait la trajectoire d'une conduite toute tracée. Il contient de l'incertitude [Ton09]. La proposition de trois modèles pour représenter le contexte et plus précisément, les comportements des personnes en situation de crise reste la partie théorique de notre projet de recherche. De toute évidence, ces trois modèles devront être mis à l'épreuve avec probablement une réévaluation des modèles, et un besoin de représentations

plus détaillées pour pouvoir interpréter les résultats. Il convient de noter que, à première vue, une combinaison de ces trois modèles, combinant connaissance, expertise, expérience et technologie / ingénierie statistique, pourrait être envisagée, permettant de bénéficier de chacune, obtenant ainsi la modélisation la plus complète et aussi la plus réaliste possible.

Synthèse Nous nous intéressons à la prise en compte du contexte dans le cadre de la gestion de crise et plus précisément pour l'analyse des comportements des populations. Ainsi, nous mettons en adéquation les facteurs de comportement avec les facteurs de contexte que nous avons défini. Notre proposition permet donc de prendre en compte les informations identifiées comme ayant une influence sur le comportement. Les indicateurs de comportement (F1a, F1b, ...) permettent quant à eux de caractériser indirectement les comportements. Chacun contenant des informations qui peuvent être manipulées par des algorithmes. Finalement, ces données/informations sont des informations contextuelles qui pourront être intégrées à un système de recommandation dans le domaine de la gestion de crise. En tant que sources externes, elles permettront d'enrichir, entre autres, le profil de l'utilisateur, et pourront avoir un impact sur la pertinence des recommandations retournées par le système.

6.4 Bilan

Dans ce chapitre, nous avons présenté différentes propositions pour améliorer la recommandation : (i) en améliorant les données d'entrée, via la structuration de logs/journaux, (ii) en gérant le démarrage à froid et (iii) en ajoutant des données/sources externes, via la prise en compte du contexte de l'utilisateur.

Nous avons proposé et testé deux approches de structuration de logs/journaux dans le cadre de logs/journaux de requêtes OLAP : soit sous la forme de graphes, soit sous la forme de résumés. Les tests réalisés dans le domaine OLAP ont montré le gain apporté par une meilleure organisation, à savoir une meilleure facilité de traitement du log, l'émergence de données/informations complémentaires et une aide à la consultation du log.

Notre approche de démarrage à froid prédictive a été proposée et testée dans le cadre d'un système de recommandation de requêtes OLAP. Cette proposition permet de pallier une insuffisance (connue) des systèmes de recommandation actuels, à savoir la gestion de la double problématique du démarrage à froid : nouvel utilisateur et nouvel élément.

Enfin, nous avons présenté notre catégorisation du contexte selon trois catégories et dix unités de contexte. Cette catégorisation de contexte a ensuite été instanciée en OLAP, dans un environnement commercial (de ventes de véhicules d'occasion) et en gestion de crise (via les comportements humains). Le tableau 6.1 récapitule les informations contextuelles pertinentes pour chaque cas d'application. Nos propositions montrent ainsi quelles informations contextuelles peuvent être prises en compte dans le processus de décision et comment les modéliser en vue de leur intégration dans le système de recommandation. Ces données/sources externes permettront de mieux connaître l'utilisateur, d'enrichir son profil, d'améliorer la pertinence des recommandations, de densifier les données d'entrée avec des données complémentaires et ainsi d'améliorer les recommandations.

Les expérimentations que nous avons pu mener ont montré l'intérêt de nos propositions ainsi que leur applicabilité.

Contexte	Unité	OLAP	Commerce (véhicules d'occasion)	Gestion de crise (comportements)
Physique	Temporelle	✓		✓
	Spatiale		✓	✓
	Environnementale		✓	✓
	Équipement		✓	✓
Personnel	Démographique	✓	✓	✓
	Psychophysiologique		✓	✓
	Sociale	✓	✓	✓
	Cognitive	✓	✓	✓
Technique	Matériel	✓		
	Données			✓

TABLE 6.1 – Récapitulatif des informations contextuelles pertinentes selon les domaines/cas d'applications

Chapitre 7

Outils, Projets, Publications

Sommaire

7.1 Outils 132
7.1.1 RecoOLAP 132
7.1.2 MEMORAe-CRS 134
7.2 Projets et Collaborations 138
7.2.1 Vision synthétique 138
7.2.2 Présentation des projets et/ou Collaborations 138
7.3 Encadrements et Publications 141
7.3.1 Encadrements 141
7.3.2 Publications 142
7.4 Bilan et Synthèse 143

Les précédents chapitres décrivent les travaux que nous avons menés dans le cadre des systèmes de recommandation (individuelle ou à visée publique), que ce soit pour leur conception, leur évaluation ou leur amélioration dans des domaines tels que OLAP, l'apprentissage en ligne, les environnements de travail collaboratif, les villes intelligentes et les systèmes d'alertes précoces. L'objectif de ce chapitre est de décrire les différentes productions qui ont permis de mettre en œuvre et de valider les concepts et principes proposés. Nous présentons les différents prototypes que nous avons développés, les projets auxquels nous avons participés ainsi que les encadrements et publications effectués.

Afin de valider la faisabilité et l'instanciabilité de notre cadre générique (présenté dans le chapitre 4), nous commençons par présenter les prototypes qui ont été développés. Dans le cadre des recommandations à visée publique, l'accès aux données et les enjeux liés à la prise de décision dans des domaines tels que les villes intelligentes et la gestion de crise ne nous ont pas encore permis de valider nos propositions. Par contre, dans le cadre des recommandations individuelles, deux prototypes existent pour OLAP et les environnements de travail collaboratif (celui pour l'apprentissage en ligne est en cours d'élaboration). Ainsi, le premier prototype que nous avons défini est le système de recommandation de requêtes OLAP : RecoOLAP (commencé dans [Neg09] et introduit dans la section 4.2.1.1). Le second prototype est le système de recommandation d'experts dans un environnement de travail collaboratif : MEMORAe-CRS (introduit dans la section 4.2.1.3).

Les autres productions scientifiques se matérialisent au travers d'articles dans des livres, revues et conférences internationales et nationales. Notre objectif était de valider l'ensemble des concepts,

modèles et outils que nous avons présentés dans les chapitres précédents (systèmes de recommandation, évaluation et amélioration). Une partie des publications a été co-écrite avec des personnes que j'ai co-encadrées. Ce chapitre vise également à recenser ces différents encadrements.

Afin de montrer, entre autres, l'intérêt et l'applicabilité de nos travaux, nous souhaitons présenter les différents projets auxquels nous avons participé. Ces projets ont abouti à des collaborations avec des laboratoires de recherche et des entreprises.

Les sections suivantes présentent de manière détaillée ces différents outils ainsi que les projets auxquels j'ai participé sans oublier mes encadrements et publications.

7.1 Outils

Nous présentons nos deux prototypes : RecoOLAP et MEMORAe-CRS.

7.1.1 RecoOLAP

Présentation

Le domaine de l'OLAP a été un domaine majeur de nos travaux de recherche (conception des systèmes de recommandation, évaluation, amélioration). Développer un système de recommandation dan ce domaine était donc incontournable. Ainsi, nous avons proposé RecoOLAP.

RecoOLAP est un système de recommandation de requêtes OLAP. Le but de notre système de recommandation est d'aider l'utilisateur à naviguer dans un cube de données en exploitant ce que les autres utilisateurs ont fait pendant leurs navigations précédentes. Pour rappel, notre cadre est fondé sur le processus suivant : (i) prétraitement du log, (ii) génération des recommandations candidates en commençant par trouver quelles sessions du log coïncident avec la session courante et ensuite, prédire ce que peut être la prochaine requête, (iii) ordonnancement des requêtes candidates en présentant à l'utilisateur la requête la plus pertinente en premier. Pour rappel, la figure 4.2 illustre ce fonctionnement.

Dans une optique de généricité, RecoOLAP propose actuellement différents paramètres : (i) Deux fonctions de prétraitement : l'une utilisant l'algorithme des k-médoïdes, l'autre étant l'identité, (ii) Deux fonctions de matching : l'une permettant d'obtenir les sessions qui coïncident avec la session courante et à quelle position de la session du log a lieu cette coïncidence et l'autre renvoyant la (ou les) session(s) la (les) plus proche(s) de la session courante au sens d'une distance entre sessions et, cinq représentants de session : successeur, dernier, union, intersection ou médoïde, (iii) Deux méthodes d'ordonnancement des requêtes : l'une basée sur la proximité entre la requête à recommander et la requête qui représente la session courante (au sens d'une distance entre requêtes), l'autre basée sur le profil de l'utilisateur.

Quatre combinaisons particulières ont été proposées afin d'illustrer l'applicabilité de nos algorithmes (voir [Neg09]) : ClusterH, ClusterSP, EdH et EdSP.

ClusterH : Le prétraitement est réalisé en utilisant l'algorithme des K-médoïdes (nous obtenons des classes de requêtes). La génération des recommandations est réalisée en cherchant des sous-séquences de la session courante prétraitée qui coïncident avec les sessions prétraitées du log puis en retournant le médoïde des classes qui succèdent à chaque sous-séquence similaire. L'ensemble des requêtes candidates ainsi obtenu est ensuite ordonné en fonction de la proximité des requêtes avec la dernière requête de la session courante. Les distances utilisées pour les calculs de similarité sont la distance de Hausdorff et la distance de Hamming.

ClusterSP : Les distances utilisées pour les calculs de similarité sont la distance de Hausdorff et la distance basée sur le plus court chemin dans un graphe. Les autres paramètres sont similaires

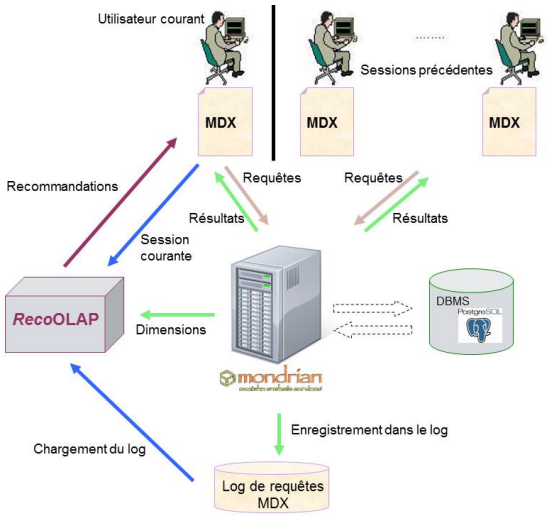

FIGURE 7.1 – Architecture du système RecoOLAP

à ceux de ClusterH.

EdH : Le log n'est pas prétraité. La génération des recommandations est réalisée en calculant la distance de Levenshtein entre les sessions de requêtes du log et la session courante (nous gardons celles qui coïncident) puis en retournant la dernière requête de chaque session candidate. L'ensemble des requêtes candidates ainsi obtenu est ensuite ordonné en fonction de la proximité des requêtes avec la dernière requête de la session courante. Les distances utilisées pour les calculs de similarité sont la distance de Hausdorff et la distance de Hamming.

EdSP : Les distances utilisées pour les calculs de similarité sont la distance de Hausdorff et la distance basée sur le plus court chemin dans un graphe. Les autres paramètres sont similaires à ceux de EdH.

Architecture

Dans cette section, nous présentons l'environnement *RecoOLAP* développé afin de valider et de rendre opérationnel notre système de recommandation de requêtes.

La figure 7.1 illustre l'architecture de notre système. Premièrement, chaque requête lancée par un utilisateur sur le cube de données, stocké dans le SGBD PostgreSQL [Pos13], via le serveur OLAP Mondrian [Pen17] obtient une réponse. Deuxièmement, les sessions de requêtes des utilisateurs précédents ont été enregistrées dans un log de sessions de requêtes. La session de requêtes de l'utilisateur courant : la session courante, ainsi que les sessions de requêtes du log sont chargées dans l'application de génération de recommandations : RecoOLAP. Pendant le processus de recommandation, RecoOLAP accède aux informations contenues dans le serveur OLAP Mondrian. Enfin, le système retourne à l'utilisateur l'ensemble ordonné des recommandations.

Notons que l'application pour la recommandation de requêtes MDX [1] est développée en Java sous JRE 1.6.0-27 avec Postgres 9.1.10 et Mondrian 3.3.0.14703.

1. *MultiDimensional Expressions* est un langage de requêtes OLAP, *de facto* standard.

7.1.2 MEMORAe-CRS

Présentation

OLAP est un environnement avec peu d'utilisateurs, nous avons voulu élargir nos travaux en nous intéressant à des domaines où le nombre d'utilisateurs est plus conséquent, notamment afin de traiter une matrice d'utilité plus dense. Ainsi, nous avons proposé MEMORAe-CRS.

MEMORAe-CRS est un système de recommandation d'experts (à partir de leurs compétences sur un sujet donné) dans l'environnement : MEMORAe [2]. Nous orchestrons un modèle d'actions, un modèle de compétence et une ontologie d'application pour formuler des recommandations sur la compétence des utilisateurs. Pour rappel, notre cadre est fondé sur le processus suivant : (i) prétraitement des traces d'interactions (en passant par une ontologie d'application), (ii) génération des recommandations candidates (via le TF-IDF ou la classification Bayésienne), (iii) ordonnancement des recommandations candidates par ordre décroissant des valeurs prédites. pour rappel, la figure 4.5 montre la structure de notre approche (où les parties colorées correspondent à celles de notre système générique de recommandation présenté dans la section 4.1. Tout d'abord, les actions des utilisateurs sont collectées et modélisées à partir d'une plateforme interactive (entrées). Après avoir été filtrées par le filtre de classification, nous obtenons des traces classifiées (prétraitement). Alternativement, nous appliquons un algorithme pour calculer un indice indiquant la corrélation entre les traces classifiées d'un utilisateur donné et d'un sujet donné (génération et ordonnancement des recommandations candidates). Ces valeurs peuvent conduire à des informations utiles qui sont présentées sous forme de recommandations personnalisées, soit à un groupe défini, c'est-à-dire un ensemble d'utilisateurs de la plateforme, soit à un utilisateur individuel.

Architecture

L'approche MEMORAe est une combinaison d'un modèle sémantique et d'une plateforme web partageant le même nom pour gérer des sources de connaissances hétérogènes dans une organisation :

- Dans l'approche MEMORAe, MEMORAe-core 2 est un modèle sémantique construit en utilisant OWL [3] et basé sur des standards du web sémantique (FOAF [4], SIOC [5], BIBO [6], ...). En ce qui concerne la typologie des ontologies, MEMORAe-core 2 contient une ontologie noyau qui représente la collaboration dans les organisations. Elle permet de modéliser des individus (et des groupes d'individus) et des ressources.
- La plateforme web e-MEMORAe est basée sur le modèle MEMORAe-core 2. La plateforme est développée à l'aide de technologies web 2.0 et permet la collaboration et le partage des ressources entre les membres d'une organisation.

Afin de pouvoir recommander des experts sur la base de leurs compétences, nous avons intégré l'ontologie MEMORAe-CRS. L'ontologie MEMORAe-CRS est un modèle d'ontologie pour la capitalisation des connaissances au sein du groupe qui collabore pour proposer un système de recommandation d'experts à base de compétences (CRS). Le modèle permet d'identifier différents types

2. http ://memorae.hds.utc.fr/

3. *Ontology Web Language* est un langage de représentation des connaissances construit sur le modèle de données de RDF

4. *Friend of a friend* est une ontologie RDF permettant de décrire des personnes et les relations quŠelles entretiennent entre elles

5. *Semantically-Interlinked Online Communities*

6. *BIBliographic Ontology* est une ontologie pour le web sémantique pour décrire des éléments bibliographiques comme les livres ou les magazines.

d'activités d'interaction dans la plateforme et de présenter et déduire la compétence des utilisateurs. L'ontologie MEMORAe-CRS est construite à partir de l'ontologie modulaire MEMORAe-core 2. La figure 7.2 illustre comment représenter les compétences dans l'ontologie MEMORAe-CRS.

