Ministry of Higher Education and Scientific Research
University of Manouba
National School of Computer Science - ENSI

THESIS

Presented for the degree of

Doctor of Computer Science

Contextual Recommender System based on Multi-Criteria Preferences

By

Rim DRIDI

LISI Laboratory, INSAT, Tunis

Defended on December 16, 2021

Jury members:

Narjes Bellamine Ben Saoud, Full Professor, ENSI Manouba	*Jury chairman*
Antoine Doucet, Full Professor, Univ. of La Rochelle, France	*Reviewer*
Ikram Amous Ben Amor, Full Professor, Enet'Com Sfax	*Reviewer*
Wided Lajouad Chaari, Associate Professor, ENSI Manouba	*Examiner*
Yahya Slimani, Full Professor, ISAMM Manouba	*Supervisor*
Lynda Tamine, Full Professor, IRIT Toulouse, France	*Co-supervisor*

Abstract

The last decade has witnessed the generation of an overwhelming amount of information online. Such a high volume of information makes it difficult for users to find their desired content (movies, music, books, etc.) in a reasonable time. There is need to filter and efficiently deliver relevant information to alleviate the information overload problem, which has created a potential issue to many Internet users. One way of solving this problem is by using recommender systems.
Recommender Systems (RS) are information filtering systems that cope with the information overload problem by filtering vital information fragment out of large amount of dynamically generated information according to user's preferences about items. RS became an important area of research, thanks to their help for users to suggest the items they might prefer, instead of crawling thousands or hundreds of items until finding the most adequate. However, there are usually various factors that may impact users preferences. Therefore, research in recommender systems is starting to recognise the importance of items multi-criteria preferences and the role of user's context in enhancing the recommendation output. In this respect, traditional recommender systems are extended to offer novel lines of research areas such as Context-Aware Recommendation Systems (CARS) and Multi-Criteria Recommendation Systems (MCRS).
This thesis investigates the inclusion of useful additional information in the recommendation process. Firstly, two novel collaborative filtering based approaches for context-aware recommendation are proposed. The first approach is based on a neighborhood-based model integrating user's inferred contextual situation in the rating prediction computing process. The second one presents a matrix factorization-based model that consists of two strategies: a weighting strategy that integrates the relevant contextual dimensions weights in the rating prediction process and an interaction strategy that incorporates the interaction measurements between correlated contextual dimensions in the rating prediction process.
Despite much of work has been done on extended recommender systems, the interesting research direction including both context-awareness and multi-criteria directions remain unexplored, where these directions are addressed separately in most existent literature. Therefore, we aim to capture more fine grained preferences to upgrade items recommendation quality by integrating users multi-criteria preferences under specific contexts. With this aim in mind, two new context-aware multi-criteria recommendation models are proposed. The first model focuses on estimating users overall ratings through predicting clustered criteria ratings then using prioritized aggregation operators as means of multi-criteria ratings aggregation. The second one addresses the prediction of users preferences through predicting clustered criteria ratings by considering the dependencies between users and contexts as well as the dependencies between the criteria.
The validation of the proposed recommendation approaches are conducted using real-world datasets, novel created datasets and popular metrics. The obtained results demonstrate that our proposals exhibit significant improvements over alternative recommendation approaches.

Keywords: recommender systems, collaborative filtering, context, multi-criteria decision, rating prediction.

Résumé

La dernière décennie a été marquée par la production d'une quantité écrasante d'informations en ligne. Un tel volume d'informations rend difficile pour les utilisateurs de trouver le contenu qu'ils souhaitent (films, musique, livres, etc.) dans un délai raisonnable. Il est nécessaire de filtrer et de fournir des informations pertinentes afin d'atténuer le problème de surcharge d'informations, qui représente un probléme potentiel pour nombreux utilisateurs d'Internet. Une facọn de résoudre ce problème est d'utiliser les systèmes de recommandation.
Les systèmes de recommandation (RS) sont des systèmes de filtrage d'informations qui traitent le problème de surcharge d'informations en filtrant les fragments d'informations essentiels parmi une grande quantité d'informations générées dynamiquement, en fonction des préférences de l'utilisateur sur les articles. RS sont devenus un important domaine de recherche, grâce à l'aide qu'ils apportent aux utilisateurs en leur suggérant les articles qu'ils pourraient préférer, au lieu de parcourir des milliers ou des centaines d'articles jusqu'à trouver le plus adéquat. Cependant, il existe généralement plusieurs facteurs susceptibles d'influencer les préférences des utilisateurs. Par conséquent, la recherche sur les systèmes de recommandation commence à constater l'importance des préférences sur les multi-critères et le rôle du contexte de l'utilisateur dans l'amélioration des résultats de recommandation. À cet égard, les systèmes de recommandation traditionnels sont étendus pour offrir des nouveaux axes de recherche tels que les systèmes de recommandation tenant compte du contexte (CARS) et les systèmes de recommandation multi-critères (MCRS).
Cette thèse étudie l'inclusion des informations supplémentaires utiles dans le processus de recommandation. Tout d'abord, deux nouvelles approches basées sur le filtrage collaboratif pour la recommandation contextuelle sont proposées. La première approche est basée sur un modèle de voisinage qui intègre la situation contextuelle inférée de l'utilisateur dans le processus de calcul de prédiction des notes. La deuxième approche est basée sur la factorisation matricielle et consiste en deux stratégies : une stratégie de pondération qui intègre les poids des dimensions contextuelles pertinentes dans le processus de prédiction des notes et une stratégie d'interaction qui intègre les mesures d'interaction entre les dimensions contextuelles corrélées dans le processus de prédiction des notes.
Malgré plusieurs travaux ont été réalisés sur les systèmes de recommandation étendus, la direction importante de recherche comprenant à la fois les directions sur la prise en compte du context et des multi-critères reste inexplorée, ces directions sont traitées séparément dans la plupart des publications existantes. Par conséquent, nous visons à capturer des préférences plus fines pour améliorer la qualité de recommandation d'articles en intégrant les préférences des utilisateurs sur les multi-critères dans des contextes spécifiques. Dans cette optique, deux nouveaux modèles de recommandation multi-critères sensibles au contexte sont proposés. Le premier modèle permet d'estimer les notes globales des utilisateurs en estimant les notes des critères groupés puis utilise des opérateurs d'agrègation prioritaires comme moyen d'agrègation des notes multi-critères. Le deuxième aborde la prédiction des préférences des utilisateurs en prédisant les évaluations des critères groupés en considérant les dépendances entre utilisateurs et contextes ainsi que les dépendances entre les critères.
La validation des approches de recommandation proposées est effectuée à l'aide des ensembles de données du monde réel, de nouveaux ensembles de données créés et des métriques populaires. Les résultats obtenus démontrent que nos propositions

présentent des améliorations significatives par rapport aux autres approches de recommandation.

Mots-clés: systèmes de recommandation, filtrage collaboratif, contexte, décision multi-critères, prédiction des notes.

Acknowledgements

Science, my lad, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth.

Jules Verne, *A Journey to the Center of the Earth*

This manuscript is the result of several years of work in the college, engineering, and finally PhD studies. In all this time I have received the support of my family, friends, teachers and colleagues, to whom I am very grateful. These lines are a sign of my gratitude to all of them.

First, I would like to express my sincere thanks and my deep gratitude to my principal advisor Professor Yahya Slimani for introducing me to the area of research, and for his availability and his real interest in this work. I will forever be grateful for his invaluable expertise, full support and helpful assistance throughout the journey of this research project. Without his incredible patience and timely wisdom and counsel, my thesis work would have been a frustrating and overwhelming pursuit.

I also express my appreciation to my co-supervisors Dr. Khedija Arour and Dr. Saloua Zammali for their constant encouragement at any aspect I had to face these years, from the stressful talks to the (sometimes confusing) reviews received, and including those about the overwhelming first classes as a teacher and the tiresome paperwork. Thanks to Mrs Khedija for giving me the opportunity to pursue this PhD with her, for her effort and continuous work during these years, and for allowing me to participate in conferences where I have learnt so much about deadlines and priorities, and how to deal with them according to a given purpose. Special thanks also to Mrs Saloua whose continuous support helped me to improve my writing and design skills. Besides, I started my research training with her, which let me grow professionally, and represented my first satisfactions in the form of papers.

Professor Lynda Tamine deserves a special mention for giving me the opportunity to collaborate with her and for accepting my request for the internship in her research group at Toulouse Computer Science Research Institute. I thank her for her numerous re-readings, her advice and for many constructive discussions which enabled me to achieve an important research project that I present in this thesis. I am so lucky to have such a kind, modest, gentle and talented mentor.

Finally, I would like to thank all the reviewers of my thesis for their valuable comments to improve my research. I also thank all members of the jury for having kindly agreed to evaluate my work.

This thesis is dedicated to :

all my family members,

my beloved parents,

all the people I love.

Thank you all for your unconditional love, fruitful support and continuous encouragement.

List of Publications

In the following international journal and conference papers, we presented large portions of contributions, descriptions, results and conclusions related to this thesis:

Journal papers:

- R. Dridi, L. Tamine, Y. Slimani, Exploiting context-awareness and multi-criteria decision making to improve items recommendation using a tripartite graph-based model, Information Processing Management (2021). **Impact Factor (IF): 4.787** [under review].
- R. Dridi, S. Zammali, T. Alsulimani, K. Arour, Effective rating prediction based on selective contextual information, Information Sciences 510 (2020) 218 - 242. **Impact Factor (IF): 5.910**.

Conference papers:

- R. Dridi, L. Tamine, Y. Slimani, Context-aware multi-criteria recommendation based on spectral graph partitioning. In: Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, 705 August 26-29, 2019, pp. 211 - 221. **(rank B)**.
- R. Dridi, S. Zammali, T. Alsulimani, K. Arour, An improved context-aware matrix factorization model incorporating fuzzy measures. In 2018 IEEE international conference on fuzzy systems, FUZZ-IEEE 2018, Rio de janeiro, Brazil, July 8-13, 2018, pp. 1 - 8. **(rank A)**.
- R. Dridi, S. Zammali, K. Arour, Fuzzy rule-based situational music retrieval and recommendation. In 14th IEEE/ACS international conference on computer systems and applications, AICCSA 2017, Hammamet, Tunisia, October 30 - Nov. 3, 2017, pp. 549 - 556. **(rank C)**.

- R. Dridi, S. Zammali, K. Arour, Situation-aware rating prediction using fuzzy rules. In: Proceeding Knowledge Science, Engineering and Management - 9th International Conference, KSEM 2016, Germany, 2016, pp. 209 - 221. **(rank B)**.

Contents

List of Figures

List of Tables

List of Abbreviations

CAMF	Context-Aware Matrix Factorization
CARS	Context-Aware Recommender System
CBF	Content-Based Filtering
CF	Collaborative Filtering
CMF	Collective Matrix Factorization
FWR	Fuzzy Weighting Recommender
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
MCDM	Multi-Criteria Decision Making
MCRS	Multi-Criteria Recommender System
MF	Matrix Factorization
NDCG	Normalized Discounted Cumulative Gain
ODP	Open Directory Project
RMSE	Root Mean Squared Error
RS	Recommender System
SF-IUF	Situation Frequency Inverse User Frequency
SGD	Stochastic Gradient Descent
SVD	Singular Value Decomposition
SVR	Support Vector Regression
WS4J	WordNet Similarity for Java

Introduction

Background and Motivations

Nowadays, the Internet has led to an exponential growth of unlimited amount of information and content that outgrow the capacity of users to process it.

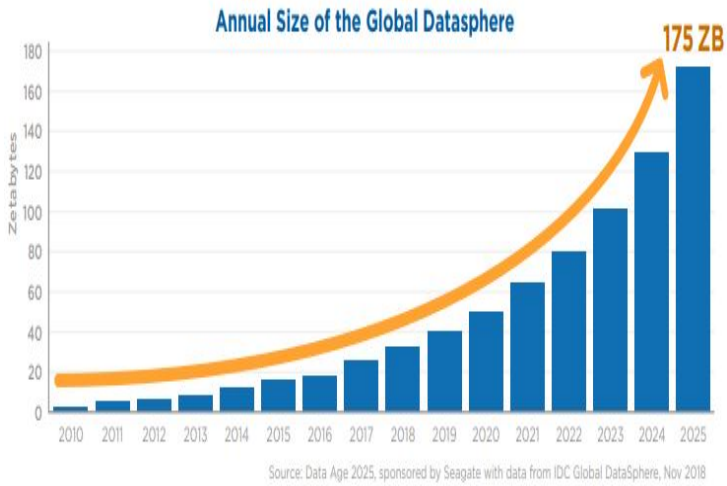


FIG. 1.: The global datasphere: the growth of the created data from 2010 to 2025 [1]

In accordance with the International Data Corporation (IDC) [1], people are more and more dependent to data in these last years in nearly all their life aspects such as entertainment, education and their relationships with others. It is apparent from Figure 1, the increasing growing of the "data existence" (also called datasphere), where IDC anticipates that the global datasphere will grow from 33 Zettabytes (a trillion gigabytes) in 2018 to 175 Zettabytes by 2025. This abundance of information induces a keen interest in research fields and technology that could overcome this information overload problem by finding the right information. When the user knows what to look for, search engines are able to search on-line records for finding relevant data to respond to user queries. However, the problem starts when the user doesn't know what to look for and doesn't want to spend a lot of time going through big amount of data, in the hope of falling on the right item. Although search engines can distinguish relevant items from a specific query, they are not enough to cope with the information overload problem. In fact, search engines lack the ability to find out the user interest level within the relevant documents set. In this respect, another way of interacting with the available data without counting on user input has been emerged, it simply shows the items that the user might be interested in: Recommender systems. Recommender Systems (RS) are intended to solve the information overload problem by providing suggestions for items to be of use to a user. They add so much value to search engines that major search engines now integrate a recommendation engine to customize their results. For similar reasons, since the mid-1990s, RS have developed into an important area of academic and corporate research, where researchers and practitioners focus on the recommendation issues which explicitly depend on the user-supplied ratings as a means of capturing user's preferences for different items. These ratings can be further employed for recommending items to different users.

Most of conventional recommender systems only rely on the interactions between users and items (e.g., the user-item ratings). Nevertheless, in practice, more data is available beyond the user-item interactions that could be valuable for improving the recommender systems effectiveness. In this respect, users interests become heterogeneous, multiple, dynamic, and even contradictory, and then can be affected by various factors. As a result, researchers explored how to integrate additional useful information in a recommendation process, hoping that the recommendation system can then generate more suitable recommendations. The data describing the circumstances in which the users-items interactions have taken place is known as contextual information. To give a concrete idea of what contextual information is and how it can impact the user's recommendations, we provide the following example. For instance, consider a user who wants to have dinner. A recommender system may give all the restaurants that match the user's taste. However, by considering additional contextual factors, the recommender may improve its suggestions by taking into account the restaurant opening hours to ensure that it is still open or its location to verify if it is nearby. In this example, it is apparent that the time and location contextual dimensions play a crucial role for providing more accurate recommendation. For this purpose, the first open issue that inspired the research reported in this dissertation can be seen as the problem of context-aware recommendation.

Another key aspect, in recommendation approaches, is to make use of the multiple preferences for the different aspects of a given item which stand for the item multi-criteria ratings. For example, when recommending a restaurant, the preferences for the food quality, the cleanliness, and the service speed are considered as the restaurant multi-criteria ratings. As a result, several researches have started to extend standard recommendation approaches by integrating multi-criteria ratings in seeking to improve the quality of recommendation. Yet, nearly all the multi-criteria recommendation approaches ignore the impact of the contextual information on user's judgments. Thus, there is a need to develop more accurate solutions for more improved recommendations. Several aspects need to be considered for developing these solutions. These aspects refer to the inclusion of both multi-criteria and contextual information into the recommendation process to enhance the recommendation results.

Research Questions & Objectives

The main goal of the research presented here is to alleviate the existing shortcomings to produce more effective recommendation. In order to achieve this goal, we aim to extend traditional recommendation approaches by considering additional useful information including criteria preferences and contextual information. In the following, we set up the primary Research Objectives (RO) addressing the main research questions guiding this work:

- *What are the challenges and limitations that can be found in recommendation systems ?*
  RO1: Investigate and analyze the existing recommender systems from the literature.

  This research question addresses the recommendation system as a whole, we thus need to develop an in-depth study for investigating and analyzing the different existing recommender systems from the literature starting from the main traditional recommender systems to the extended ones to find out the faced challenges. In fact, we aim to review the different algorithms and techniques used in existing recommender systems and discuss the most raised issues, that have to be tackled before designing a recommender system.

- *How to effectively integrate the contextual information into a predictive model for improved recommendation?*
  RO2: Identify the pertinent contextual dimensions affecting the user's ratings about an item in a significant way and using these dimensions for providing a high-level abstraction of the context to determine the user's contextual situation.
  To address the context awareness, we need to investigate the contextual information on which users interests depend. To accomplish this task, we shall identify the appropriate contextual dimensions that effectively impact the user's preferences by considering two properties: the contextual dimensions relevancy and the contextual dimensions correlations. This task makes it possible

to distinguish relevant and correlated contextual dimensions. Following this, these contextual dimensions can be interpreted into a high-level abstraction of the context to determine contextual situations. Since the user preferences represent the main thrust behind a recommender system's suggestions, we aim to put forward predictive models integrating relevant and correlated contextual dimensions as well as the inferred contextual situations to achieve the natural goal of a recommender system through predicting contextual preferences.

- *How to model the available multi-dimensional data to jointly consider context and criteria information in recommendation ?*

  RO3: Explore new forms of modeling the multi-dimensional data by examining the recommender's data from the graph theory based perspective.

  To integrate the context and criteria information in a recommendation process, we opt for modeling the input data by presenting the associations between two types of entities (users situational contexts and criteria) as a bipartite graph. Dealing with different context types would give insight into the relevance of extending the bipartite graph for modeling different context nodes including a set of contextual dimensions values representing the users contextual situations. As a result, we aim to model the three relevant entities arising from the recommendation data as a tripartite graph including users, contextual situations and criteria. We also attempt to underline a new challenge through the tripartite graph representation, consisting in weighting differently the interactions between the three mentioned entities.

- *How can the incorporation of context and criteria information help improve the prediction accuracy for items ?*

  RO4: Design new strategies for predicting items ratings based on clustering contextually similar users evaluating items with respect to multiple criteria.

  To address this research question, we opt for propounding two predictive models according to the two different forms of modeling the multi-dimensional data. Two research hypotheses based on the recommendation entities and their relationships will be posed to give insights about the desired graph co-clustering structure for each model. The first model can rely on two-order bipartite graph co-clustering for jointly partitioning users situational contexts and the rated items criteria entities. The obtained co-clusters can provide criteria predicted ratings that will be aggregated to obtain the overall item rating by employing prioritized aggregation operators. For the tripartite graph-based model, the two-order co-clustering performed in the first model should be replaced by a high-order co-clustering represented as the fusion of pair-wise co-clustering sub-problems over two bipartite graphs. For producing items predicted ratings, a novel prediction algorithm will be proposed to consider the dependence between users and their contextual situations in a low dimensional space, and also emphasize the correlation between the different criteria.

- *How to effectively evaluate the contextual proposed approaches?*
  RO5: Test whether the consideration of the appropriate contextual information can improve the performance of recommendations through conducting a series of different experiments.
  To ensure that the proposed contextual approaches can improve the rating prediction accuracy and the recommendations performance, different experiments on both the rating prediction and the top-N recommendation tasks will be conducted. This evaluation will be done by performing offline experiments using real-world contextual datasets and also new large created contextual datasets in a first step. In a second step, online experiments will be conducted through an online survey with real users.

- *Can the contextual multi-criteria proposed models provide better results than the state-of-the-art recommender systems?*
  RO6: Validate the contextual multi-criteria predictive models with specific datasets incorporating context and criteria information.
  At first, a methodological study for the experimental setting should be performed in such a way that appropriate datasets, metrics and baseline methods are used. Each proposed model should be evaluated by conducting different experiments including the research hypothesis validation, the parameters tuning and the comparison with different recommendation models on specific contextual multi-criteria datasets.

Contributions

Our goal is to expose the reader to novel views of the recommendation problem capturing real-world challenges which have been largely overlooked by the recommendation literature. The work presented in this dissertation can be used as a stepping stone to transition to the new generation of real-world recommendation systems.
The main contributions of this thesis can be expressed as following:

- *Systematic review of the literature on recommender systems.* We first start by introducing the reader to an overview of recommender systems and explain how basically these systems work. Accordingly, we present the basic concepts and formulate the recommendation problem. We then review the literature pertaining to the main techniques used in recommendation systems and cite the existing challenges and limitations to find out new perspectives related to the proposed research. We also give an overview on performance evaluation methodologies. After the general introduction to recommender systems, we trace the evolution of the recommendation problem by considering recent emerging trends. Thus, we concentrate on context-awareness, situation-awareness and multi-criteria decision making in recommender systems and we discuss the main existing approaches in these areas.