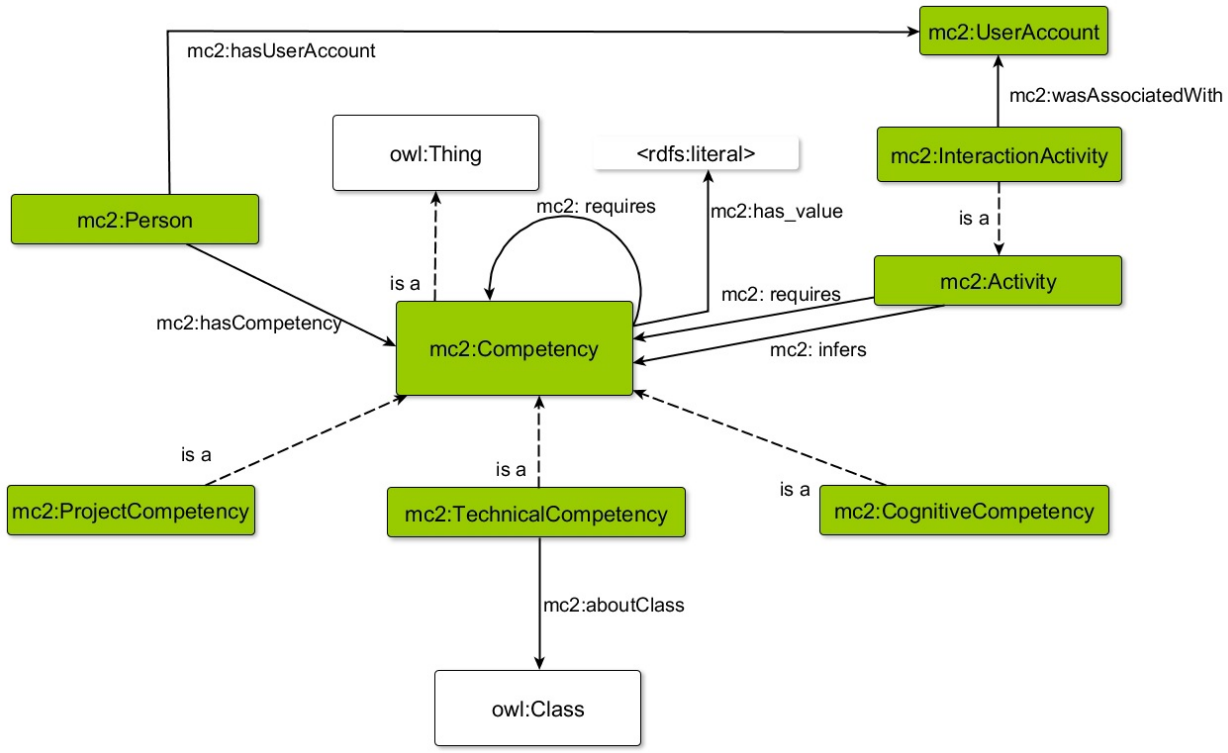

FIGURE 7.2 – Représentation des compétences dans l'ontologie MEMORAe-CRS

Interface

L'application web MEMORAe-CRS fait partie de l'approche MEMORAe. Cette plateforme est basée sur le modèle MEMORAe-CRS (voir la figure 7.5). La plateforme web e-MEMORAe prend en charge le partage des ressources modélisées dans le modèle MEMORAe-CRS (documents, weblinks, contacts, notes, annotations, agenda, ...). Toutes ces ressources sont indexées par les concepts d'une ontologie. Cette ontologie est présentée sous forme d'une carte sémantique au milieu de la page web. La carte sémantique définit une référence commune partagée entre tous les utilisateurs. Un utilisateur peut naviguer dans la carte pour afficher les ressources partagées dans les espaces de partage auxquels il a accès. Lorsque l'utilisateur clique sur un concept, ce concept devient le concept central. Ensuite, il/elle peut ouvrir un espace de partage pour voir différentes ressources indexées par ce concept central. En haut à droite, une boîte utilisateur (*user box*) est proposée où l'utilisateur courant peut ouvrir un ou plusieurs espaces de partage auxquels il/elle appartient. Les espaces de partage peuvent être ouverts en parallèle en naviguant dans la carte. Cette vue en parallèle est avantageuse car l'utilisateur peut voir les ressources indexées par le même concept central et partagées dans différents espaces de partage.

Les activités d'un utilisateur sur la plateforme e-MEMORAe sont stockées en temps réel sur le serveur et un tableau de bord est généré (tableau et graphiques) pour visualiser et résumer toutes

les activités de l'utilisateur courant indépendamment de l'espace de partage et des concepts. Le tableau de bord fournit un ensemble de filtres pour spécifier les activités à afficher.

L'analyse personnelle (*personal analysis*) est basée sur les traces de l'utilisateur courant. Un exemple d'analyse effectuée avec TF-IDF est présentée sur la figure 7.3 pour l'utilisateur « alaya.nar ». Le système de recommandation propose à cet utilisateur courant que son principal concept de compétence est « Attaque des clones » et qu'il devrait travailler d'avantage sur le concept « Action ».

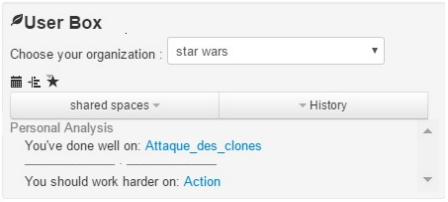

FIGURE 7.3 – MEMORAe-CRS : Exemple de recommandation - TF-IDF

Un exemple de recommandation basé sur la classification Bayésienne est présenté sur la figure 7.4. Cette recommandation est donnée comme une liste de probabilités selon lesquelles l'utilisateur de ce groupe est le plus compétent sur le concept ciblé. Cela aide les utilisateurs à trouver un collaborateur compétent au sein du groupe sur le concept ciblé.

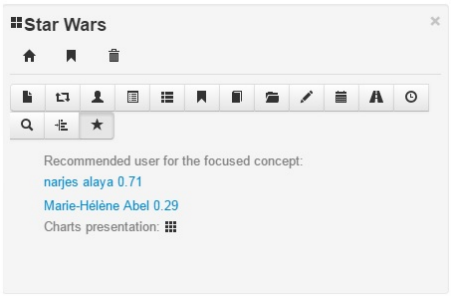

FIGURE 7.4 – MEMORAe-CRS : Exemple de recommandation - Classification Bayésienne

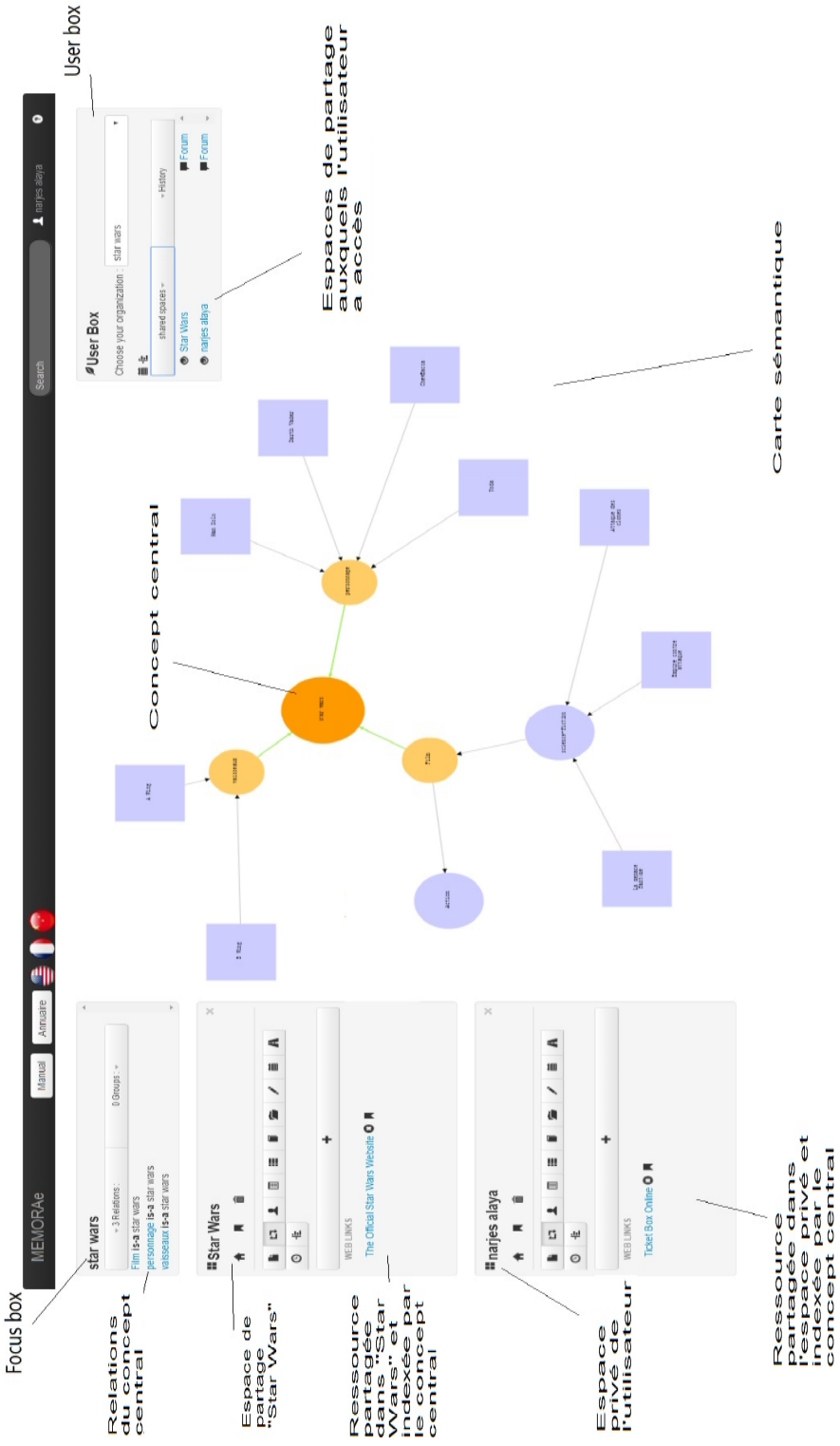

FIGURE 7.5 – Plateforme web e-MEMORAe

7.2 Projets et Collaborations

Comme indiqué dans les précédents chapitres, nos travaux ont donné lieu à différentes collaborations avec des entreprises et des laboratoires de recherche. Ces relations se sont traduites par des projets et des bourses CIFRE[7].

7.2.1 Vision synthétique

Afin de mettre en avant les caractéristiques des projets que nous avons menés durant ces dernières années[8], nous proposons le tableau synthétique suivant :

Intitulé	Période	Type	Partenaires	Thème	Rôle personnel
SNACMAN	2017	Ecole thématique	Télécom ParisTech (Institut Mines-Télécom), Université Paris-Est	Réseaux sociaux et Gestion de crises	Co-responsable et Participante
CIFRE *Ferdousi*	2016-2019	Contrat Bourse CIFRE	Société COHERIS	Systèmes de recommandation contextuels	Co-encadrante
ByFrançais	2015-2016	Contrat de recherche	Société ByFrançais	Baromètre économique	Responsable
CIFRE *Atif*	2014-2017	Contrat Bourse CIFRE	Société TalentSoft	Décisionnel et Système d'Information pour les Ressources Humaines	Co-encadrante
H2H	2013-2016	Contrat de recherche	Société H2H mobility	Mise en relation de personnes	Responsable
ARAMIS	2012-2015	Contrat de recherche	Société ARAMIS Auto	Similarité entre véhicules	Responsable

7.2.2 Présentation des projets et/ou Collaborations

Nous expliquons ces différents projets dans les paragraphes suivants.

SNACMAN
Dans le cadre de la gestion de crises, nous ne souhaitons pas nous limiter aux systèmes de recommandation mais participer à la dynamisation d'un nouvel axe de recherche. Ainsi, l'école thématique pluridisciplinaire « Réseaux sociaux et Gestion de crises » émerge de la rencontre de chercheurs français de champs disciplinaires complémentaires appartenant à la communauté scientifique internationale ISCRAM « *Information System for Crisis Response and Management* ». Elle s'inscrit dans la volonté de créer une communauté de chercheurs pluridisciplinaire en France sur la thématique de la gestion de crises et des nouveaux médias (spécifiquement ici, les réseaux sociaux) et de positionner ainsi les compétences de nos équipes françaises dans une perspective internationale sur cette thématique (car peu visibles jusqu'à présent). L'école s'appuie en effet sur la communauté

7. Conventions Industrielles de Formation par la REcherche
8. D'autres projets ont été proposés (COST (*European Cooperation in Science and Technology*), ANR (*Agence Nationale de la Recherche*), PEPS (*Projets Exploratoires Premier Soutien*), ...) mais n'ont pas été retenus.

ISCRAM dont la conférence annuelle a eu lieu la semaine suivant l'école thématique (du 21 au 24 mai 2017) à l'Ecole des Mines d'Albi-Carmaux. Organisée en étroite collaboration avec ISCRAM, l'école thématique est en anglais, ouverte aux apprenants étrangers, et mobilise nos collègues experts du domaine à l'international.

Cette école thématique est en adéquation avec nos travaux sur les systèmes d'alertes précoces, présentés dans les sections 4.2.2.2, 5.1.2 et 6.3.3.3.

CIFRE *Ferdousi*

Afin de compléter et valider nos travaux sur la contextualisation des systèmes de recommandation, cette thèse CIFRE traite des systèmes de recommandation contextuels, c'est-à-dire intégrer à un système de recommandation des informations contextuelles pour améliorer les recommandations. Dans le cadre du programme de recherche et d'innovation porté par le groupe SEB [9], Coheris [10] (éditeur de solutions CRM [11] et *Analytics*) a récemment remporté le Trophée de l'innovation Big Data Paris 2015 pour son système de recommandation temps réel. Dans le cadre de cette thèse, Coheris souhaite enrichir son premier système de recommandation en intégrant une dimension contextuelle.

L'originalité de cette recherche consiste à appliquer des techniques de recherche d'information, de filtrage d'information, d'apprentissage automatique, entre autres, au domaine en pleine expansion et encore peu exploré des systèmes de recommandations contextuels. Les algorithmes devront être conçus pour prendre en compte des données volumineuses et proposer des recommandations pertinentes.

Cette collaboration nous permet d'enrichir nos travaux sur les systèmes de recommandation et la prise en compte du contexte, présentés dans la section 6.3.

ByFrançais

Afin d'enrichir et valider nos travaux sur les mesures de similarité et les techniques d'agrégation, nous avons collaboré avec la start-up ByFrançais [12]. Dans le but de mesurer l'impact sur l'économie et l'emploi en France de l'achat d'un bien de consommation à partir des informations disponibles auprès du grand public : ByFrançais a développé un baromètre économique (basé sur des indicateurs). Dans le cadre de leur projet de calcul et d'agrégation des indicateurs, nous avons participé (i) à l'analyse du besoin (notamment l'évaluation du coût du travail) et de l'existant pour la mise en place d'un baromètre permettant l'évaluation de l'impact d'un achat en France sur l'emploi et l'économie (données nécessaires, disponibilité/existence des données nécessaires, interopérabilité des données), (ii) à l'architecture de la base de données (schéma de base de données adéquate et stratégies d'alimentation) et (iii) à la collecte et l'intégration des données.

Cette collaboration nous a permis d'améliorer nos connaissances sur les mesures de similarité et les techniques d'agrégation, en rapport avec les travaux présentés dans les sections 3.4 et 5.1.

CIFRE *Atif*

Afin de valider nos travaux sur la prise en compte des connaissances, nous avons collaboré avec la société TalentSoft [13] (spécialiste de la gestion des ressources humaines via le Cloud) dans le cadre de cette thèse CIFRE. Le développement des systèmes d'information numériques et plus particulièrement les systèmes interactifs d'aide à la décision (SIAD) orientés données (Application

9. http ://www.seb.fr/
10. https ://www.coheris.com/
11. *Customer Relationship Management*
12. http ://www.byfrancais.com/
13. http ://www.talentsoft.fr/

Analytics) rencontre des échecs divers. La plupart des études montrent que ces échecs de projets de développement de SIAD relève de la phase d'analyse des besoins et des exigences. Les exigences qu'un système doit satisfaire sont insuffisamment définies à partir de besoins réels des utilisateurs finaux. D'un point de vue théorique, l'analyse de l'état de l'art, mais également du contexte industriel particulier, conduit donc à porter une attention particulière à cette phase et à élaborer une approche collaborative d'analyse des besoins et des exigences dirigée par les problèmes. Un système d'aide à la décision est avant tout un système d'aide à la résolution de problèmes et le développement de ce type d'artefact ne peut donc se faire sans avoir convenablement identifié en amont les problèmes de décision auxquels font face les utilisateurs décideurs, afin d'en déduire les exigences et le type de SIAD. Cette approche, par un renversement de la primauté implicite de la solution technique par rapport à la typologie des problèmes de décision, a été explicitée et mise en œuvre pour le développement d'une application *Analytics* qui a permis d'atteindre l'objectif attendu : un système efficace et qui satisfasse d'un triple point de vue technique, fonctionnel et ergonomique, ses différents utilisateurs finaux.

La thèse intitulée : « Une approche collaborative d'analyse des besoins et des exigences dirigée par les problèmes : le cas de développement d'une application Analytics RH » a été soutenue le 07/07/2017.

Cette collaboration nous a permis, entre autres, d'enrichir nos travaux sur la prise en compte des connaissances, présentés dans la section 5.1.1.

H2H

Afin de décider quelles mesures de similarité sont adaptées à quelle application en particulier, notamment en *Text mining*, nous avons collaboré avec la société H2H Mobility [14]. Elle développe des applications sur téléphones et smartphones dont l'objectif principal est de mettre en relation des utilisateurs via du contenu textuel (travaux basés sur des mesures de similarité entre textes). Notre contribution dans le cadre de ce projet a consisté à donner des pistes sur des mesures de similarité entre documents afin de prendre la meilleure décision de développement possible.

Cette collaboration nous a permis d'améliorer nos connaissances sur les mesures de similarité présentées dans la section 3.4.

ARAMIS

Afin de compléter et valider nos travaux sur la contextualisation dans les systèmes de recommandation, nous avons collaboré avec la société ARAMIS Auto [15] (vendeur de véhicules d'occasion). Forte d'une croissance soutenue depuis sa création, cette société se trouve désormais confrontée à une concurrence nouvelle. Pour faire face à ce nouveau défi ARAMIS a pris le parti de renforcer ses investissements en R&D. Cette ambition implique de maitriser tout le processus de détermination du meilleur prix de rachat des voitures tout en garantissant lors de la revente une marge préétablie. En d'autres termes, le challenge vise à optimiser la valeur de rachat et de revente des voitures.