- *Contextual dimensions weighting.* Since it was proven that the degree of incorporating a contextual dimension in the recommendation process can affect the accuracy of predicting user's preferences, we address the issue of determining which contextual dimensions that can truly impact the decision-making process. For this task, we propose a weighting method on the basis of contextual dimensions fuzzy measures. Therefore, we study the tuning of the fuzzy measure values that should be attributed to each contextual dimension and each subset of dimensions. This task has the advantage of facilitating the interpretation of relevant and interacted contextual dimensions.

- *Contextual situation inference.* The contextual situation inference is particularly related to the interactions between the different contextual dimensions. Therefore, we aim at collecting the correlated contextual dimensions to create a contextual situation. In fact, integrating fused dependent contextual dimensions where a positive interaction exists between them, could further improve the performance of recommendation. For this task, we use fuzzy logic which contains an inference engine based on a set of formulated rules, that enables to infer user's current situation from user's current context. Precisely, to determine the contextual situation, we define some IF-THEN rules. Within each rule, the antecedent side (the IF part) is composed of only correlated contextual dimensions and the contextual situations are on the consequent side.

- *Context-aware rating prediction proposals.* To fully capture the influence of relevant contextual dimensions and their interaction on items ratings, we propose two improved rating prediction models based on collaborative filtering techniques, involving relevant and dependent contextual dimensions:

  1. A neighborhood-based model: this model mainly relies on selecting the neighbors properly, based on their similarity to the current user under the same contextual situations. Then, a combination of the neighbors ratings is employed as the basis for rating prediction or recommendation. Particularly, the proposed neighborhood-based model exploits the inferred

contextual situation in which the user is involved for computing items ratings prediction.

2. A matrix factorization-based model using two strategies:
   - A weighting strategy: according to their degrees of importance, the contextual dimensions influence the items ratings differently. For this reason, we incorporate the obtained weight values of relevant contextual dimensions in the rating prediction process.
   - An interaction strategy: many useful interactions may exist between the different contextual dimensions and it is of great importance to take them into consideration. For this purpose, we integrate the obtained weights of correlated contextual dimensions in the rating prediction process.

- *Large contextual datasets building.* One well-known difficulty of research in context-aware recommendation is the relative rarity of large datasets. Therefore, two large contextual datasets are constructed to upgrade the performance of our proposed models in large scale system. The construction datasets task is represented as an enrichment process of large non-contextual datasets based on a contextual dataset and operates through three steps : (i) extracting categories; (ii) computing the similarity between categories; and, (iii) creating large contextual datasets.

- *Situational contexts and item criteria modeling.* To deal with users situational contexts and the rated items criteria, we highlight a new challenge through the bipartite graph representation.

- *Users, contextual situations and item criteria modeling.* We examine the recommender's multi-dimensional data from the graph theory-based perspective by representing the interconnected entities using a tripartite graph. Such representation is an extent of the previous bipartite graph to deal with additional nodes for modeling the relationships between users, the contextual situations in which these users are involved and the rated items criteria. We also highlight a new challenge through the tripartite graph modeling, including weighting differently the three mentioned entities connections.

- *Context-aware multi-criteria rating prediction proposals.* We proposed novel context-aware multi-criteria approaches that explore the idea of partitioning the graphs modeling the recommender's data to obtain co-clusters to be considered for the rating prediction process:

1. A bipartite graph-based model: in this model, we simultaneously partition the bipartite graph using a spectral graph co-clustering method to obtain co-clusters of users situational contexts and rated items criteria. Then, these co-clusters are used to provide predicted criteria ratings that are needed for computing the overall item rating through prioritized aggregation which tailor the criteria strengths to the users preferences.
2. A tripartite graph-based model: we begin by making an assumption to give insights about dealing with the tripartite graph partitioning. This assumption motivates us to employ a high order co-clustering offering more personalized suggestions. Particularly, the tripartite graph is treated as two dependent bipartite graphs sharing the same central type. Therefore, the high-order co-clustering problem is modeled as two pair-wise partitions for two sub-problems of co-clustering with the constraint of the triplet structure. The aim of this partitioning is to obtain the desired co-clusters of contextually similar users evaluating similar criteria. Then, for predicting cluster-based multi-criteria ratings, we use a splitting approach to consider the relationship between contexts and users in a low dimensional space, and we also underline the interactions among criteria using a correlation-based rating prediction algorithm.

Thesis Organization

The organization of the chapters corresponding to the research objectives is shown in the Table 1. Following which, a brief summary of each chapter is presented.

Research Objectives	**Thesis Chapters**
RO1: Investigate and analyze the existing recommender systems from the literature.	Chapters 1 and 2
RO2: Identify the pertinent contextual dimensions affecting the user's ratings about an item in a significant way and using these dimensions for providing a high-level abstraction of the context to determine the user's contextual situation.	Chapter 3
RO3: Explore new forms of modeling the multi-dimensional data by examining the recommender's data from the graph theory-based perspective.	Chapter 4
RO4: Design new strategies for predicting items ratings based on clustering contextually similar users evaluating items with respect to multiple criteria.	Chapter 4
RO5: Test whether the consideration of the appropriate contextual information can improve the performance of recommendations through conducting a series of different experiments.	Chapter 5
RO6: Validate the contextual multi-criteria predictive models with specific datasets incorporating context and criteria information.	Chapter 6

TABLE 1: Mapping thesis chapters to research objectives

This dissertation is composed of six chapters which are organized as follows:

- *Chapter 1*: provides an overview of recommender systems and explains how basically these systems work. Thus, it presents the preliminaries related to RS, the definition of the recommendation problem and the main recommendation techniques as well as their principal challenges and limitations to find out new research perspectives. Finally, a more detailed look is taken at the performance methodologies used to evaluate the recommendation quality.

- *Chapter 2*: reviews the work related to this thesis as a whole. Work specifically connected to our proposed approaches introduced in the next two chapters. Focusing on the input data integrated in the recommendation process, a novel classification based on context-awareness, situation-awareness and multi-criteria decision making is introduced. Following this, the recommender systems area is subdivided into sub-areas. Various relevant recommendation studies from the literature in each sub-area are reported and analysed. Finally, a brief synthesis discussing the literature is presented.

- *Chapter 3*: concentrates on the impact of context-awareness and situation-awareness aspects on the recommender systems results and shows how to extend existing knowledge in the recommendation field to put forward effective contextual approaches. Therefore, we define the multifaceted concepts of context and then introduce the proposed contextual recommendation approaches that predict items ratings according to the user's contextual information. Particularly, we make use of the user's contextual information in two different ways. On one hand, we adopt a weighting method on the basis of fuzzy measures which has the role of identifying the contextual dimensions relevancy as well as the contextual dimensions correlations. On the other hand, we employ fuzzy logic in order to infer user's contextual situation through fusing relevant and correlated contextual dimensions. The obtained outputs from the two mentioned manners of exploiting the contextual information are used for producing items predicted ratings. This task is achieved by applying two well-known collaborative-filtering based methods, which gave rise to two novel prediction models : a neighborhood-based model and matrix factorization-based model.

- *Chapter 4*: clarifies how to unify jointly user's contextual information with items criteria information in one recommender. Therefore, in this chapter, we introduce the proposed context-aware multi-criteria recommendation approaches that attempt to improve the recommendation quality by considering users multi-criteria ratings under specific contexts. Accordingly, we describe how our proposals deal with the current challenging problems and contribute to the existing body of knowledge starting from modeling the multi-dimensional input data up to producing items predicted ratings. To reach our targets, a set of techniques that consider the positive impact of integrating both context-awareness and multi-criteria decision making directions into the recommendation process are proposed.

- *Chapter 5*: presents the experimental set-up and the carried out experiments for evaluating the effectiveness of the proposed context-aware recommendation approaches. Particularly, the obtained results from the evaluated tasks of rating prediction accuracy and top-N recommendation performance on both real-word available and large created contextual datasets are reported and discussed.

- *Chapter 6*: reports the conducted series of experiments for testing the effectiveness of our context-aware multi-criteria proposed models regarding a set of models discussed in the literature review. For each proposed model, two main parts of the evaluation experiments are presented, where the first one concerns the experimental setting and the second entails the comparison results discussion.

Chapter 1

Overview on Recommender Systems

1.1 Introduction

Everyday life is saturated with decisions to take: which article to read? which movie to watch? which item to buy?; and the list goes on and on. Getting the right decision looks like finding a needle in a haystack. In fact, with the overwhelming amount of information, the user faces a dilemma: there is a boundless pool of available choices but he is not able to make the exact decision. The new technologies gave to online users the access to a mass of various types of content. For instance, many popular services possess and share a huge amount of content such as Spotify[1] (music), Google Scholar [2] (scientific articles), or Netflix [3] (TV shows and movies on demand) and so on. This results a keen attention in research fields that can help address this information overload problem [3] and facilitate the process of information seeking. The most distinctive research field developed over the last decade is Recommender Systems (RS) [4].

A variety of recommendation algorithms have been developed through the past two decades in many domains, such as movies, music, books, products, restaurants, persons (online dating), etc. Recommender systems exist with the goal of helping users in finding their way through a large catalog by identifying useful personalized items, where an item is the general term employed to indicate what the system recommends. To generate recommendations, the information about the users interests, the items or combinations of these are considered. The output is generally either a list of the top recommended items, or a probability depicting a prediction about which is the most needful item to a user.

[1] https://www.spotify.com

[2] https://scholar.google.com

[3] https://www.netflix.com

We are not able to cover the entire recommendation field in this chapter, since there are many diverse works in this topic, but we will give a brief overview of the main traditional recommenders. We start by defining the recommender systems and their basic concepts. Next, the recommendation problem is described formally. We then review the literature pertaining to the three main techniques of recommendation systems and cite the existing challenges to find new perspectives related to the proposed research. Finally, we take a more detailed look at evaluating the performance of recommender systems.

1.2 Foundations of Recommender Systems

1.2.1 Recommender Systems Definition

At the very beginning it is essential to comprehend what a recommender system is, what types of functionalities do recommender systems have, and what key notions are included. Recommender systems concept was first appeared at the end of the 1990s as an independent research area issued from different other areas such as information retrieval, approximation theory, consumer modeling, management sciences and also cognitive science [5]. Resnick and Varian [4] proposed a simple definition for recommender systems expressed as the following: *A recommender system is a system able to suggest items to users.* More abstractly, *a recommender system is also able to offer new content that interests a user from a wide range of choices* [4] and hence, it copes with the information overload problem. According to Herlocker et al. [6], *a recommender system aims at predicting which items might match the preferences of given users.* Note that this description concerns the prediction aspect of recommender systems, however the approach of Resnick et al. focuses more on real-world recommendation concept. In another point of view, Burke and colleagues [7] described a recommender system as: *Any system that generates individualized recommendations and guides the users in a personalized way to suitable items within a large space of data.* Burke's definition includes novel notions like personalization to show the ability of recommendation systems to be customized alongside their strength to generate relevant recommendations. Researchers deemed in their article [8] that recommendation is linked to four core features. These features are crucial because they cover the necessary requirements of users facing enormous set of items: Decide, Compare, Explore and Discover. This is summarized as follow:

- `"Help to decide"`: predict an item rating for a user.

- "Help to compare": display a list of personalized ranked items for a user.
- "Help to discover": provide new interesting items that might correspond to the tastes and needs of a given user.
- "Help to explore": afford items related to a particular item.

Ricci et al. [9] suggested the following popular definition: *Recommender systems are software tools and techniques providing items to be useful for a user.* For Gavalas et al. [10] *a recommender system aims to engage user profile and filtering techniques to predict the rating that a user would give to an item.* In a different way, authors in [11] assumed that recommender systems output is a list of items and based on this assumption they viewed *a recommender system as a system that generates a ranked items list to a given user in accordance with items relevance scores.* In e-commerce industrial discipline, Polatidis et al. [12] stated that *recommender systems are considered as computer algorithms used to propose items to a user, like what product to buy, restaurant to try out, or movies to rent.*

Based on these definitions, we derive two primary recommendation tasks: (i) the rating prediction task and the (ii) top-N recommendations task. These tasks will be detailed in a subsequent subsection.

In summary, almost all the above definitions have mentioned three key notions that we will present in the following: user, item and rating.

1.2.2 Recommender Systems Basic Concepts

Some notions are common in the recommendation domain, the *User* who has done some actions and is to be recommended to, then the *Item* that needs to be recommended, and finally the *Rating*, which represents how much a user is interested in an item. So the tuple (User, Item, Rating) is the core of recommender systems.

- **User.** The term *User* is used to depict the set of entities to which recommendations will be given, regardless of whether they describe a person, a group of people, or other entities of interest. In our work, users designate the persons to whom items are suggested, generally presented using attributes such as the id, name, gender, age, etc. These information are modelled as "user profile" aiming to identify the user's needs for providing custom recommendations which could be suitable for the user.

  Ordinary users having a sufficient number of ratings have been distinguished from particular ones who require a special reasoning to satisfy all users needs.

In this regard, three types of particular users are identified [13]: (i) "cold start users" are the new users who have recently entered the system with very limited information (insufficient ratings); (ii) "grey sheep users" are the users with unusual tastes resulting low correlations with other users; and (iii) users whom do not have any behavior in the current context.

- **Item.** Items are objects to be recommended to users, regardless of their actual representation. Generally, typical recommended items are documents, music, movies, etc. An item can be characterized by its features or descriptions and utility (positive if it is beneficial for the user and negative if not) [9]. In particular, the paper [9] describes an item in a movie recommender system through the following attributes: title, length, genre, director and release year.

- **Rating.** We denote the preference of a user toward an item as a rating. In our study of recommendation systems, we take user's rating to be the quintessential piece of information utilised to indicate the user's interest about an item. From this point of view, the rating presents the interaction between a user and the recommender system aiming to infer the user's opinion. We equate higher ratings with a greater preference (i.e. users would like better an item rated 5 rather than an item rated 2). As claimed by [14], a rating can be viewed in different forms: (i) *binary rating*, that shows whether a given item is good for a user or not. As an exemplification, in YouTube [4] "like" and "follow" could be considered as binary ratings. While, binary rating is easy for the user to deal with and less ambiguous, it can not be sufficient for items comparison; (ii) *numerical rating* uses a numerical scale rating aiming to provide detailed feedback. Take Netflix as an example, it uses standard five-star rating scale to power its review system and recommendations. There are also variations like using a ten-star scale; and (iii) *ordinal rating* is basically used to clarify the meaning of each rating level with words such as 1/5 stars means "I do not like very much" and 5/5 stars means "I really like" [15].

1.2.3 Formulation of the Recommendation Problem

Distinct formulations of the recommendation problem have been introduced, among the most cited formulations is the one introduced in the overview of Adomavicius et al. [16]. In that work, the recommendation problem can be formally defined as follows: Let U be a set of users and I be a set of items that can be recommended. Let $G(u, i)$ be the utility function measuring the gain of usefulness of $i \in I$ for user $u \in U$. Thus,

[4] https://www.youtube.com

the recommendation problem aims to choose the item $i^{max,u} \in I$ for user $u \in U$ which maximize the utility function G as:

$$\forall u \in U, i^{max,u} = \arg\max_{i \in I} G(u, i) \tag{1.1}$$

The formulated problem, known as the rating prediction problem, has been broadly studied for different applications and became the standard one considered when building and evaluating recommendation systems.

Definition 1.1. Rating prediction. The rating prediction problem specifies that the task of a RS is to predict the rating of an item for a specific user through a utility function, i.e., estimating the preference that a user has for an item.

The aim in the prediction phase, is to define a utility function where the prediction error representing the difference between actual ratings and predicted ratings is expected to be minimized for all observed ratings. This task is greatly studied and plenty of researches deal with this issue. Yet, we cannot restrict the recommendation function to the rating prediction task. Since the predicted ratings could be adopted to generate a top-N recommendation list by selecting the highest rated items.

Definition 1.2. Top-N recommendation. The top-N recommendation problem notes that the task of a RS is to predict whether the user will select an item or not through a utility function, i.e., estimating the pertinence of an item for a user.

In both formulations, the core of recommender systems lies in properly defining the utility function.

1.3 Traditional Recommender Systems

Traditional recommendation approaches, also referred to as traditional recommendation methods or algorithms, are expected to predict the utilities of items for target users and offer accurate recommendations. It is possible to classify RS approaches by various ways in accordance with different criteria including the type of feedback they use (explicit or implicit), the recommendation task they address (rating prediction or top-N recommendation), etc. As shown in the figure 1.1, the most common classification used in the literature is based on the type of data exploited for recommendation and establishes the following three categories:

- *Content-Based Filtering (CBF) approaches.* These approaches make use of knowledge related to users or items to provide recommendation;

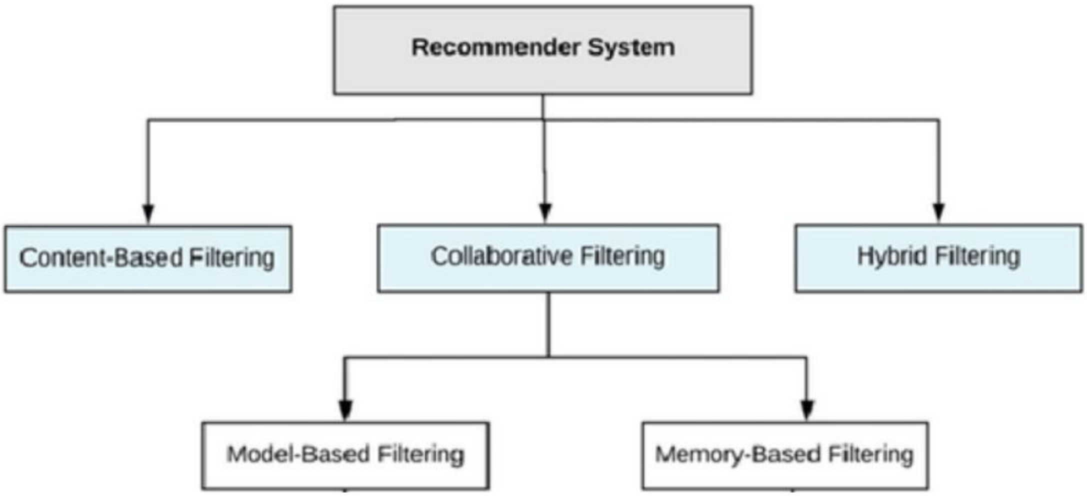


FIG. 1.1.: Recommendation approaches.

- *Collaborative Filtering (CF) approaches.* These approaches recommend items relying on similar users and their ratings;
- *Hybrid approaches.* These approaches combine the two above-mentioned filtering approaches.

Among the presented recommendation approaches, collaborative filtering approaches are selected as baselines due to their successful results so far, and extended with novel proposed techniques for a better recommendation performance.

1.3.1 Content-Based Filtering

Content-based filtering (CBF) recommender systems are based on content information about users or items to provide recommendations. This information can take different forms like features, textual descriptions, and tags. In other words, users receive items suggestions that are similar to those they positively evaluated in the past. Particularly, recommendations are made through matching the user profile features describing the user's preferences with the items features.

In content-based recommender systems, the item can be represented by a weighted terms vector extracted from its content. To define the user profile, CBF mostly concentrate on the model of the user's preference or the history of the user's interaction with the recommender.

Pandora Music Genome Project [5] is an example of a content-based approach that uses the characteristics of a song or a singer to capture the essence of music with similar

[5] https ://www.pandora.com

characteristics and to organize them. Users' feedbacks (likes or dislikes) are adopted to filter the music station's results.

We present in the figure 1.2, an example of a content-based filtering scenario that includes three users: User 1, User 2, and User 3 (the target user) and four items: Product 1, Product 2, Product 3, and Product 4. Given the fact that the target user has rated in the past the two items (Product 2 and Product 3) and after analysing the products' attributes, Product 1 is predicted to be recommended to the User 3 since it is similar to Product 3.

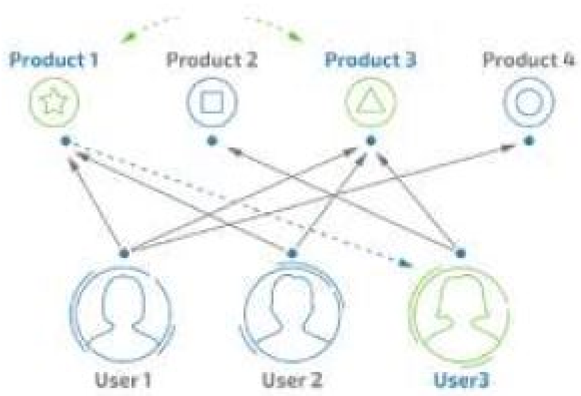


FIG. 1.2.: A content-based filtering example.