La collaboration de recherche avec ARAMIS poursuit les enjeux centraux suivants : (i) déterminer pour 90% à 100% des modèles de voitures mis en vente sur LaCentrale.fr et Leboncoin.fr, la mesure de similarité entre deux voitures résultant des annonces publiées, (ii) établir le *Market Day Supply* (approvisionnement quotidien) : par type de voitures, nombre de voitures sur le marché rapporté au nombre de voitures vendues par jour, (iii) optimiser les modèles de données et de

14. http ://h2h-mobility.com/
15. https ://www.aramisauto.com/

traitements afin de garantir des temps d'exécution compatibles avec les exigences du processus de cotation.

Cette collaboration nous a permis d'enrichir nos travaux sur les systèmes de recommandation et la prise en compte du contexte, présentés dans la section 6.3.3.2.

7.3 Encadrements et Publications

7.3.1 Encadrements

Depuis 2011, j'ai co-encadré quatre thèses (dont deux ont été soutenues).

Années	Etudiant	Domaine	Type	Rôle personnel
2016-	Zahra VAHIDI FERDOUSI	Systèmes de recommandation contextuels	Thèse avec Bourse CIFRE (Société COHERIS)	Co-encadrement (80%)
2015-	Maude ARRU	Systèmes d'alertes précoces	Thèse avec allocation ministérielle	Co-encadrement (60%)
2014-2017	Lynda ATIF	Décisionnel et Systèmes d'Information pour les Ressources Humaines	Thèse avec Bourse CIFRE (Société TalentSoft)	Co-encadrement (20%)
2013-2016	Ning WANG	Recommandation d'experts dans les environnements de travail collaboratif	Thèse (Financement : Chine) - Université de Technologie de Compiègne	Co-encadrement (33%)

Depuis 2011, j'ai également encadré ou co-encadré trois stages de Recherche : un étudiant en $1^{ère}$ année de Doctorat (Université de Sfax, Tunisie) ainsi que deux étudiants en Master 2 Recherche (Université Paris-Dauphine) :

Année	Etudiant	Domaine	Niveau	Durée	Rôle personnel
2015-2016	Hamza SELLAMI	Système de recommandation contextuel pour OLAP	M2	4 mois	Encadrement (100%)
2013	Jalel Eddine JHAJLAOUI	Similarité de requêtes OLAP	$1^{ère}$ année de Doctorat	2 mois	Encadrement (100%)
2011-2012	Jaroslaw SZYMCZAK	Données, Informations et Connaissances	M2	5 mois	Co-encadrement (33%)

7.3.2 Publications

Dans cette section, mes publications sont classées par année et par type. Pour chacune sont précisés la référence bibliographique, le type [16] et les thématiques.

Année	Référence	Type	Thématiques
2017	[AN17]	CI	Systèmes d'alertes précoces
	[ANRS17a]	CI	Systèmes d'alertes précoces
	[MNRS17]	CI	Systèmes d'Information et de Connaissance
	[Neg17a]	CI	Systèmes de recommandation contextuels, OLAP
	[BCCN17]	CI	Systèmes de recommandation, Plateformes d'apprentissage
	[WABN17]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[Neg17b]	CN	Systèmes de recommandation contextuels, OLAP
	[ANRS17b]	CN	Systèmes d'alertes précoces
	[FNC17]	P	Systèmes de recommandation contextuels
2016	[STB+16]	RI	Profils utilisateurs
	[BN16]	RI	Evaluation des systèmes de recommandation, OLAP
	[WABN16b]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[WABN16a]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[ANRG16]	CI	Systèmes d'alertes précoces
	[AMN16]	CI	Systèmes d'alertes précoces
	[ANRS16]	CI	Villes intelligentes
	[DNR16]	CI	Villes intelligentes
2015	[Neg15b]	L	Systèmes de recommandation
	[Neg15a]	L	Systèmes de recommandation
	[AAG+15]	CI	Systèmes d'Information et de Connaissance
	[WABN15b]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[WABN15c]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[WABN15a]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[CMN15]	CI	Systèmes d'alertes précoces
	[NRG15]	CI	Villes intelligentes
	[AN15]	CN	Villes intelligentes
	[dN15]	Ed	Numéro spécial
2014	[NRS15a]	Ch	Systèmes de recommandation, Villes intelligentes
	[NRS15b]	Ch	Villes intelligentes
	[AMNR14]	RI	Systèmes d'Information et de Connaissance
	[WABN14]	CI	Systèmes de recommandation, Env. de travail collaboratif
	[CMN14]	CI	Systèmes d'alertes précoces
	[Neg14]	CN	Systèmes de recommandation, OLAP
	[BdN14]	Ed	Actes

16. L pour livre, Ch pour chapitre de livre, RI pour revue internationale, RN pour revue nationale, CI pour conférence internationale, CN pour conférence nationale, T pour rapport technique, P pour poster et Ed pour édition.

2013	[MNAV13]	RI	Réseaux sociaux
	[AMN11b]	RI	Systèmes de recommandation, Organisation de log, OLAP
	[Neg13b]	CI	Systèmes de recommandation, Systèmes d'alertes précoces
	[AGNR13b]	CI	Systèmes d'Information et de Connaissance
	[ANRS13]	CI	Systèmes d'Information et de Connaissance
	[NRTT13a]	CI	Systèmes de recommandation, Démarrage à froid, OLAP
	[AGNR13a]	CN	Systèmes d'Information et de Connaissance
	[NRTT13b]	CN	Systèmes de recommandation, Démarrage à froid, OLAP
	[Neg13a]	T	Mesures de similarité

2012	[ADG+12]	CN	Systèmes d'Information et de Connaissance

2011	[GMNS11]	RI	Systèmes de recommandation, OLAP
	[Neg11]	RN	Systèmes de recommandation, OLAP
	[NMV11a]	CI	Réseaux sociaux
	[MN11]	CN	Systèmes de recommandation, OLAP
	[AMN11a]	CN	Systèmes de recommandation, Organisation de log, OLAP
	[NMV11b]	T	Réseaux sociaux

2010	[AMN10a]	CI	Systèmes de recommandation, Organisation de log, OLAP
	[CMN10]	CN	Systèmes de recommandation, Organisation de log, OLAP
	[AMN10b]	T	Systèmes de recommandation, Organisation de log, OLAP

7.4 Bilan et Synthèse

Dans ce chapitre, nous avons présenté les différentes productions qui nous ont permis de mettre en œuvre et valider les concepts, modèles et principes proposés dans les chapitres précédents.

Dans un premier temps, nous avons présenté deux systèmes de recommandation. RecoOLAP permet de recommander des requêtes OLAP à un analyste qui explore un cube de données. MEMORAe-CRS, quant à lui, permet de recommander des experts dans un environnement de travail collaboratif. Ces deux prototypes sont écrit principalement en Java et comportent plus de 20 000 lignes de code chacun.

Nos travaux se sont intégrés dans le cadre de projets. Ces derniers nous ont permis de collaborer aussi bien avec d'autres laboratoires de recherche qu'avec des entreprises. Ces projets ont permis de donner un cadre applicatif à certaines thèses que j'ai co-encadrées. Comme indiqué dans la section 7.2, ces projets m'ont également permis d'assurer différentes fonctions : responsable, co-responsable, participante ou encore co-encadrante.

D'un point de vue quantitatif, j'ai encadré ou co-encadré quatre thèses, deux Master 2 Recherche et un stagiaire en $1^{ère}$ année de thèse.

Enfin, ces travaux se sont concrétisés par un ensemble de publications que nous pouvons lister comme suit :

- 2 livres,
- 2 chapitres de livre,
- 6 articles dans des revues internationales (dont IJDWM [17] et IJBIS [18]),
- 1 article dans une revue nationale,

17. International Journal of Data Warehousing and Mining
18. International Journal of Business Information Systems

Thèmes	Projets et Collaborations	Co-encadrements	Publications
Systèmes de recommandation			2 L
• Entrepôts de données/OLAP			1 RI, 1 RN, 2 CN
• Plateformes d'apprentissage			1 CI
• Env. de travail collaboratif		1 thèse	7 CI
• Systèmes d'alertes précoces	SNACMAN	1 thèse	1 CI
• Villes intelligentes			1 Ch, 2 CI
Evaluation			
• Prise en compte des connaissances	ARAMIS, CIFRE *Atif*	1 thèse CIFRE	1 RI, 2 CI, 1 CN
• Satisfaction des utilisateurs			1 RI
• Perception du système			2 CI
Amélioration			
• Réorganisation de logs			1 RI, 1 CI, 2 CN, 1 T
• Démarrage à froid			1 CI, 1 CN
• Prise en compte du contexte	CIFRE *Ferdousi*	1 thèse CIFRE, 1 M2R	4 CI, 1 CN, 1 P
Synthèse	4 projets	4 thèses, 1 M2R	2 L, 1 Ch, 4 RI, 1 RN, 7 CN, 21 CI, 1 T, 1 P

TABLE 7.1 – Tableau de synthèse des thématiques de recherche, des projets et des publications (en rapport avec les systèmes de recommandation)

- 26 articles dans des conférences internationales (dont ASONAM [19], ISCRAM [20], HICSS [21]),
- 10 articles dans des conférences nationales (dont INFORSID [22], EDA [23]),
- 1 poster,
- 3 rapports techniques,
- 2 ouvrages édités (des actes et un numéro spécial de revue internationale).

Il est à noter que, de manière générale, notre volonté a été de faire (re)connaitre nos travaux au niveau international mais aussi national.

La synthèse de nos différents travaux en rapport avec les systèmes de recommandation (les thématiques de recherche correspondent à celles abordées dans ce mémoire) est présentée dans le tableau 7.1.

Finalement, la figure 7.6 résume le cheminement temporel de nos travaux de recherche.

19. International conference on Advances in Social Network Analysis and Mining
20. International conference for Information Systems for Crisis Response and Management
21. Hawaii International Conference on System Sciences
22. INFormatique des ORganisations et Systèmes d'Information et de Décision
23. Entrepôts de Données et Analyse en ligne

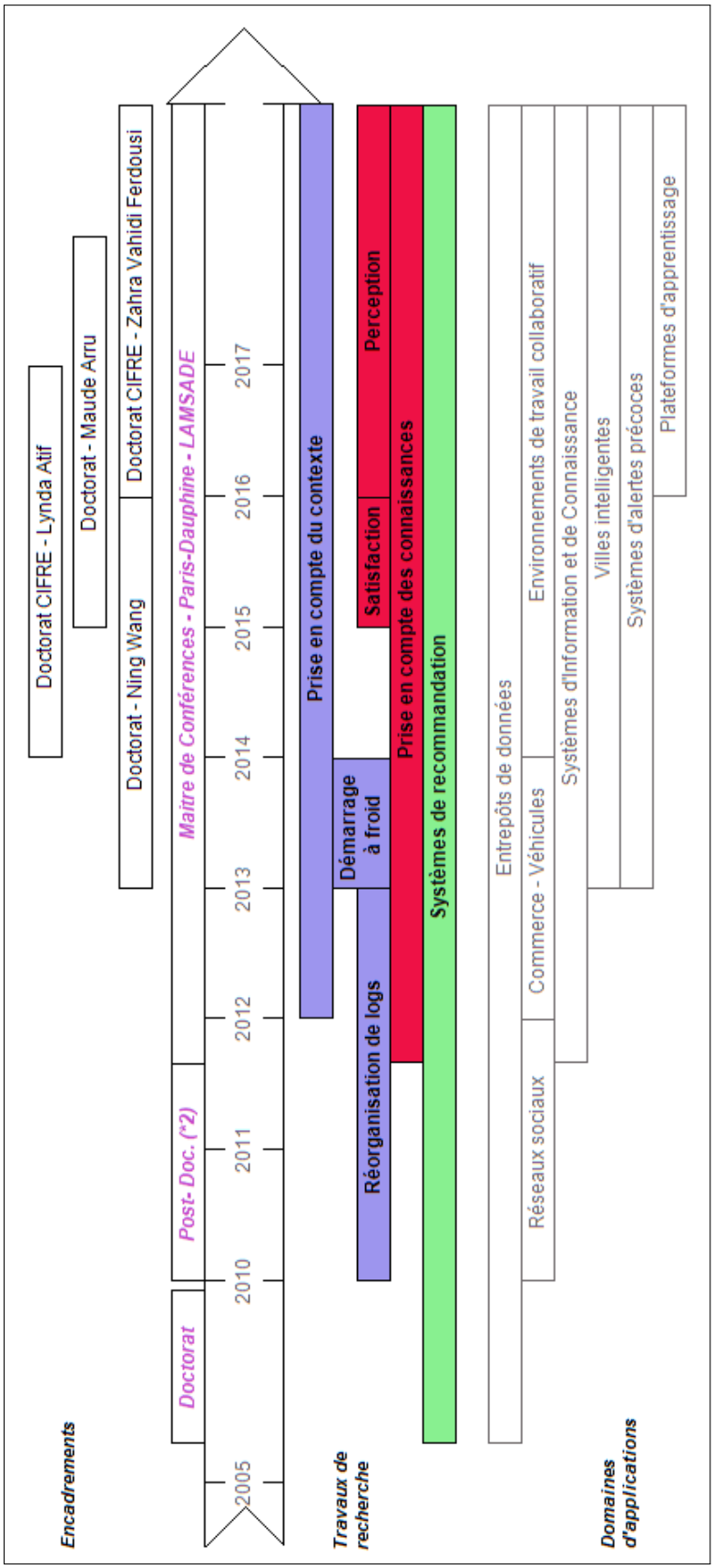

FIGURE 7.6 – Cheminement de nos travaux de recherche

Chapitre 8

Conclusion et Perspectives générales

Sommaire

8.1 Bilan de nos travaux 147
8.2 Perspectives 151

8.1 Bilan de nos travaux

Les travaux que nous avons menés durant ces dernières années s'articulent autour de l'extraction et de l'analyse de données issues de sources hétérogènes pour les rendre facilement accessibles et exploitables par les utilisateurs/décideurs. Avec l'avènement du web et les évolutions technologiques, entre autres, la masse de données à exploiter/analyser est devenue très volumineuse. Si bien qu'il est devenu difficile de savoir quelles sont les données à rechercher et où les trouver. Des techniques informatiques pour faciliter cette recherche ainsi que l'extraction des informations pertinentes existent. Celle sur laquelle nous nous sommes concentrés dans le cadre de nos travaux est la recommandation. Il s'agit de guider l'utilisateur dans son exploration des données afin qu'il trouve des informations pertinentes.

Les systèmes de recommandation ont été étudiés, depuis le début des années 90, dans différents domaines comme le web, le e-commerce et bien d'autres. La recommandation est devenue un axe de recherche en plein essor : de nombreux systèmes que nous utilisons au quotidien proposent des recommandations (de plus ou moins bonne qualité). Malheureusement, chaque système est différent, spécifique à l'application et aux utilisateurs pour lesquels il a été conçu. Cette spécialisation va à l'encontre de la qualité logicielle qui prône, entre autres, la généricité, la portabilité et l'interopérabilité. De plus, l'évaluation des systèmes de recommandation reste encore trop souvent algorithmique (objective) oubliant de placer l'utilisateur (à satisfaire) au cœur de l'évaluation. Enfin, le volume et la variété de données/informations utilisables par les systèmes de recommandation sont devenus importants, or, le volume et la variété ne sont rien sans la qualité. Cela limite l'évolution et l'amélioration des systèmes de recommandation.

Nos travaux de recherche visent à répondre à ces manques. Plus précisément, les travaux présentés dans ce mémoire apportent des modèles, méthodes et outils pour le développement de systèmes de recommandation génériques pertinents et performants. Nous avons identifié trois problématiques. La première vise à concevoir un système de recommandation de qualité (au sens de la qualité logicielle) quel que soit le domaine d'application, la seconde se centre sur l'évaluation des systèmes de recommandation et la troisième consiste en leur amélioration.

Système générique de recommandation
Les systèmes de recommandation sont, pour la plupart, des systèmes dédiés, spécifiques à une application donnée. Cependant, les systèmes de recommandation sont des systèmes informatisés qui sont régis par les principes de l'ingénierie et de la qualité logicielle. Or, la généricité, selon laquelle un système réutilisable/adaptable a beaucoup plus de valeur qu'un système dédié, est l'un des sept principes de l'ingénierie logicielle [GJM02]. Ainsi, un système de recommandation devrait être générique et utilisé dans différents domaines d'applications.

Bien que de nombreux systèmes de recommandation existent, il n'y avait pas, jusqu'à ce jour, de proposition pour un système générique instanciable dans différents domaines. L'approche que nous proposons est générique dans le sens où elle n'impose pas une méthode spécifique de calcul. Au lieu de cela, les actions à réaliser sont laissées comme paramètres de l'approche, qui, de ce fait, peut être instanciée de différentes manières afin de changer la méthode de calcul des recommandations (en changeant ces paramètres, la manière dont les recommandations sont calculées change). Notre approche repose sur un processus en trois étapes :

1. prétraitement du log et/ou de la matrice d'utilité,
2. génération des recommandations candidates en cherchant les éléments recommandables qui coïncident le mieux avec les attentes/besoins de l'utilisateur,
3. ordonnancement des recommandations candidates afin de présenter les recommandations à l'utilisateur par ordre d'intérêt.