Basically, a content-based recommender system comprises the following steps [17]:

1. Preprocessing of items content (e.g. Web pages, documents, product descriptions, etc.) to extract structured pertinent information (e.g. Web pages represented as keyword vectors).

2. Starting from items liked or disliked in the past, the profile of a target user is learned through machine learning techniques.

3. Matching the profile representation of the target user and that of items to be recommended using similarity metrics.

4. Recommending a ranked list of potentially pertinent items.

This technique presents advantages such as user independence, since CBF systems only use ratings of the active user to build the recommendation model. Additionally, when a new item appears and has not yet been rated, CBF systems are able to recommend it. However, CBF suffer from several issues such as the over-specialization, as they are not capable of finding unexpected items: the user will receive recommendations of items similar to the ones rated before.

1.3.2 Collaborative Filtering

To date, collaborative filtering (CF) is the most popular algorithm used to design various applications and sites for recommender systems such as Facebook [6], Twitter[7], Google [8], LinkedIn [9] and Netflix. The underlying idea behind CF is that users with common interests in the past are more likely to keep exhibiting similar interests in the future. The principal property to work with collaborative filtering are the ratings given by users for items. Therefore, the typical input of CF recommender systems is represented by a matrix of ratings representing users by rows and items by columns. More precisely, the user-item matrix defining users' preferences for items is used to find like minded users by computing similarities between their profiles defining a "neighborhood" to provide recommendations.

As an example, we show in the figure 1.3 a recommendation scenario that contains a set of three users and four items. We can observe that, User 1 and the target user (User 3) share two items: Product 2 and Product 3, while User 2 and User 3 only share Product 2. Thus, User 1 is the most similar user to the target one. Accordingly, based on the collaborative filtering technique, User 3 gets recommended items that he has not rated before (Product 1 and Product 4) but that were already positively rated by the most similar user (User 1).

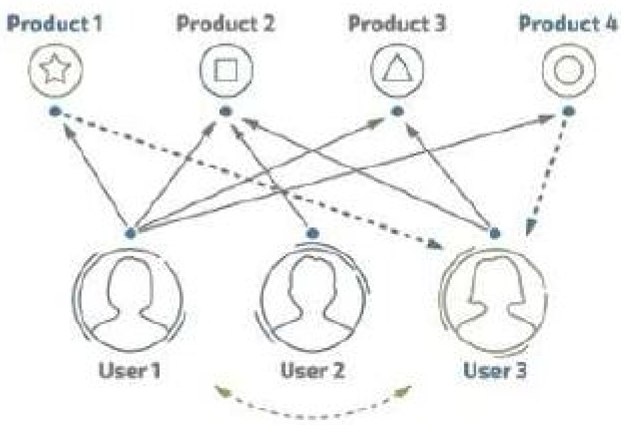


FIG. 1.3.: A collaborative filtering example.

In general, a collaborative filtering system requires the following steps to generate recommendations:

[6]https ://facebook.com/
[7]https://twitter.com
[8]https://www.google.com
[9]https://fr.linkedin.com

1. Identification of the subject of the recommendation (ratings of the target user).
2. Identification of the most similar users to the target one using a similarity function (cosine similarity, Pearson's correlation, etc.).
3. Identification of the rated items by the similar users and not rated by the target one.
4. Prediction of the rating of each selected item based on users' similarity.
5. Recommendation of items according to the predicted ratings.

The earliest CF recommender system is called Tapestry [18]. It was created to assist users filter the emails they receive by leveraging their colleagues appreciations. The authors have coined the term "Collaborative Filtering" because users collaborated by setting undesired emails. Tapestry paved the way for two principal recommendation algorithms in collaborative filtering: memory-based and model-based algorithms.

1.3.2.1 Memory-based algorithms:

The memory-based algorithm uses the entire user-item matrix to find similarities between users for estimating rating predictions. It is commonly referred to as neighborhood-based or heuristic-based algorithm. This approach uses previous users ratings for predicting ratings for new items using one of these two ways: user-based CF recommendation or item-based CF recommendation.

1. User-based collaborative filtering:
   In the user-based approach, recommendations rely on the similarities between an active user and the other users. Therefore, the first step is to determine the neighbors of the target user u using a similarity measure. The most similar users to the target one constitute the neighbors set, that we denote by *B(u)*. The ratings given by these neighbors to an item i are used to predict the rating $\hat{r}_{ui}$. The predicted rating can be obtained by a simple average of the neighbors ratings or by a weighted average considering the degree of similarity between users:

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in B(u)} sim(u,v).r_{vi}}{\sum\limits_{v \in B(u)} sim(u,v)} \tag{1.2}$$

   Where *sim(u, v)* represents the similarity measured between the users u and v.

The similarity measure is very important for selecting neighbors and for computing the rating prediction. Multiple measures can be used for this purpose. We present in the following some of the most commonly employed:

Pearson Correlation.

$$sim_{PC}(u,v) = \frac{\sum\limits_{x \in I_{uv}} (r_{ux} - \bar{r_u})(r_{vx} - \bar{r_v})}{\sqrt{\sum\limits_{x \in I_{uv}} (r_{ux} - \bar{r_u})^2} \sqrt{\sum\limits_{x \in I_{uv}} (r_{vx} - \bar{r_v})^2}} \tag{1.3}$$

Where $\bar{r_u}$ and $\bar{r_v}$ represent respectively the average ratings given by user u and v, and the items set I_{uv} is equal to $I_u \cap I_v$.

Jaccard similarity.

$$sim_{JS}(u,v) = \frac{|I_u \cap I_v|}{|I_u \cup I_v|} \tag{1.4}$$

Cosine similarity.

$$sim_{CS}(u,v) = \frac{r_u^\top . r_v^\top}{\|r_u^\top\| \|r_v^\top\|} \tag{1.5}$$

Where x . y represents the scalar product between the two vectors x and y, and $\|x\|$ represents the norm of x.

2. Item-based collaborative filtering:
Item-based CF makes recommendations by exploiting the neighbors of items. Thus, for predicting the rating $\hat{r}_{ui}$, the neighbors of the item *i* should be selected and the ratings given by the target user *u* to these neighbors should be considered. For this purpose, we compute the similarities between the item *i* and other items belonging to the items set. Then, we select the most similar items forming the neighbors set of *i* denoted by *B(i)*. Similarly to user-based CF, the predicted rating $\hat{r}_{ui}$ is computed as:

$$\hat{r}_{ui} = \frac{\sum\limits_{j \in B(i)} sim(i,j).r_{uj}}{\sum\limits_{j \in B(i)} sim(i,j)} \tag{1.6}$$

Where *sim(i, j)* represents the similarity measured between the items *i* and *j*. The similarity measures defined in Equations 1.3-1.5 can also be applied.

1.3.2.2 Model-based algorithms:

The model-based approach uses a collection of ratings in a learning phase, in which a model of user preferences is built to make intelligent rating predictions based on the observed data. Model-based CF algorithms are developed using data mining techniques and machine learning algorithms such as bayesian networks, clustering, neural networks, linear regression and latent factor models. These latter models are known as prevalent since they use latent variables in order to explain user preferences and perform a dimensionality reduction of the rating matrix for recommendation purposes.

1.3.3 Hybrid Approaches

We call a hybrid filtering recommendation system, a system that associates two or more recommendation techniques for better recommendation performance. As stated by Burke [7], a hybrid recommender system combines multiple techniques together to obtain some synergy between them. Most of the times, hybrid recommender systems have been proposed to overcome the weaknesses of collaborative filtering and content-based algorithms by combining them together instead of using them separately.

This trend had also been affected in competitions such as the Netflix Prize [10], where the winning candidate highlighted the fact that better results are often obtained when different recommendation algorithms are associated in a single model [19]. Hybrid approaches combining the collaborative and the content-based filtering methods can take various forms, a classification covering the principal hybridization combination strategies is presented by Adomavicius et al. [5] as follows :

- *Combining separate recommendations.* The predictions of individual recommendation algorithms are combined to obtain a single recommendation through different methods such as average weighting combinations [20].
- *Integrating content-based properties to collaborative filtering methods.* User-based methods can be adapted to calculate similarities depending on content-based user profiles [21].
- *Integrating collaborative properties to content-based methods.* CF models such as latent factor models can be adopted to a group of content-based profiles for text recommendation [22].

[10] http://www.netflixprize.com

- *Unifying content-based and collaborative properties.* Several approaches have been suggested within this scope. An example of unification of CBF and CF is the work described in [23], where authors propose to use characteristics of both methods in a single unified probabilistic method.

The combination of different recommendation techniques in a hybrid approach relies on the nature of the final application. However, some combinations of recommendation techniques may became very expensive to implement increasing the complexity of the recommender systems. Another drawback of the hybridization is lowering the speed of the recommender as more models are used at the same time.

1.3.4 Discussion

We've seen the three recommender system techniques as well as their main characteristics. However, there are still many open challenges and issues limiting the usefulness of real-world recommendations applications that need to be addressed. Even though hybrid recommendation techniques would overcome the problems of the combined techniques, there are certain limitations that are inherent to the recommendation problem and additional problems could arise when combining different methods. We outline the major ones in the following, knowing that we do not tackle all of them in this thesis. See Table 1.1 for a synthesis of the strengths and weaknesses of the different recommendation approaches.

1.4 Recommender Systems Evaluation

Evaluating a recommender system is a challenging task since it allows the RS to meet its fundamental objectives including, but not limited to, recommending pertinent items to users. To test the effectiveness of recommender systems and compare different recommendation approaches, it is important to define appropriate evaluation methodologies and metrics to measure the recommendation quality. We review in this section the process of evaluating a recommendation system.

TABLE 1.1: Synthesis of the advantages and limitations of RS

Approach	Advantages	Limitations
CF	1. Provides a useful recommendation when the overlap between users' feedback is high and the users' content is static. 2. A user may receive preferred items that he never searched for before. 3. A user is able to use information selected and evaluated by other users. 4. CF does not need a knowledge domain. 5. CF does not require items contents. 6. User's preferences can be predicted based on other users' interactions.	1. CF system is unable to recommend relevant new items to a user that has not provide enough ratings (user cold start problem). 2. A user with very specific interests makes the system unable to find good neighbors, and therefore relevant items (grey sheep problem). 3. An item that has not been rated by a considerable number of users, can not be recommended (new item cold start problem). 4. User-item ratings matrix can become sparse when the number of items or users highly increases. So, users can not rate all the items of the overall database. Thus, even the most common items have only few ratings (sparsity problem).
CBF	1. CBF system can deliver items recommendations based on their content information even if they have not been rated and interactions made by other users are not required. 2. A user's dynamic preferences can be controlled using profiles. 3. CBF is able to recommend new items. 4. It is possible to explain the results of recommendations by providing the set of content features that caused an item to appear in the recommendation list which increase the system's transparency (explainability).	1. Enough information about items and domain knowledge are often needed to be gathered to select appropriate items (content availability). 2. CBF approaches become unsuitable when only few users interactions are available to build user profile (user cold-start problem). 3. CBF approaches tend to recommend similar items to the ones already known by the user and cannot provide unexpected or novel suggestions (overspecialization).