Aussi, notre proposition est en rupture par rapport à tous les systèmes de recommandation qui sont « sur-spécialisés ». Nous sommes allés plus loin qu'une simple formulation théorique en instanciant notre système générique dans cinq domaines (les entrepôts de données, les environnements de travail collaboratif, l'apprentissage en ligne, les villes intelligentes et les systèmes d'alertes précoces) avec des méthodes de calculs différentes.

Ces domaines ont des objectifs et des enjeux très différents. Alors que le système de recommandation suggère des supports de cours en apprentissage en ligne ou des requêtes dans les entrepôts de données, il recommande des utilisateurs dans les environnements de travail collaboratif. Ces recommandations s'adressent à des utilisateurs du système de recommandation ayant leurs propres intérêts/besoins. Par contre, le système de recommandation suggère des actions/initiatives à mettre en place dans les villes intelligentes ou pour les systèmes d'alertes précoces, à des décideurs dont les décisions auront un impact sur les populations. Nous avons ainsi identifié deux grandes « familles » de recommandations :

- les recommandations individuelles qui s'adressent à un utilisateur (ou groupe d'utilisateur) et qui sont utilisées dans un cadre restreint à l'utilisateur (ou au groupe d'utilisateurs), comme par exemple la recommandation de ressources pour un apprenant dans les plateformes d'apprentissage en ligne, et,
- les recommandations à visée publique qui s'adressent à un décideur (ou groupe de décideurs) et qui sont utilisées pour prendre des décisions qui auront un impact sur les populations, comme par exemple la recommandation d'actions à mettre en place lors du déclenchement d'une alerte en gestion de crises.

Finalement, les cinq instanciations de notre modèle générique fonctionnent. Nous avons d'ailleurs implémenté deux d'entre elles dans le cadre des recommandations individuelles [1] : RecoOLAP pour la recommandation de requêtes OLAP dans les entrepôts de données et MEMORAe-CRS pour la

1. Dans le cadre des recommandations à visée publique, l'accès aux données et les enjeux liés à la prise de décision ne nous ont pas encore permis d'implémenter nos propositions.

Instanciation	Entrepôts de données	Env. de travail collaboratif	Apprentissage en ligne	Villes intelligentes	Systèmes d'alertes précoces
Besoin	Explorer un cube de données	Trouver des experts sur un sujet donné	Combler des lacunes	Atteindre un objectif donné	Mettre des actions en place lors du déclenchement d'une alerte
Nature	Élément : requête	Utilisateur : expert	Élément : ressource	Élément : action	Élément : action
Classe	individuelle	individuelle	individuelle	à visée publique	à visée publique
Recommandation	Ensemble ordonné de requêtes OLAP	Ensemble ordonné d'experts sur un sujet donné	Ensemble ordonné de ressources	Ensemble ordonné d'actions	Ensemble ordonné d'actions
Prototype	*RecoOLAP*	*MEMORAe-CRS*			

TABLE 8.1 – Synthèse des instanciations de notre système générique de recommandation

recommandation d'experts dans les environnements de travail collaboratif (l'implémentation dans le domaine de l'apprentissage en ligne est en cours).

Le tableau 8.1 synthétise nos cinq propositions.

Evaluer un système de recommandation

Les systèmes et techniques évoluent au cours du temps en essayant d'être toujours plus proches des attentes/besoins des utilisateurs. Pour cela, il est nécessaire d'évaluer les systèmes de recommandation et les recommandations elles-mêmes pour vérifier s'ils sont pertinents/performants pour l'utilisateur. En informatique, la majorité des mesures utilisées sont des mesures d'évaluation objective. Malheureusement, la plupart de ces mesures intègrent dans leurs calculs l'évaluation des recommandations par les utilisateurs sans indiquer comment cette évaluation subjective est obtenue. L'évaluation subjective a été très peu traitée dans la littérature de par la difficulté de sa mise en place et/ou de celle à impliquer les utilisateurs. Or, [KWG+12] prône l'importance de la prise en compte de l'expérience de l'utilisateur. L'intérêt de l'évaluation subjective réside dans le fait de positionner l'utilisateur (pour qui les recommandations doivent être pertinentes) au cœur de l'évaluation afin d'évaluer la pertinence des recommandation du point de vue de l'utilisateur (et non d'un point de vue algorithmique).

Afin de s'assurer de la pertinence des recommandations du point de vue de l'utilisateur, nous avons donc proposé deux méthodes pour évaluer subjectivement les systèmes de recommandation. La première prend en compte les connaissances des utilisateurs via un modèle d'étude de la maturité et de la connaissance où la prise en compte de la connaissance est un critère de maturité du système de recommandation. La seconde mesure la perception qu'ont les utilisateurs du système de recommandation via un indicateur de perception. Cet indicateur permet de synthétiser les composants

subjectifs de la perception (qu'ont les utilisateurs du système de recommandation) pour indiquer si le système est bien ou mal perçu. Nous avons également proposé un cadre d'évaluation subjective des recommandations en mesurant la satisfaction des utilisateurs via une analyse de la pertinence sémantique. Nos mesures ont été validées sur des données et des utilisateurs réels. Elles montrent comment évaluer les systèmes de recommandation et/ou les recommandations en se concentrant sur l'utilisateur, qui est le cœur du système (celui qui doit être satisfait via des recommandations pertinentes).

Améliorer un système de recommandation

Malgré de bonnes performances, les systèmes de recommandation peuvent être améliorés pour tenter d'être encore plus près des attentes/besoins des utilisateurs. Dans ce but, nous avons identifié trois axes d'amélioration possibles.

Le premier consiste en l'amélioration des données d'entrée. En effet, les systèmes de recommandation qui utilisent des logs/journaux sont souvent confrontés à des fichiers difficilement exploitables qui ne sont pas initialement conçus pour contribuer à la suggestion de recommandations. Ce sont des fichiers plats, non structurés, dans lesquels il est difficile de trouver des informations. Une meilleure organisation améliore le traitement du log et fait émerger des données/informations complémentaires. De ce fait, nous avons proposé deux approches pour les structurer et ainsi faciliter le traitement par le système de recommandation : (i) la structuration sous forme de graphe (plus facilement exploitable qu'un fichier log plat classique et permettant de modéliser les données et certains « liens » entre elles) et (ii) la structuration sous forme de résumé (mieux adapté à de grands volumes de données et permettant de garder un même format - un log de requêtes par exemple sera résumé par un autre log de requêtes).

Le second axe d'amélioration concerne le fameux problème du démarrage à froid, bien connu dans les systèmes de recommandation, lors de l'ajout de nouveaux éléments ou de nouveaux utilisateurs. Nous proposons une approche capable de pallier une double problématique (nouvel utilisateur et nouvel élément). En OLAP, cette double nouveauté apparaît lorsqu'un nouveau cube (nouvel item) est interrogé par un nouveau décideur (ou un décideur ayant changé de fonction - nouvel utilisateur). Notre approche est prédictive dans le sens où elle doit recommander des éléments qui n'ont jamais été évalués. Elle est basée sur l'analyse du comportement du décideur durant son utilisation courante et sur le comportement de l'utilisateur lors de ses utilisations passées avant l'ajout des nouveaux éléments/utilisateurs. Cette approche prédictive relève quatre défis : (i) extraire le comportement des utilisateurs, (ii) prévoir les opérations permettant de passer d'un comportement à un autre, (iii) sélectionner les opérations pertinentes et (iv) classer les recommandations candidates. Cette approche permettant de gérer le problème du démarrage à froid ne doit être utilisée que jusqu'à ce que le système recueille suffisamment de données pour revenir à un système de recommandation standard.

Enfin, le troisième axe consiste en l'ajout de données/sources externes et plus particulièrement, en prenant en compte le contexte de l'utilisateur. En effet, le contexte influence les intérêts et les décisions des utilisateurs [AT08]. De même, la performance des systèmes de recommandation est impactée par le contexte qui est flou et omniprésent [BB05, AT08]. Intégrer les informations contextuelles dans les systèmes de recommandation, qui deviennent des systèmes de recommandation contextuels, permettra, de notre point de vue, d'avoir des recommandations plus en adéquation avec les attentes/besoins des utilisateurs et de ce fait, plus pertinentes pour eux. Ainsi, à partir d'une étude bibliographique, nous avons proposé une catégorisation du contexte, selon trois catégories et dix unités, qui recoupe, enrichit et catégorise les travaux existants dans différents domaines. Puis nous avons présenté deux modélisations que nous avons instanciées pour valider leur pertinence.

8.2 Perspectives

Dans ce mémoire, nous abordons le paradoxe des problématiques actuelles des systèmes de recommandation, à savoir : avoir des systèmes indépendants du domaine d'application (*domain-independent*) [RAS07, Bal08, KSU09, RRSK11] et plus centrés sur l'utilisateur (*user-centred*) [RRSK11, Wak12]. Nos travaux suivent cette direction.

Nous proposons cinq axes d'études complémentaires à nos travaux : à court terme, l'explication des recommandations et la prise en compte de la sérialité des éléments/utilisateurs, à moyen terme, les systèmes de recommandation contextuels et les systèmes de recommandation multi-domaines et à long terme, la mise en œuvre d'une plateforme générique.

Explication des recommandations
L'interaction du système de recommandation avec l'utilisateur est importante. En général, les utilisateurs souhaitent avoir un contrôle sur les recommandations faites et pouvoir indiquer si une recommandation ne leur convient pas. Cette démarche s'inscrit dans la recherche de systèmes plus centrés sur l'utilisateur. De plus, le fait de donner une explication personnalisée à l'utilisateur peut également nous aider à gagner sa confiance [HKR00, TM07, RRSK11]. En effet, réussir à indiquer la raison pour laquelle une recommandation est proposée à l'utilisateur peut contribuer à améliorer la confiance qu'il a du système mais aussi, si l'utilisateur peut interagir avec le système de recommandation, améliorer le système lui-même en mettant en adéquation les raisons d'une telle recommandation et les raisons d'accord ou de refus de celle-ci par l'utilisateur. Des travaux se sont déjà intéressés à l'explication des recommandations [HKR00, BM05, Tin09]. Cependant, nombre d'entre eux se limitent à afficher des valeurs numériques ou à donner une explication du type « *Ceux qui ont acheté A ont aussi acheté B* ». Etre capable d'expliquer la recommandation soulève des problèmes tels qu'expliquer comment fonctionne le système de recommandation, permettre aux utilisateurs d'indiquer au système de recommandation que la recommandation ne convient pas, convaincre les utilisateurs de suivre la recommandation, améliorer la facilité d'utilisation. Nos travaux dans les environnements de travail collaboratif [Wan16] qui affichent les valeurs prédites qui ont menées à la proposition de telle recommandation étaient un premier pas dans ce sens. Une idée plus élaborée consisterait à coupler un système de votes des utilisateurs (pour indiquer la pertinence de chaque recommandation proposée) avec une phrase courte personnalisée indiquant la raison pour laquelle chaque recommandation est proposée, par exemple, « *Ce livre appartient à votre genre préféré : Thriller/Policier, il a obtenu une note moyenne de 4/5 auprès des autres utilisateurs et il est disponible à la librairie de votre quartier* ».

Prise en compte de la sérialité des éléments/utilisateurs
Les systèmes de recommandation ont pour fondement la similarité entre éléments ou utilisateurs afin de calculer les recommandations. Malheureusement, cela engendre des limitations. Par exemple, pour les approches à base de contenu, l'un des inconvénients lié est l'effet *Port-Folio* [Bur02] (non-diversité). En effet, à force de recommander des éléments similaires, l'espace de recherche des recommandations se restreint. Tout évolue au cours du temps et les recommandations doivent nous permettre d'avancer. Proposer des recommandations, non pas à partir d'une « simple » similarité avec un élément courant ou passé, mais dans la continuité de celui-ci est une idée à explorer. Cela suppose que les éléments recommandables ont un lien et qu'ils peuvent être plus ou moins ordonnés les uns à la suite des autres. Par exemple, recommander des ressources pédagogiques à un apprenant dans une plateforme d'apprentissage qui ne sont pas uniquement similaires à celles qu'il est

en train d'étudier mais aussi dans la continuité des notions à comprendre serait un plus. Nous avons déjà esquissé cette notion de continuité dans nos travaux sur la recommandation de requêtes OLAP lors de la recherche du représentant de session [Neg09]. Une autre idée serait de représenter les éléments/utilisateurs recommandables selon un graphe orienté pondéré où les nœuds sont les éléments/utilisateurs et où le sens des arêtes indique la sérialité des éléments/utilisateurs selon la distance/similarité donnée par le poids de l'arête.

Systèmes de recommandation contextuels
[PCL+16] ont récemment initié l'idée selon laquelle la prochaine évolution des systèmes de recommandation serait de passer d'un système de recommandation contextuel (*context-aware*) qui s'adapte au contexte à un système de recommandation piloté par le contexte (*context-driven*). L'idée est de ne plus prendre en compte le contexte comme une information complémentaire, mais de le faire passer à l'avant plan, c'est-à-dire faire de la recommandation en se basant sur ce qui se passe autour de l'utilisateur à l'instant donné (la situation), et ce qu'il est en train d'accomplir (l'intention), au lieu de se baser sur son comportement dans le passé. Cela aurait pour avantages de limiter la « sur-spécialisation », de donner aux utilisateurs plus de contrôle (il est plus facile de contrôler son contexte, sa situation ou son but du moment plutôt que des données passées accumulées), de limiter les recommandations de « genre » (basées sur l'intention et la situation et non sur des informations non choisies par l'utilisateur), de fournir de meilleures explications (plus compréhensibles pour les utilisateurs). Cet axe de recherche soulève de nombreux défis liés notamment à la modélisation et à l'intégration d'information (contextuelle) continue/temps réel sans endommager sa valeur, et à la sérialité des éléments (il existe une dépendance séquentielle entre les éléments puisque le contexte est une information continue).

Systèmes de recommandation multi-domaines
La prolifération des sites de commerce électronique, des médias sociaux, ... a permis aux utilisateurs de fournir des commentaires, d'exprimer leurs préférences/intérêts et de maintenir des profils utilisateurs dans de multiples systèmes, reflétant la variété de leurs goûts/intérêts. Tirer parti de toutes ces informations disponibles dans différents systèmes et relatives à différents champs/spécialités peut être bénéfique pour générer des profils utilisateurs plus complets et de meilleures recommandations, par exemple, en atténuant les problèmes de démarrage à froid ou de faible densité dans un champs cible ou en proposant des recommandations personnalisées « croisées » pour des éléments de champs différents. Les systèmes de recommandation multi-domaines (*cross-domain recommender systems*) visent à générer ou à améliorer des recommandations pour un champs particulier en exploitant les profils utilisateurs (ou toutes autres données/informations) issues d'autres champs [RRSK11]. Cet axe de recherche émergent [KZEK14, WZGS15] soulève de nombreux défis comme la définition de la notion de « champs », la définition de la tâche/stratégie de tels systèmes ou encore les techniques/modèles appropriés pour le transfert des données/informations/connaissances d'un champs à un autre, ... [KIG17].

Vers une plateforme générique de recommandation
Notre objectif à long terme est de fournir une plateforme générique de recommandation pouvant s'adapter aux besoins/attentes des utilisateurs et à différents domaines d'applications permettant ainsi, quel que soit l'utilisateur ou le domaine d'application, de proposer des recommandations pertinentes (à base de contenu, collaboratives, hybrides). L'hypothèse sous-jacente à une telle proposition est de réussir à identifier quelle approche particulière (quelles fonctions/méthodes pour

chacune des trois étapes de notre processus générique ainsi que pour la recommandation par défaut) est la plus appropriée pour quel cas. L'idée serait de proposer, via cette plateforme, un système de recommandation de paramètres pour le système générique de recommandation, c'est-à-dire un système de recommandation qui suggère les paramètres adaptés/pertinents en fonction du domaine d'application et des recommandations attendues. Il s'agirait d'instancier notre système générique de recommandation, présenté dans le chapitre 4, avec une matrice d'utilité Utilisateurs $\times$ Paramètres (indiquant l'utilité des paramètres pour les utilisateurs en fonction du contexte). Le système retournerait ainsi les (combinaisons de) paramètres les plus pertinent(e)s pour l'utilisateur qui souhaite instancier le système générique de recommandation le mieux possible dans un cadre précis (domaine d'application donné et recommandations attendues).

L'étape suivante d'un tel système serait de proposer un système auto-adaptatif capable de paramétrer le système de recommandation sans intervention humaine.

Bibliographie

[AAG+15] Linda Atif, Pierre-Emmanuel Arduin, Michel Grundstein, Elsa Negre, and Camille Rosenthal-Sabroux. The role of the enterprise's information and knowledge system within the digital enterprise. In *KMIKS (International conference on Knowledge Management, Information and Knowledge Systems) 2015, Hammamet - Tunisia, April 2015*, 2015.

[AC04] A. Amin and P. Cohendet. *Architectures of Knowledge : Firms, Capabilities, and Communities*. Oxford University Press, 2004.