Approach	Advantages	Limitations
Hybrid	1. Overcomes the weakness of both CF and CBF approaches such as sparsity and grey sheep.	1. Problem with the frequent changes of the user's content. 2. The hybrid approach can induce higher complexity and cost. 3. Needs both item and interaction data that are often not available.

1.4.1 Experimental Setting

In general, it is useful to follow some basic guidelines in the experimental studies:

- Hypothesis: before running the experiments, a concise and restrictive hypothesis could be posed. Therefore, a preliminary experiment is needed to test and validate this hypothesis.
- Controlling parameters: it is important to examine the sensitivity of some parameters before running the main experiments. In fact, to fairly compare different algorithms, some parameters must be tuned.

To evaluate a RS, the evaluation methodology defines the followed experimental protocol that can fall into one of the two main levels: the offline or the online evaluation.

1.4.1.1 Offline evaluation

Offline evaluations are popular methods performed in the literature to assess recommendation approaches. This kind of evaluation is realized by using collected datasets of items gathering users interactions. User behavior when interacting with the recommendation system is simulated by using the collected dataset. Since the majority of RS deal with the users behavior collected in the past, the offline evaluation does not need any interactions with real users allowing the comparison of wide range of approaches at low cost. However, offline evaluations cannot measure the effect of the recommendation system on the user behavior, they only give a first level performance evaluation by providing a good approximation of how the system would behave with real users.

The basic structure for offline evaluation process is based on the train-test and cross-validation techniques. The dataset containing the information of users, items and ratings is often partitioned. Part of this data is used to infer the optimal utility function and referred to as training set. The other part is known as the testing set and adopted to measure the recommendations performance. When the same data is used for both training and evaluation, the dataset splitting is useful for preventing algorithms from overfitting to the evaluation testing set. To split the dataset, different ways could be adopted, knowing that the chosen manner could depend on the domain of application and its constraints.

- **Random split.** The partitioning of the dataset is done randomly by choosing a certain percentage for each set. For example, 80% for training and 20% for the testing. The *k-fold cross-validation* method is performed by doing again this procedure k times and evaluating the results every time: each subset of the data is used as training set while the k - 1 remaining subsets form the test set. Then, the evaluation metrics, as presented in the following section are computed for each subset and finally averaged over the k runs.

- **Chronological split.** The separation of sets is based on the temporal information of interactions. It consists in considering a certain time threshold and selecting the recent user interactions for testing and the older information for the training set. This time-dependent splitting brings the problem of preventing knowledge of future preferences, since the testing set ratings are more recent than those in the training set.

After choosing the partitioning way, two tasks could be tested whether the items ranking or the items ratings prediction.

1.4.1.2 Online evaluation

Online evaluation is generally conducted with real users that interact with the system and give feedback based on their experiences. This type of evaluation focuses on measuring the change in user behavior during the interaction with the recommender system. Questionnaires or user studies could be provided to the user for evaluating the performance of the RS. The risk taken when carrying out online evaluation is requiring plenty of efforts in gathering the feedback responses from users. Moreover, comparing several algorithms through online experiments is expensive and time-consuming.

Besides choosing an evaluation methodology, evaluation metrics are also necessary to assess the performance of recommender systems. We describe those metrics in the subsequent section.

1.4.2 Evaluation Metrics

We now turn our attention to the different metrics adopted to assess the performance of recommender systems. A distinction needs to be made between the evaluation metrics by taking into account the goal of the system itself. Generally, these metrics can be categorized into *prediction accuracy metrics* that determine how well a system can predict the appropriate rating for an item and *top-N metrics* that measure the suitability of top-N recommendations to users. We present in the following the commonly used evaluation metrics:

1.4.2.1 Prediction accuracy metrics

Prediction accuracy is considered as the most discussed property in the recommendation literature. It measures how close the recommendation system rating predictions are to the users real ratings. To date, the majority of RS are based on a rating prediction phase, where the main assumption is that a RS that produces more accurate predicted ratings will be more preferred by the user. This category of evaluation metrics comprises the well known Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) which are considered as standard metrics for many RS such as the Netflix Prize [24]. The lower the error value, the better the predictive accuracy of the recommender system is.

- **Mean Absolute Error (MAE)** measures the average absolute deviation between the system's predicted ratings and the user's actual ratings. It is given by the following equation:

$$MAE = \frac{1}{N} \sum_{i \in N} |r_{ui} - \hat{r}_{ui}| \tag{1.7}$$

  Where:

  - N: the total number of recommended items.
  - $\hat{r}_{ui}$: the predicted rating of user u for item i.
  - r_{ui}: the real rating of user u for item i.

- **Root Mean Squared Error (RMSE)** measures the quadratic error and it is hence more sensitive to large errors, since the errors are squared before they are averaged. This means that the RMSE is useful when large errors are especially undesirable. The RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i \in N} (r_{ui} - \hat{r}_{ui})^2} \tag{1.8}$$

1.4.2.2 Top-N metrics

For evaluating the top-N recommendations, the used evaluation metrics focus on measuring the quality of top-N recommendation lists generated by RS. In this family of measures, we found two popular metrics borrowed from the field of information retrieval: *Precision* and *Recall*.

- **Precision@N** measures the fraction of relevant recommended items in the top-N position and is defined as follows:

$$Precision@N = \sum_{i=1}^{N} \frac{rel(i)}{N} \tag{1.9}$$

  Here, rel (i) indicates the relevance level of the item at position i, rel (i) = 1 if the item is relevant and rel (i) = 0 otherwise.

- **Recall@N** calculates the ratio of selected relevant items returned in the top-N position, to the total number of available relevant items Nr. Recall can be computed with the help of the following equation:

$$Recall@N = \sum_{i=1}^{N} \frac{rel(i)}{Nr} \tag{1.10}$$

  Increasing the recommendation list size may result in a higher recall but a lower precision, since a longer recommendation list tends to include relevant items. The *F-measure* evaluates the balance between these two metrics and is described as follows:

$$F - measure = \frac{2.Precision.Recall}{Precision + Recall} \tag{1.11}$$

  Besides evaluating the relevance of items in the recommendation list, it is also important to evaluate the ranking quality. In particular, we introduce the following two widely used ranking measures *Discounted Cumulative Gain (NDCG)* and the *Mean Reciprocal Rank (MRR)*.

- **NDCG@N** Normalized Discounted Cumulative Gain is calculated based on computing Discounted Cumulative Gain (DCG) which measures the effectiveness of a ranked list based on items relevance. NDCG is the normalized variant of DCG, where Ideal DCG (IDCG) is the best possible DCG.

$$DCG@N = \frac{1}{N}\sum_{i=1}^{N}\frac{2^{rel(i)}-1}{log_2(i+1)} \quad IDCG@N = \frac{1}{N}\sum_{i=1}^{k}\frac{1}{log_2(i+1)}$$

$$NDCG@N = \frac{DCG@N}{IDCG@N} \tag{1.12}$$

- **MRR@N** Mean Reciprocal Rank is described as the multiplicative inverse of the rank of the first relevant item, L represents the relevant items list in the testing set for each user, and $Rank_i$ denotes the position of the relevant item i in the recommendation list.

$$MRR@N = \frac{1}{|L|}\sum_{i=1}^{|L|}\frac{1}{rank_i} \tag{1.13}$$

1.4.2.3 Alternative performance metrics

While most research in recommender systems has focused on accuracy metrics, additional characteristics of recommendations could be taken into consideration. Thus, other performance metrics such as novelty and diversity may be measured [25]. Novelty and diversity are different though related notions.

- **Novelty** evaluates whether the recommended items are new to the user or not. It would be interesting if the user is recommended with novel items. Novelty can be measured by comparing the top-N recommendations against already used or rated recommendations. Given I_R, the set of items that have been previously recommended to a user u , and I_T, the set of the top-N recommended items to u, novelty can be defined as follows:

  $$Novelty_u = \frac{|I_T \setminus I_R|}{|I_T|}$$

  We compute this value for the user u and take the average $\frac{1}{N}\sum_{u=1}^{N} Novelty_u$ as the measurement of novelty, where N denotes the number of users.

- **Diversity** is related to how dissimilar the recommended items are with respect to each other. The diversity can be determined using the items content (e.g. movie or music genres) or the items ratings by measuring Intra-List Similarity (ILS) [26]. ILS calculates the similarity between two items i_n and i_m in the recommendation list L using a similarity metric such as Jaccard similarity coefficient [27]. For a user u, ILS can be computed as:

  $$ILS_u = \frac{1}{2}\sum_{i_n \in L}\sum_{i_m \in L} sim(i_n, i_m)$$

  From here, we can calculate the overall ILS as the average over all users.

1.5 Conclusion

In this chapter, we introduced the reader to an overview of recommender systems and explain how basically these systems work. Therefore, we presented the basic concepts, the recommendation problem formulation and the main recommendation techniques as well as their principal limitations. We also gave an overview on performance evaluation methodologies.
After this general introduction to RS, the next chapter attempts to extend existing knowledge and trace the evolution of the recommendation problem by considering recent emerging trends. Therefore, we will concentrate on context-awareness, situation-awareness and multi-criteria decision making in RS and we will discuss the main existing approaches in these areas.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we conduct a review of the work that relates to this thesis as a whole. Work specifically related to our proposals described in the next two chapters. Under these considerations, we focus on the type of the input information integrated into the recommendation process. Following this, we introduce a novel classification based on context awareness, situation awareness and multi-criteria decision making. This classification subdivides the recommender systems area into sub-areas. Various recommendation studies from the literature in each sub-area are described and analysed. Finally, we end this chapter with a brief synthesis.

2.2 Context Awareness

The use of contextual information is considered as a key component to boost the performance of systems that fall within numerous research disciplines, like mobile computing, information retrieval and recommender systems [28, 29]. In fact, the contextual information illustrated through different factors makes it possible to afford the most relevant information to the user when it is most needed. In what follows, we define the basic concepts of context and the notions that it entails.

2.2.1 Context Definition

By dint of the complexity and the wideness of the context concept, it has no a single definition. Indeed, context is a multifaceted concept that has been studied in various research fields and many gave multiple definitions, often different from the others and more specified than the general dictionary definition which describe context as: "conditions or circumstances that have an effect on something". Given the growing importance of context, an entire conference, CONTEXT[1], is devoted for presenting and discussing this topic in wide range of various disciplines including artificial intelligence, cognitive science, linguistics, philosophy, and psychology. Based on a general point of view, the majority of renowned dictionaries have defined the context by almost similar definitions. According to Oxford Advanced Learner's Dictionary[2], "a context is the situation in which something happens and that helps you to understand it". WordNet Search 3.1[3] considers a context as "the set of facts or circumstances that surround a situation or event". For Cambridge dictionary[4], the context is viewed as " the situation within which something exists or happens, and that can help explain it". Moreover, In Webster's dictionary[5] "a context is defined as the interrelated conditions in which something exists or occurs like environment and setting".

More specifically than the dictionaries definitions, many researchers presented several context definitions from different fields. The idea of including context in computer sciences was introduced in 1994 by Schilit [30], which defined the context as: *location and the identity of nearby people and objects*. In accordance with Schilit, context encompasses more than just user's location, because other things of interest are also mobile and changing. Context could also include lighting, noise level, communication bandwidth, network connectivity and even the social situation (e.g. whether you are with your manager or with a co-worker). Later, a more abstract definition [31] presented by Dey and Abowd in 1999 states that: *context is defined as any information that can be used to characterize the situation of entities (place, people, and things), including the user and application and the interaction between them.* This is probably the most commonly and widely used definition for context in the computational sciences.

2.2.2 Context Acquisition

As stated by [32], the context acquisition is the process through which contextual information is obtained. The context can be gathered in several ways, based on the contextual information that the system requires. Traditionally, there are three context acquisition methods [33]:

[1]http ://context-07.ruc.dk
[2]http://www.oxfordlearnersdictionaries.com/
[3]http://wordnetweb.princeton.edu/perl/webwn
[4]http://dictionary.cambridge.org/dictionary/
[5]https://www.merriam-webster.com/dictionary/

- Explicitly: for the explicit information capturing, the user is invited to intentionally specify his/her context. For instance, this can be done by asking questions to users through web forms. However, collecting these contexts is considered as an annoying task avoided by users due to privacy concerns and time-consuming effort.
- Implicitly: this type of information is captured from the surrounding environment of the user such as the location, weather time and social networks. These contexts can be obtained by physical sensors or software sensors. Generally, contextual information is registered in the internal memory of the mobile device and periodically transferred to the distant server for modeling.
- Inferring: this is the case when the context is inferred by means of different intelligent data analysis techniques from data mining, machine learning or deep learning (such as bayesian classifiers, neural networks, etc.). The success of this contextual information inference relies very significantly on the chosen learning techniques, and it also differs across various applications.

2.2.3 Context Modeling

Context modelling gives a formal representation of the collected contextual data. The most frequent modelling approaches are the following:

- Key-value models: are based on a set of vectors. A key can be represented by an id or a name of a contextual factor. However, the value can be a scalar or vectorial. These attributes can be weighted in accordance with their importance degree.
- Logic-based models: use an inference process to extract new facts from the existing rules. The contextual information is shaped by facts, expressions and rules.
- Ontology-based models: exploit domain ontologies or predefined concept hierarchies. The user context is shaped by classes, properties and relations.
- Graph-based models: can present the mutual relationship between contexts. This model is based on nodes (context) and paths (relations).

2.2.4 Paradigms for Incorporating Context

Determining how and when to integrate contexts is a crucial stage in the recommendation systems. In this respect, researchers in [5], categorize the context-based recommendation approaches into three categories as shown in the figure 2.1: contextual pre-filtering, contextual post-filtering and contextual modeling.

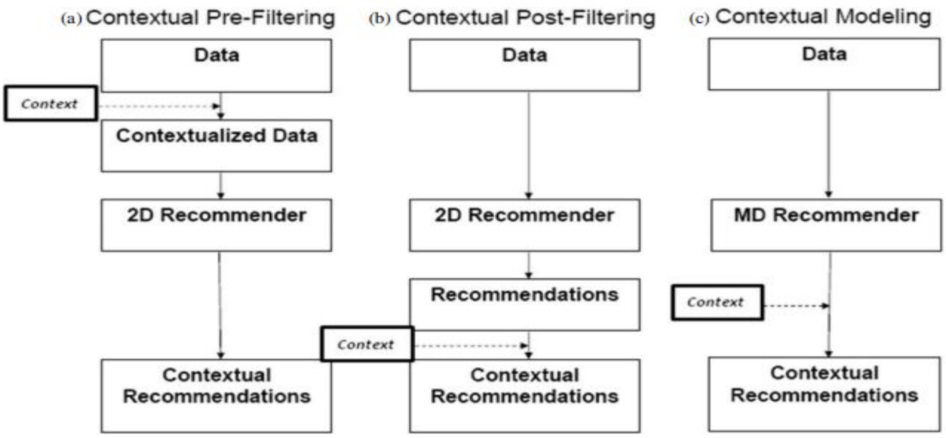


FIG. 2.1.: The difference between the three forms of context uses [2]

2.2.4.1 Contextual pre-filtering (or contextualization of recommendation input)

As stated in [5], this recommendation paradigm is based on the assumption that ratings are grouped depending on the specific contextual information before applying a recommendation algorithm. In fact, the context is essentially used as a query to filter out all the ratings that are not under that specific context. Therefore, ratings can be predicted by applying any traditional non-contextual recommendation approach on the filtered data. An example would be: when recommending movies to a user on weekends, employ only the movies that he has previously watched on weekends.

2.2.4.2 Contextual post-filtering (or contextualization of recommendation output)

In this recommendation paradigm, the predictions are performed using any traditional two-dimensional recommender system on the whole dataset while ignoring the contextual information. Afterwards, the resulting set of recommendations is contextualized for each user. This is done either by filtering out recommendations based on the contextual information or by adjusting the ranking of items in the recommendation list.

2.2.4.3 Contextual modeling

In the latter recommendation paradigm, contextual information are leveraged directly in the model at the same time when it is used to generate recommendations.

2.2.5 Context-Aware Recommender Systems Approaches

The recommendation field is one branch that adopted contextual information allowing recommender systems to be mightily contextualized to enhance the way in which these systems work. With the goal of understanding the state of the art of this field, we provide a thorough literature review which analyses relevant Context-Aware Recommender Systems (CARS) approaches along several application domains, context types, recommendation techniques and paradigms for incorporating context.
In our discussion, we will use the term *contextual dimension* referring to a contextual factor (e.g.,weather, time, etc.). The term *contextual condition* refers to a specific value of a contextual dimension (e.g., rainy, morning).
Among the earliest works on context-aware recommendation, the one proposed by Adomavicius et al. [16], who built a multidimensional recommentation model by integrating additional contextual dimensions besides the typical information on users and items. For rating prediction, this approach applied the collaborative filtering.
Since the early works on context-aware recommender systems, there have been many efforts made in this field where researchers have often tried to make use of contextual information to enhance standard recommendation algorithms. These recommendation approaches can generally be sub-divided by the formation of the utility function into memory-based and model-based approaches.

2.2.5.1 Memory-based approaches

In the literature, many attempts have been made in order to build context-aware recommendation systems by applying memory-based algorithms. Two primary types of memory-based algorithms have been introduced: the user-based, which finds neighbors according to users similarity; and the item-based, which finds neighbors depending on items similarity. Typical examples of these approaches are the neighborhood-based collaborative filtering approaches. In this respect, Lamche and co-workers [34], proposed and evaluated a context-aware recommender system in a mobile shopping scenario. It employed the nearest neighbor algorithm to recommend pertinent items according to the relevant selected contextual dimensions. For the task of Point-of-Interest (POI) recommendation, authors in [35] integrated the spatial, temporal and the social context in their recommendation model. They exploited various contextual dimensions in a collaborative filtering algorithm by varying their weights to investigate the effect of including each dimension on the recommendation accuracy. Otebolaku et al. [36] proposed an approach that emphasizes the importance of similarity between contextual dimensions. To predict user preferences, K-nearest neighbors (KNN) algorithm was adopted based on the similarity between user contexts and those of other users. In the work presented in [37], a prediction model focusing on selecting and weighting the contextual dimensions was proposed. More precisely, to select relevant contextual dimensions different selection methods were adopted based on variance, popularity and frequency. Then, these selected dimensions were weighted using the optimization algorithm particle swarm optimisation. The study proposed in [38] comprehended the researches of context-awareness, CF and hotel RS to build a personalized hotel recommendation system. This research considered also the hotel features and used neighborhood-based CF for rating prediction.

2.2.5.2 Model-based approaches

Several efforts followed the evolution of model-based approaches to adapt them for context-aware recommendation [39–45]. Therefore, many extended models of Matrix Factorization (MF) technique were proposed in the literature, like the contextual matrix factorization, also known as Context-Aware Matrix Factorization (CAMF). It was initially introduced in [39] to model the relatedness between the contexts and item ratings providing additional model parameters. Along with standard CAMF recommender systems, we investigate more recent CAMF researches. For instance, in [46] authors proposed a context-aware latent factor model realized using matrix factorization. This study integrated contextual information of both user and item in the absence of the historical user or item data to perform event recommendations. In [40], another CAMF was proposed for multi-context recommendation. In this model, users were involved in different contexts and all these users' contexts were considered in the process of rating prediction. Likewise, in [41] a factorization model was designed, in which multiple contextual dimensions were grouped into different context types to be used for recommendation. However, integrating inappropriate contextual dimensions degraded the recommendation performance. By contrast, authors put forward in [42], a context-aware recommender algorithm based on matrix factorization that emphasis the interactions of the contextual dimensions with users and items. In this approach, the information gain algorithm was used to select the relevant contextual dimensions. A closer work suggested in [43] proposed a context relevance identification method to elicit the useful contextual dimensions upon rating POI. This method is able to detect relevant contextual dimensions that do influence the users preferences and the decision-making process. Then, a matrix factorization-based prediction model was used to provide ratings prediction for items under various contexts. Another idea rarely suggested in the literature is introducing contextual correlation in CAMF approaches. Contextual correlation means useful inter-dependencies that may exist between contextual dimensions or contextual situations. In [44], authors introduced the contextual correlation into CAMF based on measuring the similarity between contextual situations. The underlying assumption behind the contextual correlation notion is that, more similar two contexts are, the two recommendation lists for a same user in those contexts should be similar too. In [45], authors also supported the notion of contextual correlation by providing a recommender system based on matrix factorization technique, where highly correlated contextual situations were clustered by using k-modes algorithm. In the proposed approach, different factors were taken into account including contextual information, user ratings and item features.