[ACQL14] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, and José Ochoa Luna. Online courses recommendation based on LDA. In *Proceedings of the 1st Symposium on Information Management and Big Data - SIMBig 2014, Cusco, Peru, October 8-10, 2014.*, pages 42–48, 2014.

[ADE+13] Alberto Abelló, Jérôme Darmont, Lorena Etcheverry, Matteo Golfarelli, Jose-Norberto Mazón, Felix Naumann, Torben Bach Pedersen, Stefano Rizzi, Juan Trujillo, Panos Vassiliadis, and Gottfried Vossen. Fusion cubes : Towards self-service business intelligence. *IJDWM*, 9(2) :66–88, 2013.

[ADG+12] Pierre-Emmanuel Arduin, Quang Minh Doan, Daniela Grigori, Malika Grim-Yefsah, Michel Grundstein, Elsa Negre, Camille Rosenthal-Sabroux, and Virginie Thion. Evaluation d'un système d'information et de connaissance - de l'importance de la prise en compte de la connaissance (C). In *Actes du XXXème Congrès INFORSID, Montpellier, France, 29 - 31 mai 2012*, pages 371–378, 2012.

[AG16] Carole Adam and Benoit Gaudou. Modelling human behaviours in disasters from interviews : application to melbourne bushfires. In *Social Simulation Conference (SSC), Rome, Italy*, 2016.

[AGHH12] Grigoris Antoniou, Paul Groth, Frank van van Harmelen, and Rinke Hoekstra. *A Semantic Web Primer*. The MIT Press, 2012.

[AGNR13a] Pierre-Emmanuel Arduin, Michel Grundstein, Elsa Negre, and Camille Rosenthal-Sabroux. Formaliser pour améliorer la communication entre utilisateurs et concepteurs. In *Actes du XXXIème Congrès INFORSID, Paris, France, 29-31 Mai 2013.*, pages 285–300, 2013.

[AGNR13b] Pierre-Emmanuel Arduin, Michel Grundstein, Elsa Negre, and Camille Rosenthal-Sabroux. Formalizing an empirical model : A way to enhance the communication between users and designers. In *IEEE 7th International Conference on Research Challenges in Information Science, RCIS 2013, Paris, France, May 29-31, 2013*, pages 1–10, 2013.

[AGRS15] Pierre-Emmanuel Arduin, Michel Grundstein, and Camille Rosenthal-Sabroux. *Système d'information et de connaissance*. ISTE, 2015.

[AGW10] M. Angermann, M. Guenther, and K. Wendlandt. Communication architecture of an early warning system. *Natural Hazards and Earth System Science*, 10(11) :2215–2228, 2010.

[AH09] Staffan Andersson and Paul M. Heywood. The politics of perception : Use and abuse of transparency international's approach to measuring corruption. *Political Studies*, 57 :746–767, 2009.

[AHC+14] Amin Abdalla, Yingjie Hu, David Carral, Naicong Li, and Krzysztof Janowicz. An ontology design pattern for activity reasoning. In Victor de Boer, Aldo Gangemi, Krzysztof Janowicz, and Agnieszka Lawrynowicz, editors, *WOP*, volume 1302 of *CEUR Workshop Proceedings*, pages 78–81. CEUR-WS.org, 2014.

[Ama17] Amazon. www.amazon.fr, 2017.

[AMN10a] Julien Aligon, Patrick Marcel, and Elsa Negre. A framework for summarizing a log of OLAP queries. In *International Conference on Machine and Web Intelligence (ICMWI 10), Algiers, Algeria*, October 2010.

[AMN10b] Julien Aligon, Patrick Marcel, and Elsa Negre. Résumés et interrogations de logs de requêtes OLAP (compléments). Technical report, Université François Rabelais Tours - Laboratoire d'Informatique, 2010.

[AMN11a] Julien Aligon, Patrick Marcel, and Elsa Negre. Résumés et interrogations de logs de requête OLAP. In *Extraction et gestion des connaissances (EGC'2011), Actes, 25 au 29 janvier 2011, Brest, France*, pages 239–250, 2011.

[AMN11b] Julien Aligon, Patrick Marcel, and Elsa Negre. Summarizing and querying logs of OLAP queries. In *Advances in Knowledge Discovery and Management - Volume 3 [Best of EGC 2011, Brest, France].*, pages 99–124, 2011.

[AMN16] Maude Arru, Brice Mayag, and Elsa Negre. Early-warning system perception : a study on fire safety. In *13th International Conference on Information Systems for Crisis Response and Management, Rio de Janeiro, Brasil*, May 22-25 2016.

[AMNR14] Pierre-Emmanuel Arduin, Brice Mayag, Elsa Negre, and Camille Rosenthal-Sabroux. How to compromise on the best price ? A group tacit knowledge-based multicriteria approach. *Journal of Decision Systems*, 23(1) :99–112, 2014.

[AN15] Amirah AlJassir and Elsa Negre. Les hommes numériques dans la ville intelligente. In *INFORSID (Atelier), Biarritz*, 2015.

[AN17] Maude Arru and Elsa Negre. People behaviors in crisis situations : Three modeling propositions. In *Information Systems for Crisis Response and Management - ISCRAM, Albi, May 2017*, 2017.

[ANRG16] Maude Arru, Elsa Negre, Camille Rosenthal-Sabroux, and Michel Grundstein. Towards a responsible early-warning system : Knowledge implications in decision support design. In *Tenth IEEE International Conference on Research Challenges in Information Science, RCIS 2016, Grenoble, France, June 1-3, 2016*, pages 1–12, 2016.

[ANRS13] Pierre-Emmanuel Arduin, Elsa Negre, and Camille Rosenthal-Sabroux. Towards a new tacit knowledge-based approach for decision process. In *KMIKS (International conference on Knowledge Management, Information and Knowledge Systems) 2015, Hammamet - Tunisia, April 2013*, 2013.

[ANRS16] Pierre-Emmanuel Arduin, Elsa Negre, and Camille Rosenthal-Sabroux. Knowledge and decision for smart cities initiatives - cases of paris and nice. In *IEEE International Conference on Industrial Informatics and Computer Systems (CIICS) - Dubaï, March 2016*, 2016.

[ANRS17a] Maude Arru, Elsa Negre, and Camille Rosenthal-Sabroux. A representation of people behaviors in crisis situations. In *Proceedings of International Conf. on Knowledge Management, Information and Knowledge Systems (KMIKS), Tunisia, April 2017*, 2017.

[ANRS17b] Maude Arru, Elsa Negre, and Camille Rosenthal-Sabroux. Vers une modélisation des comportements en situation de crise. In *INFORSID (Atelier), Toulouse*, 2017.

[ANRSG16] Maude Arru, Elsa Negre, Camille Rosenthal-Sabroux, and Michel Grundstein. Towards a responsible early-warning system : Knowledge implications in decision support design. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference*, pages 1–12. IEEE, 2016.

[AT05] G. Adomavicius and A. Tuzhilin. Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6) :734–749, 2005.

[AT08] Gediminas Adomavicius and Alexander Tuzhilin. Context-aware recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 335–336, New York, NY, USA, 2008. ACM.

[AT11] Gediminas Adomavicius and Alexander Tuzhilin. *Context-Aware Recommender Systems*, pages 217–253. Springer US, Boston, MA, 2011.

[AWY14] D.A. Adeniyi, Z. Wai, and Y. Yongquan. Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method. *Applied Computing and Informatics*, (0) :–, 2014.

[AZM14] Rawad Abou Assi, Fadi A. Zaraket, and Wes Masri. Ucov : a user-defined coverage criterion for test case intent verification. *CoRR*, abs/1407.3091, 2014.

[BAG+12] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali. Smart cities of the future. *The European Physical Journal Special Topics*, 214(1) :481–518, 2012.

[Bak01] Edwin Bakker. Early warning by ngos in conflict areas. In Math Noortmann and Bob Reinalda, editors, *Bas Arts*, pages 263–277. Non-State Actors in International Relations, 2001.

[Bak05] Kirk Baker. Singular value decomposition tutorial. 2005.

[Bal08] Linas Baltrunas. Exploiting contextual information in recommender systems. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pages 295–298, New York, NY, USA, 2008. ACM.

[Bat13] Michael Batty. Big data, smart cities and city planning. *Dialogues in Human Geography*, 3(3) :274–279, 2013.

[BB05] Mary Bazire and Patrick Brézillon. Understanding context before using it. In *Proceedings of the 5th International Conference on Modeling and Using Context*, CONTEXT'05, pages 29–40, Berlin, Heidelberg, 2005. Springer-Verlag.

[BBH+10] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2) :161 – 180, 2010. Context Modelling, Reasoning and Management.

[BC15] Fatiha Bousbahi and Henda Chorfi. Mooc-rec : A case based recommender system for moocs. *Procedia - Social and Behavioral Sciences*, 195 :1813 – 1822, 2015. World Conference on Technology, Innovation and Entrepreneurship.

[BC17] Imène Brigui-Chtioui and Philippe Caillou. Multidimensional decision model for classifying learners : the case of massive online open courses (moocs). *Journal of Decision Systems*, 26(2) :170–188, 2017.

[BCCN17] Imene Brigui-Chtioui, Philippe Caillou, and Elsa Negre. Intelligent digital learning : Agent-based recommender system. In *International Conference on Machine Learning and Cybernetics, ICMLC 2017, Singapore*, 2017.

[BdN14] Sandro Bimonte, Laurent d'Orazio, and Elsa Negre, editors. *Actes des 10e journées francophones sur les Entrepôts de Données et l'Analyse en Ligne, EDA 2014, Vichy, France, 5-6 Juin, 2014*, volume B-10 of *RNTI*. Hermann-Éditions, 2014.

[BDR91] Sanjiv K. Bhatia, Jitender S. Deogun, and Vijay V. Raghavan. User profiles for information retrieval. In Zbigniew W. Ras and Maria Zemankova, editors, *ISMIS*, volume 542 of *Lecture Notes in Computer Science*, pages 102–111. Springer, 1991.

[BG09] Charles Bouveyron and Stephane Girard. Classification supervisée et non supervisée des données de grande dimension. La revue de Modulad, Modulad, 40, pp.81-102, 2009.

[BGM+05] L. Bellatreche, A. Giacometti, P. Marcel, H. Mouloudi, and D. Laurent. A personalization framework for olap queries. In *ACM International Workshop on Datawarehousing and OLAP, DOLAP*, pages 9–18, 2005.

[BHK98] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[BLPR12] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Comput.*, 16(5) :507–526, June 2012.

[BLSW10] Luís M. A. Bettencourt, José Lobo, Deborah Strumsky, and Geoffrey B. West. Urban Scaling and Its Deviations : Revealing the Structure of Wealth, Innovation and Crime across Cities. *PLoS ONE*, 5(11) :e13541+, November 2010.

[BM05] Mustafa Bilgic and Raymond Mooney. Explaining recommendations : Satisfaction vs. promotion. In *In Proceedings of Beyond Personalization : A Workshop on the Next Stage of Recommender Systems Research at the International Conference on Intelligent User Interfaces*, 2005.

[BMFT16] Frederick Benaben, Aurelie Montarnal, Audrey Fertier, and Sebastien Truptil. *Big-Data and the Question of Horizontal and Vertical Intelligence : A Discussion on Disaster Management*, pages 156–162. Springer International Publishing, Cham, 2016.

[BN16] Sandro Bimonte and Elsa Negre. Evaluation of user satisfaction with OLAP recommender systems : an application to RecoOLAP on a agricultural energetic consumption datawarehouse. *IJBIS*, 21(1) :117–136, 2016.

[BP00] Daniel Billsus and Michael J. Pazzani. User modeling for adaptive news access. *User Model. User-Adapt. Interact.*, 10(2-3) :147–180, 2000.

[BPB+13] Sandro Bimonte, Marilys Pradel, Daniel Boffety, Aurelie Tailleur, Géraldine André, Rabi Bzikha, and Jean-Pierre Chanet. A new sensor-based spatial olap architecture centered on an agricultural farm energy-use diagnosis tool. *Int. J. Decis Support Syst. Technol.*, 5(4) :1–20, October 2013.

[BR07] Suela Bohe and Béatrice Rumpler. Modèle évolutif d un profil utilisateur Application à la Recherche d Information dans une bibliothèque numérique de thèses. In *CORIA*, pages 197–209, March 2007.

[BR16] Justin Basilico and Yves Raimond. Recommending for the world. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 375–375, New York, NY, USA, 2016. ACM.

[BS97] Marko Balabanovic and Yoav Shoham. Fab : content-based, collaborative recommendation. *Communications of the ACM*, 40(3) :66–72, 1997.

[BS00] Margaret Buchanan-Smith. Role of early warning systems in decision making processes. In *an Expert Group Meeting held September 5-7, in Lisbon, Portugal.*, 2000.

[Bur02] Robin Burke. Hybrid recommender systems : Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4) :331–370, November 2002.

[BW10] Luis Bettencourt and Geoffrey West. A unified theory of urban living. *Nature*, 467(7318) :912–913, October 2010.

[BYHM04] Ricardo A. Baeza-Yates, Carlos A. Hurtado, and Marcelo Mendoza. Query Recommendation Using Query Logs in Search Engines. In Wolfgang Lindner, Marco Mesiti, Can Türker, Yannis Tzitzikas, and Athena Vakali, editors, *EDBT Workshops*, volume 3268 of *Lecture Notes in Computer Science*, pages 588–596. Springer, 2004.

[BYHMD05] Ricardo Baeza-Yates, Carlos Hurtado, Marcelo Mendoza, and Georges Dupret. Modeling User Search Behavior. In *LA-WEB '05 : Proceedings of the Third Latin American Web Congress*, page 242, Washington, DC, USA, 2005. IEEE Computer Society.

[BYRN99] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[CCFG12] Ansem Ben Cheikh, Stéphane Coulondre, Agnès Front, and Jean-Pierre Giraudin. An engineering method for context-aware and reactive systems. In *Sixth Int. Conf. on Research Challenges in Information Science, RCIS 2012, Valencia, Spain, May 16-18 2012*, pages 1–12, 2012.

[CCS93] E. F. Codd, S. B. Codd, and C. T. Salley. Providing OLAP (On-Line Analytical Processing) to User-Analysis : An IT Mandate, 1993.

[CCST11] Hamdi Chaker, Max Chevalier, Chantal Soulé-Dupuy, and André Tricot. Business context information manager : An approach to improve information systems. In *Modeling and Using Context - 7th Int. and Interdisciplinary Conf., CONTEXT 2011, Karlsruhe, Germany, September 26-30, 2011.*, pages 67–70, 2011.

[CD97] Surajit Chaudhuri and Umeshwar Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record*, 26(1) :65–74, 1997.

[CEG12] Paolo Cremonesi, Francesco Epifania, and Franca Garzotto. User profiling vs. accuracy in recommender system user experience. In *Proceedings of the International Working*

Conference on Advanced Visual Interfaces, AVI '12, pages 717–720, New York, NY, USA, 2012. ACM.

[CEP09] Gloria Chatzopoulou, Magdalini Eirinaki, and Neoklis Polyzotis. Query Recommendations for Interactive Database Exploration. In *SSDBM*, pages 3–18, 2009.

[CFCG10] Ansem Ben Cheikh, Agnès Front, Stéphane Coulondre, and Jean-Pierre Giraudin. Event based modeling for context-reactive information systems. In *Sixth Int. Conf. on Signal-Image Technology and Internet-Based Systems, SITIS 2010, Kuala Lumpur, Malaysia, December 15-18, 2010*, pages 261–268, 2010.

[CFT07] G. Castellano, A. M. Fanelli, and M. A. Torsello. Lodap : A log data preprocessor for mining web browsing patterns. In *Proceedings of the 6th Conference on 6th WSEAS Int. Conf. On Artificial Intelligence, Knowledge Engineering and Data Bases - Volume 6*, AIKED'07, pages 12–17, Stevens Point, Wisconsin, USA, 2007. World Scientific and Engineering Academy and Society (WSEAS).

[CGT13] Paolo Cremonesi, Franca Garzotto, and Roberto Turrin. *User-Centric vs. System-Centric Evaluation of Recommender Systems*, pages 334–351. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[CH06] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theor.*, 13(1) :21–27, September 2006.

[Che09] Nicolas Cheang. Early warning system for financial crises. *Macao Monetary Research Bulletin*, (11) :61–77, 2009.

[CJSV09] Max Chevalier, Christine Julien, Chantal Soulé-Dupuy, and Nathalie Vallès-Parlangeau. Usagers et recherche d'information. modélisation des usagers pour la recherche d'information adaptative. *Ingénierie des Systèmes d'Information*, 14(3) :75–95, 2009.

[CKS03] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum. *CMMI Guidlines for Process Integration and Product Improvement*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[CMN10] Sonia Colas, Patrick Marcel, and Elsa Negre. Organisation de log de requêtes OLAP sous forme de site web. In *Actes des 6èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne, EDA 2010, Djerba, Tunisie, Juin 2010*, pages 81–95, 2010.

[CMN14] Tina Comes, Brice Mayag, and Elsa Negre. Decision support for disaster risk management : Integrating vulnerabilities into early-warning systems. In *Information Systems for Crisis Response and Management in Mediterranean Countries - First International Conference, ISCRAM-med 2014, Toulouse, France, October 15-17, 2014. Proceedings*, pages 178–191, 2014.

[CMN15] Tina Comes, Brice Mayag, and Elsa Negre. Beyond early : Decision support for improved typhoon warning systems. In *12th Proceedings of the International Conference on Information Systems for Crisis Response and Management, Krystiansand, Norway, May 24-27, 2015.*, 2015.

[CMS08] Sonia Colas, Nicolas Monmarché, and Mohamed Slimane. Génération de plan de site web pour les non-voyants par des fourmis artificielles. *Revue d'Intelligence Artificielle*, 22(2) :137–159, 2008.

[CNW+12] Hafedh Chourabi, Taewoo Nam, Shawn Walker, J. Ramon Gil-Garcia, Sehl Mellouli, Karine Nahon, Theresa A. Pardo, and Hans Jochen Scholl. Understanding smart cities : An integrative framework. In *Proceedings of the 2012 45th Hawaii Int. Conf. on System Sciences*, HICSS '12, pages 2289–2297, Washington, DC, USA, 2012. IEEE Computer Society.

[Col96] George Colliat. Olap, relational, and multidimensional database systems. *SIGMOD Record*, 25(3) :64–69, 1996.

[Coo01] Colleen Cool. The concept of situation in information science. *Annual Review of Information Science and Technology*, (35) :5–42, 2001.

[CP06] Li Chen and Pearl Pu. Evaluating critiquing-based recommender agents. In *AAAI*, pages 157–162. AAAI Press, 2006.

[Cro05] Bill Crowley. Rediscovering the history of readers advisory service. *Public Libraries*, 44(1) :38, 2005.

[CS93] R.E Chuang and David Sher. chi2 test for feature detection. *Pattern Recognition*, 26(11) :1671–1681, 1993.

[CSMF00] Lei-da Chen, Khalid S. Soliman, En Mao, and Mark N. Frolick. Measuring user satisfaction with data warehouses : An exploratory study. *Inf. Manage.*, 37(3) :103–110, April 2000.

[Dav89] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.*, 13(3) :319–340, September 1989.

[DB09] Leo Dobes and Jeff Bennett. Multi-criteria analysis : "good enough" for government work ? *Agenda : A Journal of Policy Analysis and Reform*, 16 :7–29, 2009.

[DBB07] Jérôme Darmont, Fadila Bentayeb, and Omar Boussaid. Benchmarking data warehouses. *IJBIDM*, 2(1) :79–104, 2007.

[DDF+90] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6) :391–407, 1990.

[DGS13] DGSCGC. *Guide ORSEC - Alerte et information des populations*. Direction générale de la sécurité civile et de la gestion des crises, 2013.

[Dic45] L. R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3) :297–302, July 1945.

[dN15] Laurent d'Orazio and Elsa Negre, editors. *International Journal of Decision Support System Technology (IJDSST)*, volume 7. IGI Global, 2015.

[DNR16] Renata Paola Dameri, Elsa Negre, and Camille Rosenthal-Sabroux. Triple helix in smart cities : A literature review about the vision of public bodies, universities, and private companies. In *49th Hawaii Int. Conf. on System Sciences, HICSS, Koloa, HI, USA, January 5-8, 2016*, pages 2974–2982, 2016.

[DP97] Thomas H. Davenport and Laurence Prusak. *Working Knowledge : How Organizations Manage What They Know*. Harvard Business School Press, Boston, MA, USA, 1997.

[DPP06] Zhongli Ding, Yun Peng, and Rong Pan. A bayesian approach to uncertainty modelling in owl ontology. Technical report, DTIC Document, 2006.

[DSA01] A. Dey, D. Salber, and G. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, 2001.

[DSiS91] Britt-Marie. Drottz-Sjoberg and Handelshögskolan i Stockholm. *Perception of risk : studies of risk attitudes, perceptions and definitions / Britt-Marie Drottz-Sjoberg.* Center for Risk Research, Stockholm School of Economics Stockholm, 1991.

[DTBC08] Mariam Daoud, Lynda Tamine, Mohand Boughanem, and Bilal Chebaro. Construction des profils utilisateurs à base d'une ontologie pour une recherche d'information personnalisée. In *Conf. francophone en Recherche d'Information et Applications (CORIA), Trégastel, Mars 2008*, pages 225–240, http ://www.univ-rennes1.fr, mars 2008. Université de Rennes 1.

[Dup03] Jean-Pierre Dupuy. *La panique.* Empêcheurs de penser en rond, 2003.

[dW08] Joost de Wit. Evaluating recommender systems. Master's thesis, University of Twente, May 2008.

[EAJ+10] Fathi Essalmi, Leila Jemni Ben Ayed, Mohamed Jemni, Kinshuk, and Sabine Graf. A fully personalization strategy of e-learning scenarios. *Computers in Human Behavior*, 26(4) :581 – 591, 2010. Emerging and Scripted Roles in Computer-supported Collaborative Learning.

[edX15] edX. Research guide. http ://edx.readthedocs.org/en/latest/, 2015.

[EF71] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2) :124, 1971.

[Ess10] Ilham Esslimani. *Vers une approche comportementale de recommandation : apport de l'analyse des usages dans un processus de personnalisation.* PhD thesis, Université Nancy II, 2010.

[FA14] Sallam Osman Fageeri and Rohiza Ahmad. An efficient log file analysis algorithm using binary-based data structure. *Procedia - Social and Behavioral Sciences*, 129 :518 – 526, 2014. 2nd Int. Conf. on Innovation, Management and Technology Research.

[Fac17] Facebook. https ://fr-fr.facebook.com, 2017.

[FB06] Rosta Farzan and Peter Brusilovsky. Social navigation support in a course recommendation system. In *4th Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*, AH'06, pages 91–100, Berlin, Heidelberg, 2006. Springer-Verlag.

[FGG97] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29(2-3) :131–163, November 1997.

[FMG15] Mohamed Frikha, Mohamed Mhiri, and Faiez Gargouri. A semantic social recommender system using ontologies based approach for tunisian tourism. *ADCAIJ*, 4(1), 2015.

[FMR05] J. Figueira, V. Mousseau, and B. Roy. ELECTRE methods. In J. Figueira, S. Greco, and M. Ehrgott, editors, *Multiple Criteria Decision Analysis : State of the Art Surveys*, pages 133–162. Springer Verlag, Boston, Dordrecht, London, 2005.

[FNC17] Zahra Vahidi Ferdousi, Elsa Negre, and Dario Colazzo. Context factors in context-aware recommender systems. In *Atelier interdisciplinaire sur les systèmes de recommandation*, 2017.

[Fod02] Imola Fodor. A survey of dimension reduction techniques. Technical report, 2002.

[For17] Forbes.com. http ://www.forbes.com, 2017.

[Fra32] Académie Française. Dictionnaire. atilf.atilf.fr/academie.htm ; www.academie-francaise.fr, 1932.

[FS02] Yongjian Fu and Ming-Yi Shih. A Framework for Personal Web Usage Mining. In *International Conference on Internet Computing*, pages 595–600, 2002.

[Gai78] Brian R Gaines. Fuzzy and probability uncertainty logics. *Information and Control*, 38(2) :154–169, 1978.

[Ger10] German Committee for Disaster Reduction (DKKV) ; Humanitarian & Development Network (HDN) ; Platform for the Promotion of Early Warning (UNISDR PPEW). *Emerging challenges for early warning systems in context of climate change and urbanization*, 2010.

[GFK+07] Rudolf Giffinger, Christian Fertner, Hans Kramar, Robert Kalasek, Nataï£¡a Pichler-Milanovic, and Evert Meijers. *Smart Cities - Ranking of European medium-sized cities*. Vienna University of Technology, 2007.

[GJ02] Alejandro Gaytàn and Christian A. Johnson. A review of the literature on early warning systems for banking crises. Working Papers Central Bank of Chile 183, Central Bank of Chile, October 2002.

[GJ05] Robert L Goldstone and Marco A Janssen. Computational models of collective behavior. *Trends in cognitive sciences*, 9(9) :424–430, 2005.

[GJM02] Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2002.

[GK65] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. 1965.

[GKL11] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *WSDM*, pages 595–604, 2011.

[GMN08] Arnaud Giacometti, Patrick Marcel, and Elsa Negre. A framework for recommending olap queries. In *DOLAP*, pages 73–80, 2008.

[GMN09] Arnaud Giacometti, Patrick Marcel, and Elsa Negre. Recommending Multidimensional Queries. In *DaWaK*, pages 453–466, 2009.

[GMNS09] Arnaud Giacometti, Patrick Marcel, Elsa Negre, and Arnaud Soulet. Query recommendations for OLAP discovery driven analysis. In *DOLAP 2009, ACM 12th International Workshop on Data Warehousing and OLAP, Hong Kong, China, November 6, 2009, Proceedings*, pages 81–88, 2009.

[GMNS11] Arnaud Giacometti, Patrick Marcel, Elsa Negre, and Arnaud Soulet. Query recommendations for OLAP discovery-driven analysis. *IJDWM*, 7(2) :1–25, 2011.

[GPB10] Mustansar Ghazanfar and Adam Prugel-Bennett. An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering. Event Dates : 17-19 March, 2010, April 2010.

[GR11] Matteo Golfarelli and Stefano Rizzi. Data warehouse testing : A prototype-based methodology. *Information & Software Technology*, 53(11) :1183–1198, 2011.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2) :199–220, June 1993.

[Gru12] M. Grundstein. *Three Postulates That Change Knowledge Management Paradigm*, pages 1–26. Huei-Tse Hou (Ed.), In Tech, 2012.

[GS97] David Gefen and Detmar W. Straub. Gender differences in the perception and use of e-mail : An extension to the technology acceptance model. *MIS Q.*, 21(4) :389–400, December 1997.

[GSB+10] P Genovesi, R Scalera, S Brunel, D Roy, and W Solarz. Towards an early warning and information system for invasive alien species (ias) threatening biodiversity in europe. Technical report, EEA (European Environment Agency), 2010.

[Gué06] Sébastien Guérif. *Réduction de dimension en Apprentissage Numérique Non Supervisé*. PhD thesis, Université Paris 13, 2006.

[Hau14] Felix Hausdorff. *Grundzüge der Mengenlehre*. von Veit, 1914.

[HBP08] Katharine Haynes, Jenni Barclay, and Nick Pidgeon. The issue of trust and its influence on risk communication during a volcanic crisis. *Bulletin of Volcanology*, 70(5) :605–621, Mar 2008.

[HI06] Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications : Models and approach. *Pervasive Mob. Comput.*, 2(1) :37–64, February 2006.

[HKR00] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, pages 241–250, New York, NY, USA, 2000. ACM.

[HKTR04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1) :5–53, January 2004.

[Hoc56] J Hochberg. Perception : toward the recovery of a definition. *Psychological review*, 63 6 :400–5, 1956.

[HP00] Jiawei Han and Jian Pei. Mining frequent patterns by pattern-growth : Methodology and implications. *SIGKDD Explor. Newsl.*, 2(2) :14–20, December 2000.

[HSM+07] Dominik Heckmann, Eric Schwarzkopf, Junichiro Mori, Dietmar Dengler, and Alexander Krï£¡ner. The user model and context ontology gumo revisited for future web 2.0 extensions. In *C&O :RR*, volume 298 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[HSRF95] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *CHI '95 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[Hsu08] Mei-Hua Hsu. A personalized english learning recommender system for ESL students. *Expert Systems with Applications*, 34(1) :683 – 688, 2008.

[HT09] J.B. Heppen and S.B. Therriault. Developing early warning systems to identify potential high school dropouts. Technical report, National High School Center, 2009.

[Hua08] Anna Huang. Similarity Measures for Text Document Clustering. In Jay Holland, Amanda Nicholas, and Delio Brignoli, editors, *New Zealand Computer Science Research Student Conference*, pages 49–56, April 2008.

[HW79] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR : Applied Statistics*, 28(1) :100–108, 1979.

[HWN+08] Mohd Helmy, Abd Wahab, Mohd Norzali, Haji Mohd, Hafizul Fahri Hanafi, Mohamad Farhan, and Mohamad Mohsin. Data pre-processing on web server logs for generalized association rules mining algorithm, 2008.

[HZC04] Zan Huang, Daniel Zeng, and Hsinchun Chen. A link analysis approach to recommendation under sparse data. In *AMCIS*, page 239. Association for Information Systems, 2004.

[Inm92] W. H. Inmon. *Building the data warehouse.* QED Information Sciences, Inc., Wellesley, MA, USA, 1992.

[Int13] International Federation of Red Cross and Red Crescent Societies (IFRC). Community early warning systems : guiding principles, 2013.

[JA16] Dietmar Jannach and Gediminas Adomavicius. Recommendations with a purpose. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 7–10, New York, NY, USA, 2016. ACM.

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[JEQR10] Bernard Engel Joseph E. Quansah and Gilbert L. Rochon. Early warning systems : A review. *Journal of Terrestrial Observation*, 2(5), 2010.

[JLVV01] Matthias Jarke, Maurizio Lenzerini, Yannis Vassiliou, and Panos Vassiliadis. *Fundamentals of Data Warehouses.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[JRTZ09] Houssem Jerbi, Franck Ravat, Olivier Teste, and Gilles Zurfluh. Applying Recommendation Technology in OLAP Systems. In *International Conference on Enterprise Information Systems (ICEIS), Milan (Italie), 06/05/2009-10/05/2009*, number 24 in LNBIP, pages 220–233, http ://www.springerlink.com, 2009. Springer.

[JS99] A. Shani J. Sena. Intellectual capital and knowledge creation : Towards an alternative framework. *Knowledge Management Handbook*, pages 1–16, 1999.

[JZFF10] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems : An Introduction.* Cambridge University Press, New York, NY, USA, 1st edition, 2010.

[KBG+09] Nodira Khoussainova, Magdalena Balazinska, Wolfgang Gatterbauer, YongChul Kwon, and Dan Suciu. A case for a collaborative query management system. In *CIDR*, 2009.

[kBGM15] Dheeraj kumar Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. An item-based collaborative filtering using dimensionality reduction techniques on mahout framework. *CoRR*, abs/1503.06562, 2015.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37, August 2009.

[Ken71] Allen Kent. *Information Analysis and Retrieval.* John Wiley and sons, Inc., 1971.

[KI04] Georgia Koutrika and Yannis Ioannidis. Personalization of queries in database systems. In *Proceedings of the 20th International Conference on Data Engineering*, ICDE '04, pages 597–, Washington, DC, USA, 2004. IEEE Computer Society.

[KIG17] Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. Cross domain recommender systems : A systematic literature review. *ACM Comput. Surv.*, 50(3) :36 :1–36 :34, June 2017.

[Kim96] Ralph Kimball. *The Data Warehouse Toolkit : Practical Techniques for Building Dimensional Data Warehouses.* John Wiley, 1996.

[KK06] Przemyslaw Kazienko and Pawel Kolodziejski. Personalized Integration of Recommendation Methods for e-commerce. *IJCSA*, 3(3) :12–26, 2006.

[Kło08] Mieczysław A. Kłopotek. Approaches to cold start in recommender systems. In Mieczysław A. Kłopotek and Jerzy Tchórzewski, editors, *Proceedings of 10th International Conference on Artificial Intelligence*, volume 5 of *Proceedings of Artificial Intelligence Studies*, pages 29–33, Siedlce, Poland, September 18-19, 2008.

[Klo09] M. A. Klopotek. Approaches to cold-start in recommender systems. *Studia Informatica : systems and information technology*, Vol. 1(12) :47–54, 2009.

[KPC16] Zhao Kang, Chong Peng, and Qiang Cheng. Top-n recommender system via matrix completion. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 179–184. AAAI Press, 2016.

[KSU09] Jérôme Kunegis, Alan Said, and Winfried Umbrath. The universal recommender. *CoRR*, abs/0909.3472, 2009.

[KWFH04] Abhijit Kadlag, Amol V. Wanjari, Juliana Freire, and Jayant R. Haritsa. *Supporting Exploratory Queries in Databases*, pages 594–605. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[KWG+12] Bart P. Knijnenburg, Martijn C. Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5) :441–504, October 2012.

[KZEK14] Alfred Kobsa, Michelle X. Zhou, Martin Ester, and Yehuda Koren, editors. *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*. ACM, 2014.

[las17] last.fm. http ://www.lastfm.fr, 2017.

[LBP08] Cyprien Lomas, Michael Burke, and Carie L Page. Collaboration tools. 2008.

[LIC03] Paul Legris, John Ingham, and Pierre Collerette. Why do people use information technology ? : A critical review of the technology acceptance model. *Inf. Manage.*, 40(3) :191–204, January 2003.

[Lik32] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 22(140) :1–55, 1932.

[Lin17] LinkedIn. https ://fr.linkedin.com, 2017.

[LL91] C-T Lin and C. S. George Lee. Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on computers*, 40(12) :1320–1336, 1991.

[LL01] Kenneth C. Laudon and Jane Price Laudon. *Management Information Systems : Managing the Digital Firm*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 7th edition, 2001.

[LLL98] Charles Ling, , Charles X. Ling, and Chenghui Li. Data mining for direct marketing : Problems and solutions. In *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 73–79. AAAI Press, 1998.

[Lou12] Ahmed Said Loubiri. Utilisation d'une ontologie et du rï£¡seau social facebook pour la modï£¡lisation du contexte pour les applications mobiles dï£¡pendantes du contexte. Master's thesis, Montrï£¡al (Quï£¡bec, Canada), 2012.

[LU03] Abdelkrim Lahlou and Pascal Urien. Sim-filter : User profile based smart information filtering and personalization in smartcard. In *The 15th Conference on Advanced Information Systems Engineering (CAiSE '03), Klagenfurt/Velden, Austria, 16-20 June, 2003, Workshops Proceedings, Information Systems for a Connected Society*, 2003.

[Lud04] Sydney C. Ludvigson. Consumer confidence and consumer spending. *Journal of Economic Perspectives*, 18(2) :29–50, June 2004.

[LVC+03] P. Lyman, H. R. Varian, P. Charles, N. Good, L. L. Jordan, and J. Pal. How much information ? http ://www2.sims.berkeley.edu/research/projects/how-much-info-2003, 2003.

[LZ13] Anisio Lacerda and Nivio Ziviani. Building user profiles to improve user experience in recommender systems. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 759–764, New York, NY, USA, 2013. ACM.

[Mar12] Patrick Marcel. *Leveraging query logs for user-centric OLAP.* PhD thesis, Habilitation à diriger des recherches - Université de Tours, France, 2012.

[MG11] M. McLuhan and W.T. Gordon. *Counterblast 1954.* Gingko PressInc, 2011.

[MHT+17] Aurélie Montarnal, Shane E. Halse, Andrea H. Tapia, Sébastien Truptil, and Frédérick Bénaben. Automated emergence of a crisis situation model in crisis response based on tweets. In *Collaboration in a Data-Rich World - 18th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2017, Vicenza, Italy, September 18-20, 2017*, pages 658–665, 2017.

[MK09] Chaitanya Mishra and Nick Koudas. Interactive query refinement. In *Proceedings of the 12th International Conference on Extending Database Technology : Advances in Database Technology*, EDBT '09, pages 862–873, New York, NY, USA, 2009. ACM.

[MN11] Patrick Marcel and Elsa Negre. A survey of query recommendation techniques for data warehouse exploration. In *Actes des 7èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne, Clermont-Ferrand, France, EDA 2011, Juin 2011*, pages 119–134, 2011.

[MNAV13] Rokia Missaoui, Elsa Negre, Dyah Anggraini, and Jean Vaillancourt. Social network restructuring after a node removal. *Int. J. Web Eng. Technol.*, 8(1) :4–26, 2013.

[MNRS17] Brice Mayag, Elsa Negre, and Camille Rosenthal-Sabroux. How to characterize your information system ? In *KMIKS (International conference on Knowledge Management, Information and Knowledge Systems)*, 2017.

[MPRB04] Ghita Kouadri Mostefaoui, Jacques Pasquier-Rocha, and Patrick Brézillon. Context-aware computing : A guide for the pervasive computing community. In *ICPS*, pages 39–48. IEEE Computer Society, 2004.

[MSS04] Fiona Murphy, Larry Stapleton, and David Smith. Tacit knowledge and human centred systems : The key to managing the social impact of technology. In *International Multitrack Conference of Advances in Control Systems*, 2004.

[MZLK11] Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Trans. Inf. Syst.*, 29(2) :9 :1–9 :23, April 2011.

[Nad16] Wanvimol Nadee. *Modelling User Profiles for Recommender systems.* PhD thesis, Science and Engineering Faculty, Queensland University of Technology, 2016.

[Nas96] T.F Nas. *Cost-benefit analysis : Theory and application.* Sage Publications, Thousand Oaks, 1996.

[Nav01] Gonzalo Navarro. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1) :31–88, March 2001.

[NB14] Mohammad-Hossein Nadimi-Shahraki and Mozhde Bahadorpour. Cold-start problem in collaborative recommender systems : Efficient methods based on ask-to-rate technique. *CIT*, 22(2) :105–113, 2014.

[Neg09] Elsa Negre. *Exploration collaborative de cubes de données.* PhD thesis, Université François-Rabelais de Tours, 2009.

[Neg11] Elsa Negre. Quand la recommandation rencontre la personnalisation. ou comment générer des recommandations (requêtes MDX) en adéquation avec les préférences de l'utilisateur. *Technique et Science Informatiques*, 30(8) :933–952, 2011.

[Neg13a] Elsa Negre. Comparaison de textes - quelques approches... Technical report, Cahiers du LAMSADE, 2013.

[Neg13b] Elsa Negre. Towards a knowledge (experience)-based recommender system for crisis management. In *Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2013, Compiegne, France, October 28-30, 2013*, pages 713–718, 2013.

[Neg14] Elsa Negre. RecoOLAP : Un système de recommandation de requêtes OLAP. In *Actes des 10e journées francophones sur les Entrepôts de Données et l'Analyse en Ligne, EDA 2014, Vichy, France, 5-6 Juin, 2014*, pages 171–174, 2014.

[Neg15a] Elsa Negre. *Recommender and Information Systems.* Wiley, 2015.

[Neg15b] Elsa Negre. *Systèmes de recommandation - Introduction.* ISTE, 2015.

[Neg17a] Elsa Negre. Pertinent context information for olap applications. In *KMIKS (International conference on Knowledge Management, Information and Knowledge Systems)*, 2017.

[Neg17b] Elsa Negre. Prise en compte du contexte dans les systèmes de recommandation de requêtes olap. In *Actes des journées francophones sur les Entrepôts de Données et l'Analyse en Ligne, EDA 2017, Lyon, France*, 2017.

[Net17] Netflix. https ://www.netflix.com/fr/, 2017.

[Ngu10] C.P. Nguyen. *Conception d'un système d'apprentissage et de travail pervasif et adaptatif fondé sur un modèle de scénario.* PhD thesis, Ecole Nationale Supérieure des Télécommunications de Bretagne, 2010.

[NMV11a] Elsa Negre, Rokia Missaoui, and Jean Vaillancourt. Predicting a social network structure once a node is deleted. In *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2011, Kaohsiung, Taiwan, 25-27 July 2011*, pages 297–304, 2011.

[NMV11b] Elsa Negre, Rokia Missaoui, and Jean Vaillancourt. Social network structure prediction. Technical report, Université du Québec en Outaouais - LAboratoire de Recherche sur l'Information Multimédia, 2011.

[NR17] Rachana Naik and Abhishek Raghuvanshi. Hybrid news recommendation system using tf-idf and associative calculus. *IJSDR, Volume 2, Issue 3 : 48-52*, 2017.

[NRG15] Elsa Negre, Camille Rosenthal-Sabroux, and Mila Gascó. A knowledge-based conceptual vision of the smart city. In *48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, USA, January 5-8, 2015*, pages 2317–2325, 2015.

[NRS14] Elsa Negre and Camille Rosenthal-Sabroux. Recommendations to improve the smartness of a city. In Renata Paola Dameri and Camille Rosenthal-Sabroux, editors, *Smart City*, Progress in IS, pages 101–115. Springer International Publishing, 2014.

[NRS15a] Elsa Negre and Camille Rosenthal-Sabroux. Recommendations to improve the smartness of a city. In *Smart city - How to create public and economic value with high technology in urban space*. Springer Verlag, 2015.

[NRS15b] Elsa Negre and Camille Rosenthal-Sabroux. Smart cities : A salad bowl of citizens, ict and environment. In *Social, Economic, and Environmental Sustainability. In the Development of Smart Cities*. 2015.

[NRTT13a] Elsa Negre, Franck Ravat, Olivier Teste, and Ronan Tournier. Cold-start recommender system problem within a multidimensional data warehouse. In *IEEE 7th International Conference on Research Challenges in Information Science, RCIS 2013, Paris, France, May 29-31, 2013*, pages 1–8, 2013.

[NRTT13b] Elsa Negre, Franck Ravat, Olivier Teste, and Ronan Tournier. Problème du démarrage à froid pour les systèmes de recommandation dans les entrepôts de données. In *Actes du XXXIème Congrès INFORSID, Paris, France, 29-31 Mai 2013.*, pages 491–506, 2013.

[NTdSX13] José A. Rodrigues Nt., Luiz Fernando Cardoso Tomaz, Jano Moreira de Souza, and Geraldo Xexéo. Bringing knowledge into recommender systems. *Journal of Systems and Software*, 86(7) :1751–1758, 2013.

[NWFF08] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3) :299–318, 2008.

[OS15] D.F.O. Onah and J.E. Sinclair. Collaborative filtering recommendation system : A framework in massive open online courses. In *INTED2015 Proceedings*, 9th International Technology, Education and Development Conference, pages 1249–1257. IATED, 2-4 March, 2015 2015.

[OW04] Peter. O'Neill and New South Wales. *Developing a risk communication model to encourage community safety from natural hazards [electronic resource] / Peter O'Neill.* State Emergency Services] [Sydney, N.S.W, 2004.

[Pal07] Joel Palmius. Criteria for measuring and comparing information systems. In *30th Information Systems Research Seminar in Scandinavia (IRIS-30)*, 2007.

[Pan07] Feng Pan. *Representing Complex Temporal Phenomena for the Semantic Web and Natural Language*. PhD thesis, Los Angeles, CA, USA, 2007. AAI3291802.

[PB07] Michael J. Pazzani and Daniel Billsus. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg, 2007.

[PCL+16] Roberto Pagano, Paolo Cremonesi, Martha Larson, Balázs Hidasi, Domonkos Tikk, Alexandros Karatzoglou, and Massimo Quadrana. The contextual turn : From context-aware to context-driven recommender systems. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 249–252, New York, NY, USA, 2016. ACM.

[PDPV+15] Damienne Provitolo, Edwige Dubos-Paillard, Nathalie Verdière, Valentina Lanza, Rodolphe Charrier, Cyrille Bertelle, and M.A. Aziz-Alaoui. Human behaviors in the face of disasters : from observing to conceptual and mathematical modeling. *Cybergeo : Revue européenne de géographie / European journal of geography*, 735, Sep 2015.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6) :559–572, 1901.

[Pen17] Pentaho Corporation. Mondrian open source OLAP engine. Available at http ://mondrian.pentaho.org, 2017.

[Pos13] PostgreSQL. Postgresql open source object-relational database system. Available at http ://www.postgresql.org/, 2013.

[PPPS03] Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, and Constantine D. Spyropoulos. Web Usage Mining as a Tool for Personalization : A Survey. *User Modeling and User-Adapted Interaction*, 13(4) :311–372, November 2003.

[Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technology Journal*, 36 :1389–1401, 1957.

[PZW11] Georgios Pitsilis, Xiangliang Zhang, and Wei Wang. *Clustering Recommenders in Collaborative Filtering Using Explicit Trust Information*, pages 82–97. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Qam10] Ali Mustafa Qamar. *Generalized Cosine and Similarity Metrics : A Supervised Learning Approach based on Nearest Neighbors*. Theses, Université de Grenoble, November 2010.

[Qui86] J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1) :81–106, March 1986.

[Qui93] J. Ross Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[RAS07] C. Rack, S. Arbanowski, and S. Steglich. A generic multipurpose recommender system for contextual recommendations. In *Eighth International Symposium on Autonomous Decentralized Systems (ISADS'07)*, pages 445–450, March 2007.

[RIS+94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens : An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[RKR08] Al Mamunur Rashid, George Karypis, and John Riedl. Learning preferences of new users in recommender systems : an information theoretic approach. *SIGKDD Explorations*, 10(2) :90–100, 2008.

[RL14] Dr. P. Sengottuvelan R. Lokeshkumar, R. Sindhuja. A survey on preprocessing of web log file in web usage mining to improve the quality of data. *International Journal of Emerging Technology and Advanced Engineering*, 4(8) :229–234, 2014.

[RN88] Joseph L. Rodgers and Alan W. Nicewander. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1) :59–66, 1988.

[Roc71] J. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, pages 313–323. Englewood Cliffs, NJ : Prentice-Hall, 1971.

[Roy96] B. Roy. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.

[RRSK11] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[SA15] Fayrouz Soualah Alila. *CAMLearn* : une architecture de système de recommandation sémantique sensible au contexte : application au domaine du m-learning*. PhD thesis, 2015. Dijon.

[Sal83] Gerard Salton. *Introduction to Modern Information Retrieval (McGraw-Hill Computer Science Series)*. McGraw-Hill Companies, September 1983.

[Sar00] Sunita Sarawagi. User-adaptive exploration of multidimensional data. In *VLDB*, pages 307–316, 2000.

[SBFS00] C. Sapia, Prof R. Bayer, Phd Forwiss, and Carsten Sapia. Promise - modeling and predicting user query behavior in online analytical processing environments. Technical report, 2000.

[SBRJ16] Thorsten Sommer, Ursula Bach, Anja Richert, and Sabina Jeschke. *A Web-Based Recommendation System for Engineering Education E-Learning Solutions*, pages 293–302. Springer International Publishing, Cham, 2016.

[SCDT00] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web Usage Mining : Discovery and Applications of Usage Patterns from Web data. *SIGKDD Explorations*, 1(2) :12–23, 2000.

[Sch90] Michael Schrage. *Shared Minds : The New Technologies of Collaboration*. Random House Inc., New York, NY, USA, 1990.

[SDP09] Kostas Stefanidis, Marina Drosou, and Evaggelia Pitoura. "you may also like" results in relational databases. In *Proc. of PersDB 2009, in conjunction with the VLDB 2009 Conference*, 2009.

[SFHS07] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. The adaptive web. chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007.

[SGM00] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of Similarity Measures on Web-page Clustering. In *Proceedings of the 17th National Conference on Artificial Intelligence : Workshop of Artificial Intelligence for Web Search (AAAI 2000), 30-31 July 2000, Austin, Texas, USA*, pages 58–64. AAAI, July 2000.

[SGW16] Maciej J Swat, Pierre Grenon, and Sarala Wimalaratne. Probonto–ontology and knowledge base of probability distributions. *Bioinformatics*, page btw170, 2016.

[Sha17] Microsoft Sharepoint. https ://products.office.com/fr-fr/sharepoint/collaboration, 2017.

[Sil83] Norbert Sillamy. *Dictionnaire usuel de psychologie*. Bordas, 1983.

[Sil01] Barry G Silverman. More realistic human behavior models for agents in virtual worlds : emotion, stress, and value ontologies. *ACASA Technical Report*, 2001.

[SJCO06] Barry G Silverman, Michael Johns, Jason Cornwell, and Kevin O'Brien. Human behavior models for agents in simulators and games : part i : enabling science with pmfserv. *Presence : Teleoperators and Virtual Environments*, 15(2) :139–162, 2006.

[SKKR00] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender systems : A case study. In *WebKDD Workshop at the ACM SIGKKD*, 2000.

[SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[SKR01] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5(1/2) :115–153, 2001.

[SM11] YS Sneha and G Mahadevan. A study on clustering techniques in recommender systems. In *International Conference on Computational Techniques and Artificial Intelligence*, pages 97–100, 2011.

[SPUP01] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Generative models for cold-start recommendations. In *Proceedings of the 2001 SIGIR workshop on Recommender Systems*, 2001.

[SPUP02] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *25th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 253–260, New York, NY, USA, 2002. ACM.

[SSB10] Tarek Sboui, Mehrdad Salehi, and Yvan Bédard. A systematic approach for managing the risk related to semantic interoperability between geospatial datacubes. *IJAEIS*, 1(2) :20–41, 2010.

[ST71] Siegfried Streufert and Eugene A Taylor. Objective risk levels and subjective risk perception. Technical Report 40, Purdue University and Group Psychology Programs, Office of Naval Research, August 1971.

[STB+16] I. Spiecker, O. Tambou, P. Bernal, M. Hu, C. Molinaro, E. Negre, I. Wolfgang Sarlet, L. Schertel Mendes, N. Witzleb, and F. Yger. Multi-country - the regulation of commercial profiling - a comparative analysis. *European Data Protection Law Review, Volume 2*, 2016.

[Swi14] S. Swithern. Global humanitarian assistance report. New York, 2014.

[SWY75] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11) :613–620, November 1975.

[TDG+16] H. Tanghe, S. Ait Daoud, P-O Gibert, S. Sahri, S. Benbernou, J. Totel, L. Bertin, C. Diaconu, I. Ghorbel, D. Geldwerth-Feniger, L. Abidi, C. Cérin, P. Werle, and M. Lafaille. *Livre blanc : Approches contemporaines en hébergement et gestion de données.* CNRS, 2016.

[Tho02] E. Thomsen. *OLAP Solutions : Building Multidimensional Information Systems.* Wiley, 2002.

[Tin09] Nava Tintarev. *Explaining recommendations.* PhD thesis, University of Aberdeen, 2009.

[TLBD10] Lynda Tamine-Lechani, Mohand Boughanem, and Mariam Daoud. Evaluation of contextual information retrieval effectiveness : Overview of issues and research. *Knowl. Inf. Syst.*, 24(1) :1–34, July 2010.

[TM07] Nava Tintarev and Judith Masthoff. Effective explanations of recommendations : User-centered design. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 153–156, New York, NY, USA, 2007. ACM.

[Ton09] Anne-Marie Toniolo. Le comportement : entre perception et action, un concept à réhabiliter. *L'Année psychologique*, 109(01) :155–193, 2009.

[TPJ+12] Dieudonné Tchuente, André Péninou, Nadine Jessel, Marie-Françoise Canut, and Florence Sèdes. Modelisation générique du processus de développement des profils utilisateurs dans les systèmes d'information (regular paper). In *Colloque Veille Stratégique Scientifique et Technologique (VSST), Ajaccio, 24/05/2012-25/05/2012*, page (support électronique), http ://www.ups-tlse.fr, mai 2012. Université Paul Sabatier - Toulouse.

[Tsu93] S. Tsuchiya. Improving knowledge creation ability through organizational learning. In *ISMICK'93 Proceedings, International Symposium on the Management of Industrial and Corporate Knowledge*, 1993.

[Uni06] United Nations. *Global Survey of Early Warning Systems : an Assessment of Capacities, Gaps and Opportunities Toward Building a Comprehensive Global Early Warning System for All Natural Hazards : a Report Prepared at the Request of the Secretary-General of the United Nations.* 2006.

[VK14] Jenni Viitanen and Richard Kingston. Smart cities and green growth : outsourcing democratic and environmental resilience to the global technology sector. *Environment and Planning A*, 46(4) :803–819, April 2014.

[WABN14] N. Wang, M-H. Abel, J-P. Barthes, and E. Negre. Towards a recommender system from semantic traces for decision aid. In *International conference on Knowledge Management and Information Sharing, KMIS 2014, Rome, Italy*, pages 274–280, 2014.

[WABN15a] Ning Wang, Marie-Hélène Abel, Jean-Paul A. Barthès, and Elsa Negre. Mining user competency from semantic trace. In *19th IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2015, Calabria, Italy, May 6-8, 2015*, pages 48–53, 2015.

[WABN15b] Ning Wang, Marie-Hélène Abel, Jean-Paul A. Barthès, and Elsa Negre. A recommender system from semantic traces based on bayes classifier. In *Proceedings of International Conf. on Knowledge Management, Information and Knowledge Systems (KMIKS), Tunisia, April 2015*, 2015.

[WABN15c] Ning Wang, Marie-Hélène Abel, Jean-Paul A. Barthès, and Elsa Negre. Recommending competent users from semantic traces using a bayes classifier. In *2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon Tong, Hong Kong, October 9-12, 2015*, pages 1351–1356, 2015.

[WABN16a] Ning Wang, Marie-Helene Abel, Jean-Paul Barthes, and Elsa Negre. Recommending a competent person in a digital ecosystem. In *IEEE International Conference on Industrial Informatics and Computer Systems (CIICS) - Dubaï, March 2016*, 2016.

[WABN16b] Ning Wang, Marie-Hélène Abel, Jean-Paul A. Barthès, and Elsa Negre. An answerer recommender system exploiting collaboration in CQA services. In *20th IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2016, Nanchang, China, May 4-6, 2016*, pages 198–203, 2016.

[WABN17] Ning Wang, Marie-Hélène Abel, Jean-Paul A. Barthès, and Elsa Negre. Trace-based computer supported cooperative work as support for learners group design. In *21th IEEE International Conference on Computer Supported Cooperative Work in Design, CSCWD 2017, Wellington, New Zealand, May 2017*, 2017.

[Wai10] Nuwan Waidyanatha. Towards a typology of integrated functional early warning systems. *IJCIS*, 6(1) :31–51, 2010.

[Wak12] Simon Wakeling. The user-centered design of a recommender system for a universal library catalogue. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 337–340, New York, NY, USA, 2012. ACM.

[Wan16] Ning Wang. *Towards a Competency Recommender System from Collaborative Traces.* PhD thesis, Université de Technologie de Compiègne, France, 2016.

[WBC07] Ryen W. White, Mikhail Bilenko, and Silviu Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *SIGIR*, pages 159–166, 2007.

[WGC+09] Shelton Waggener, David Greenbaum, Ian Crew, Rick Jaffe, and Aron Roberts. Collaborative tools strategy. 2009.

[wik15] wikistat.fr. Introduction à l'analyse en composantes principales (acp). http ://wikistat.fr/pdf/st-l-descript-estim-intro.pdf, Mars 2015.

[Wik17a] Wikipédia. Internet des objets. https ://fr.wikipedia.org/wiki/Internet_des_objets#cite_note-ETH2015-1, 2017.

[Wik17b] Wikipédia. Technologies de l'information et de la communication. http ://fr.wikipedia.org/wiki/Technologies_de_l%27information_et_de_la_communication, 2017.

[WJ08] Lynn Perry Wooten and Erika Hayes James. Linking crisis management and leadership competencies : The role of human resource development. *Advances in Developing Human Resources*, 10(3) :352–379, 2008.

[WZGS15] Hannes Werthner, Markus Zanker, Jennifer Golbeck, and Giovanni Semeraro, editors. *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015.* ACM, 2015.

[XCC14] Ning Xu, Lei Chen, and Bin Cui. Loggp : A log-based dynamic graph partitioning method. *Proc. VLDB Endow.*, 7(14) :1917–1928, October 2014.

[ZAV10] Alexia Zoumpoulaki, Nikos Avradinis, and Spyros Vosinakis. A multi-agent simulation framework for emergency evacuations incorporating personality and emotions. In *Hellenic Conference on Artificial Intelligence*, pages 423–428. Springer, 2010.

[ZCEZM11] Raafat Zarka, Amélie Cordier, Elod Egyed-Zsigmond, and Alain Mille. Trace replay with change propagation impact in client/server applications. In Alain Mille, editor, *Ingénierie des connaissances (IC 2011)*, Sciences exactes et naturelles, pages 607–622. Publibook, May 2011.

[ZLO07] Andreas Zimmermann, Andreas Lorenz, and Reinhard Oppermann. An operational definition of context. In *CONTEXT*, volume 4635 of *Lecture Notes in Computer Science*, pages 558–571. Springer, 2007.

[ZMKL05] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 22–32, New York, NY, USA, 2005. ACM.

[ZTZX14] Mi Zhang, Jie Tang, Xuchen Zhang, and Xiangyang Xue. Addressing cold start in recommender systems : A semi-supervised co-training algorithm. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 73–82, New York, NY, USA, 2014. ACM.

Liste des tableaux

3.1 Extrait du profil de l'utilisateur *Marie* . . . 21
3.2 Extrait du catalogue de livres . . . 25
3.3 Extrait du profil de l'utilisateur *Marie* . . . 26
3.4 Coïncidence entre les caractéristiques des livres et les préférences (profil) de Marie . 26
3.5 Avantages et inconvénients liés au modèle vectoriel et aux approches syntaxiques . . 31
3.6 Types d'erreurs . . . 36

4.1 Récapitulatif des combinaisons pour le système de recommandation de requêtes OLAP 48
4.2 Protocole de communication des agents . . . 51
4.3 Evaluations initiales utilisées pour entrainer le système de recommandation. . . . 53
4.4 Valeurs de l'espace latent à partir de la matrice d'utilité, pour les apprenants. . . . 54
4.5 Valeurs de l'espace latent à partir de la matrice d'utilité, pour les ressources. . . . 54
4.6 Premières valeurs singulières à partir de la matrice d'utilité. . . . 54
4.7 Résultats initiaux et classement du système de recommandation pour l'apprenant ST_3. 55
4.8 Résultats finaux et classements de l'agent de filtrage pour l'apprenant ST_3. . . . 55
4.9 Exemple de calcul du TF-IDF pour « activité, trace, ensemble de traces » : Les traces de John dans un groupe de 10 utilisateurs . . . 60
4.10 Intervalles et valeurs pour Gotham . . . 68
4.11 Tableau récapitulatif des différentes instanciations . . . 75

5.1 Correspondance entre les scores et les niveaux de notre indicateur . . . 85
5.2 Protocole d'évaluation . . . 91
5.3 Sessions d'analyses . . . 91

6.1 Récapitulatif des informations contextuelles pertinentes selon les domaines/cas d'applications . . . 130

7.1 Tableau de synthèse des thématiques de recherche, des projets et des publications (en rapport avec les systèmes de recommandation) . . . 144

8.1 Synthèse des instanciations de notre système générique de recommandation . . . 149

Liste des figures

2.1 Démarche décisionnelle 14

3.1 Le système de recommandation basé sur le contenu vu comme une boîte noire (adapté de [JZFF10]) 24
3.2 Le système de recommandation collaboratif vu comme une boîte noire (adapté de [JZFF10]) 26
3.3 Le système de recommandation hybride vu comme une boîte noire (adapté de [JZFF10]) 28
3.4 Intersection de nos travaux selon les trois axes (nature, classe et contexte) - Visualisation en 3D 40
3.5 Intersection de nos travaux selon les trois axes (nature, classe et contexte) - Visualisation en 2D 40

4.1 Vue d'ensemble de notre système générique de recommandation 44
4.2 Vue d'ensemble de notre approche de recommandation de requêtes OLAP 48
4.3 Vue d'ensemble de notre approche à base d'agents pour les plateformes d'apprentissage numérique 50
4.4 Processus de recommandation (UML) 52
4.5 Structure de notre système de recommandation dans les environnements de travail collaboratif basé sur l'exploitation sémantique des traces. 58
4.6 Vue d'ensemble de notre approche de recommandation pour les villes intelligentes . . 64
4.7 Vue d'ensemble de notre approche de recommandation pour la gestion de crises . . . 70

5.1 Evaluation subjective et objective de la recommandation 78
5.2 Modèle d'évaluation subjective des systèmes de recommandation avec prise en compte des connaissances (adapté de [Pal07]) 81
5.3 Indicateur de perception 86
5.4 Modèle conceptuel de l'entrepôt de données spatiales relatif à la consommation énergétique 89
5.5 Requête agrégée : exemple d'une requête « LFC_q » 91

6.1 Graphes représentatifs du log selon différents types d'adjacence. 100
6.2 Graphe pondéré représentatif du log prenant en compte le séquencement et les sessions où le médoïde est relié à la première requête de chaque session. 101
6.3 Exemple de page web pour la requête q_1 104
6.4 Arbre couvrant de poids minimal fournissant la structure du site 105
6.5 Vue d'ensemble du processus de démarrage à froid. 109
6.6 Vue d'ensemble de notre système générique de recommandation avec prise en compte du contexte. 114

6.7 Catégorisation des facteurs de contexte 115
6.8 Ontologie générale de notre modélisation de contexte (cinq unités) dans le cadre d'un système de recommandation de requêtes OLAP 120
6.9 Un modèle hiérarchique d'aide à la décision multicritère vu comme une agrégation des coûts-bénéfices et du contexte 123
6.10 Des indicateurs vers les comportements 124

7.1 Architecture du système RecoOLAP 133
7.2 Représentation des compétences dans l'ontologie MEMORAe-CRS 135
7.3 MEMORAe-CRS : Exemple de recommandation - TF-IDF 136
7.4 MEMORAe-CRS : Exemple de recommandation - Classification Bayésienne 136
7.5 Plateforme web e-MEMORAe 137
7.6 Cheminement de nos travaux de recherche 145

Glossaire

AM	Arbre de Recouvrement Minimum *Minimal Spanning Tree*
ACM	Analyse des Correspondances Multiples *Multiple Correspondence Analysis*
ACO	Optimisation par colonies de fourmis *Ant Colony Optimization*
ACP	Analyse en Composantes Principales *Principal Components Analysis*
CIFRE	Convention Industrielle de Formation par la REcherche
CMMI	Modèle intégré du niveau de maturité *Capability Maturity Model Integration*
CRM	Gestion de la relation client *Customer Relationship Management*
CWE	Environnement de Travail Collaboratif *Collaborative Working Environment*
EWS	Système d'Alertes Précoces *Early Warning System*
ILS	Similarité Intra-Liste *Intra-List Similarity*
ISO	Organisation Internationale de Normalisation *International Organization for Standardization*
kNN	k plus proches voisins *k Nearest Neighbors*
LSA	Analyse sémantique latente *Latent Semantic Analysis*
MAE	Erreur absolue moyenne *Mean Absolute Error*
MEMORAe-CRS	Système de recommandation d'experts dans les CWE

MOOC	Plateforme de formation en ligne *Massive Open Online Course*
NMAE	MAE normalisée Normalized MAE
NRMSE	RMSE normalisée Normalized RMSE
NTIC	Nouvelle Technologie de l'Information et de la Communication *New Information and Communication Technology*
OLAP	Processus d'analyse en ligne *On-Line Analytical Process*
OLTP	Traitement transactionnel en ligne *On-Line Transactional Process*
OWL	Langage d'Ontologies Web *Web Ontology Language*
QSL	Langage de manipulation pour résumer des requêtes OLAP
RDF	Langage pour le web sémantique *Resource Description Framework*
RecoOLAP	Système de recommandation de requêtes OLAP
RMSE	Racine carrée de l'erreur quadratique moyenne *Root Mean Square Error*
SI	Système d'Information *Information System*
SIAD	Système d'Information pour l'Aide à la Décision
SICO	Système d'Information et COnnaissance *Information and Knowledge System*
SVD	Décomposition en Valeurs Singulières *Singular Value Decomposition*
TF-IDF	Fréquence du terme - Fréquence inverse de document *Term Frequency - Inverse Document Frequency*
TIC	Technologie de l'Information et de la Communication *Information and Communication Technology*
Ucov	couverture utilisateur *User Coverage*

Habilitation à Diriger des Recherches

Elsa NEGRE

Systèmes de recommandation : généricité, évaluation et améliorations

Résumé

Nos travaux s'articulent autour de l'extraction et de l'analyse de données issues de sources hétérogènes pour les rendre facilement accessibles et exploitables par les utilisateurs/décideurs. En effet, il devient de plus en plus difficile de savoir quelles sont les données à rechercher et où les trouver lorsque la masse de données/informations s'accroît. Des techniques informatiques existent pour faciliter cette recherche et permettre une extraction pertinente des données/informations. L'une d'entre elles est la recommandation qui guide l'utilisateur lors de son exploration, en cherchant pour lui les informations susceptibles d'être pertinentes.

Un enjeu intéressant est de proposer un système de recommandation capable de s'adapter à différents cas d'applications, avec de bonnes performances du point de vue de l'utilisateur/décideur et palliant certains manques des systèmes de recommandation existants. Dans le cadre de nos travaux, l'ensemble des données à explorer peut provenir de différents domaines (les environnements de travail collaboratif, les plateformes d'apprentissage en ligne, les entrepôts de données, les villes intelligentes, les systèmes d'alertes précoces, ...) et l'utilisateur à aider peut être un individu isolé ou une entité multiple à visée publique. Conscients que la masse de données/informations à explorer dans de tels cas peut être très importante, complexe et variée, il nous est apparu nécessaire de proposer des systèmes de recommandation appropriés pour y faire face. Nous proposons donc une approche générique de recommandation, en rupture complète avec les travaux existants, que nous instancions pour permettre de recommander soit des éléments, soit des utilisateurs, sous forme de recommandations individuelles ou à visée publique dans différents domaines. Puis, nous nous intéressons à l'évaluation des (systèmes de) recommandations. Afin d'assurer la pertinence des recommandations du point de vue de l'utilisateur/décideur, nous proposons des méthodes pour évaluer subjectivement d'une part le système de recommandation et d'autre part les recommandations retournées. Enfin, malgré de bonnes performances, parfois, les recommandations ne sont pas considérées comme suffisamment pertinentes. Nous proposons donc des techniques pour améliorer les (systèmes de) recommandations. Elles concernent l'amélioration des données d'entrée, le démarrage à froid et l'ajout de données/sources externes (notamment le contexte de l'utilisateur/décideur). Nos propositions ont été validées par la participation à différents projets ainsi que le co-encadrement de thèses de Doctorat et le suivi de travaux de Master Recherche.

Mots clés : Systèmes de recommandation, Analyse de données, Aide à la décision, Systèmes d'Information

Abstract

Our work focuses on heterogeneous data extraction and analysis in order to make it easily accessible and exploitable by users/decision-makers. Indeed, it becomes increasingly difficult to know which data to look for and where to find it when the mass of data/information increases. IT techniques exist to facilitate this research and allow relevant extraction of data/information. One of them is the recommendation that guides the user during his/her exploration, seeking for him/her information that may be relevant.

An interesting challenge is to propose a recommender system that can adapt to different applications, with good performances from user point of view and that can overcome some lacks of existing systems. As part of our work, the explored data set can come from different fields (collaborative work environments, massive open online courses, data warehouses, smart cities, early warning systems, ...) and the helped user may be an isolated individual or a multiple public entity. Aware that the mass of data/information to explore in such cases can be very large, complex and varied, it became necessary to propose appropriate recommender systems to cope. We propose a generic framework which is instantiated to recommend either items or users, as individual recommendations or public recommendations in multiple and varied fields. Then, we are interested in evaluating the recommendations. In order to ensure the relevance of the recommendations from user point of view, we propose methods to subjectively evaluate both the recommender system and the returned recommendations. Finally, despite good performance, sometimes, the recommendations are not considered sufficiently relevant. We therefore propose methods to enhance recommendations and recommender systems. They relate to the improvement of input data, cold start and the integration of external data/sources (including user context). Our proposals have been validated by participation in various projects and the co-supervisor of PhD and Master theses.

Keywords : Recommender systems, Data science/analysis, Decision support systems, Information systems