Besides matrix factorization based latent factor models, others model-based algorithms have been receiving attention counting on multidisciplinary techniques such as machine learning and deep learning. These techniques have revolutionized the data mining and information retrieval techniques offering an effective impact on context-aware recommendation. For example, Bozanta et al. [47] developed a hybrid context-aware recommendation system where ratings for new items were predicted by integrating three types of data: user-related, item-related and contextual information. Besides that, a weighted hybridization technique was applied to compute the items scores using available ratings prediction algorithms and the artificial neural network was used to determine the optimal weights for each algorithm. Therefore, final ratings were computed by multiplying the neural network weights and the ratings from different algorithms. A classifcation approach was used in [48] where Meng et al. were interested particularly in developing a bayesian context-aware representation model for grocery recommendation. The user and item latent vectors were learnt by leveraging basket context information from previous user-item interactions. The representation of users and items were jointly modeled in a bayesian manner, which represents users and items as gaussian distributions. Having deduced the users and items representation vectors, the items preferences scores for each user that estimate user purchase probabilities can be calculated based on these latent representations. Sarker et al. [49] proposed a prediction model based on Naive Bayes classifer to handle noisy instances in mobile phone data for providing context-aware mobile services. They also employed the rule-based classification technique decision tree to build contextual prediction rules for the purpose of generating the prediction model. In this model, authors assumed that the temporal, spatial, and social contexts, are the relevant contexts to their problem domain. In [50], authors built context-aware local

recommendation models where users were clustered, regarding visited destinations each period of the year. Here, the k-means clustering technique is applied to generate k clusters of countries where residents have similar behaviors according to their country of residence and to the visited destinations in different periods of visits. In reference [51], a context-aware smartphone application was developed based on artificial intelligence mechanisms to reduce the large dimensionality of context data. The principal component analysis was considered for dimensionality reduction and decision tree for building the prediction model.
During the last few years, machine learning-based regression models such as fuzzy logic methodologies has revolutionized several information storage and retrieval disciplines like recommender systems. Among the earliest contributions on fuzzy based recommendations, the one presented in [52], introducing fuzzy logic methods for constructing recommender systems. The proposed methods regarding reclusive modeling differ from collaborative filtering recommendation, they deal with single individual user preferences, object representation, item profiling and domain expert prototypes. Similarly, closer context-aware fuzzy logic recommender approaches were introduced in [53, 54].
Besides the presented machine learning approaches, a surge of interest in applying deep learning to recommendation systems has emerged. In fact, deep learning techniques are making major advances in problems that machine learning did not have good results. However, the existing work in this domain is still quite limited, and furthermore, it does not utilize contextual information, which is largely present in the real-world scenarios. Authors in [55] have proposed deep context-aware modeling approaches that suggest integrating different representations of context in the recommendation process. These models utilized explicit and latent context representations derived from various contextual dimensions and learned nonlinear relations between latent features of items, users, and contextual information. Kim et al. [56] developed a context-aware hybrid recommendation model that combines MF based collaborative filtering method with a deep learning technique known as convolutional neural network for document recommendation. Consequently, it can capture the contextual information of item description documents for the rating prediction task. However, this model only models the document context from an item view. In [57], researchers proposed to integrate the information from deep learning and topic modeling to extract more global context information and make a better understanding of user reviews. For context aware recommendation, this model combined both ratings and review information into a unified model.

2.2.6 Discussion

We present in Table 2.1 the major existing CARS approaches to compare them depending on different criteria:

- **Context Acquisition**: this criterion specifies how to acquire the contextual information;
- **Context Relevancy**: this criterion examines whether, in the given approach, researchers take into account contextual dimensions/conditions relevancy or not. Context relevancy could be depicted by a context weighting method or a context selection method or both methods;
- **Context Correlation**: this criterion verifies whether, in the given approach, researchers consider the correlation between contextual dimensions/situations or not;
- **Recommendation Approach**: this criterion shows the category of the approach used to recommend items according to the integrated context.
- **Recommendation Strategy**: this criterion specifies the technique adopted for recommendation;
- **Recommendation Paradigm**: this criterion indicates how the contextual information are integrated in the recommendation process.

As observed from Table 2.1, we note that the existing CARS follow the common classification of traditional RS: collaborative filtering, content-based filtering and hybrid recommendation approaches. That means that these works did not invent a new specific classification for CARS. In these approaches, the context is often integrated directly into the recommendation model when it is used for producing recommendations. In fact, a better recommendation accuracy could be delivered by taking advantage of using contextual modeling paradigm in generating recommendations. Whereas, the contextual pre-filtering is considered as a simple method working well with a large amount of data [33]. However, it does not adapt well to many contextual dimensions and could give rise to the sparseness problem. While the contextual post-filtering paradigm takes interactions into account, it could also increase the data sparseness and the computational cost. Nevertheless, according to conducted evaluations studies [63], there is no clear out-performance between the contextual pre-filtering, contextual post-filtering and contextual modeling approaches. The best method depends on the used recommendation model and application domain.

TABLE 2.1: Synthetic overview of surveyed works on context-aware recommendation

Ref.	**Context Acquisition**	**Context Relevancy**		**Context Correlation**	**Approach**	**Strategy**	**Paradigm**
		Weighting	**Selection**				
[40, 58]	Not treated	No	No	No	CF	Model-based:MF	Contextual modeling
[59]	Implicitly	No	No	No	CF	Model-based:MF	Contextual modeling
[57]	Inference	No	No	No	Hybrid	Model-based:MF, clustering	Contextual modeling
[41]	Implicitly	No	Yes	No	CF	Model-based:MF	Contextual modeling
[43]	Implicitly and explicitly	No	Yes	No	CF	Model-based:MF	Contextual modeling
[45]	Explicitly	No	No	Yes	CF	Model-based:MF, clustering	Contextual modeling
[60]	Inference	No	No	Yes	CF	Memory-based:kNN	Contextual modeling
[44]	Implicitly and explicitly	No	No	Yes	CF	Model-based:MF	Contextual modeling
[61]	Implicitly	Yes	No	Yes	CF	Model-based:MF	Pre-filtering
[38]	Explicitly	No	No	No	CF	Memory-based: neighborhood/ user-based, item-based	Pre-filtering
[34]	Explicitly	Yes	Yes	No	Hybrid	Memory-based: neighborhood/ user-based	Pre- and post-filtering
[35]	Inference	Yes	No	No	CF	Memory-based: neighborhood / user-based	Contextual modeling
[37]	Explicitly	Yes	Yes	No	CF	Memory-based: neighborhood / user-based	Contextual modeling
[62]	Implicitly	Yes	No	Yes	CF	Memory-based: neighborhood / user-based	Pre-filtering
[36]	Implicitly and explicitly	Yes	No	Yes	CF	Memory-based: neighborhood / user-based	Contextual modeling
[47]	Explicitly	No	Yes	Yes	Hybrid	Model-based:artificial neural network	Contextual modeling
[48]	Explicitly	No	No	No	CF	Model-based:bayesian classifier	Pre-filtering
[42]	Explicitly	No	Yes	No	CF	Model-based:MF	Contextual modeling

Ref.	Context Acquisition	Context Relevancy		Context Correlation	Approach	Strategy	Paradigm
		Weighting	**Selection**				
[49]	Implicitly	Yes	Yes	No	CBF	Model-based:decision tree, Bayesian classifier	Contextual modeling
[51]	Implicitly	No	No	Yes	—	Model-based:decision tree	Contextual modeling
[55]	Implicitly and Explicitly	Yes	Yes	No	CF	Model-based:neural networks	Contextual modeling
[56]	Inference	Yes	Yes	No	Hybrid approach	Model-based:MF, neural networks	Contextual modeling

We can also see that many existing CARS especially in mobile or POI recommendation domains use implicit context acquisition [41, 49, 51, 59, 61, 62]. This could be explained by the interesting technological development of the systems or mobile devices that gather different contextual information without requiring an action from the user. For instance, time, location and social networks information could be obtained by intelligent devices which integrate several sensors, positioning and recording systems. In different recommendation domains such as e-commerce, there are several studies [34, 37, 38, 42, 45, 47, 48] that exploit the explicit way for context acquisition which could offer more reliable source of information. Despite this, some researchers would rather prefer the implicit context extraction manner since in the explicit way many users may not want to provide their information [64]. Few researchers [36, 43, 44, 55] combined both explicit and implicit context extraction approaches aiming to make the system more flexible. Context can also be inferred using data mining or machine learning methods [35, 56, 57, 60] by monitoring users activities with the system.

Another important aspect of the literature is the widespread interest in using collaborative filtering approaches, which play a principal role in the success of several CARS. These recommendation systems only depend on the user past behavior. Contrary to content-based approaches which require additional information about items. In CF approaches, the most widely used algorithms are the model-based considering users ratings to build a learning model. As shown in Table 2.1, matrix factorization methods are the most employed in the model-based approaches. In the presented approaches, several variations and extensions of MF methods have been used [41, 43, 45, 57, 59, 65]. Model-based algorithms were developed using different machine learning techniques where a recommendation approach can be viewed as a classification problem to identify what might interest the user and what might not. Various algorithms are used for this task, such as decision trees [49, 51], clustering [45, 57], neural networks [47, 55, 56] and bayesian classifiers [48, 49].

Despite the popularity of the research around CARS, some of the existing studies still mainly rely on incomplete assumptions about how to work with contextual information. Many CARS [38, 40, 48, 57–59] assumed that all existing contextual dimensions have equal effects and should contribute to make recommendations.

Differently, other approaches has been performed to underline contextual dimensions relevancy. We can classify these studies into two main fields. The first one [41–43, 66, 67] focus on selecting the most prominent contextual dimensions or conditions. For the second field [35, 36, 61, 62], the context relevancy is viewed as assigning a weight of importance for each contextual dimension/condition.

These works have examined context selection and context weighting separately. While other works [34, 37] merged these two fields into a unified recommendation model. Various methods have been explored by the previous presented researches to weight and select the most relevant contextual dimensions. For instance, optimization methods were adopted in [37] to obtain the power of influence of each contextual dimension. We can also find statistical methods [66, 67] used for selecting the most useful contextual dimensions. Another contextual dimensions selection methods based on measuring the deviation of the user's predicted ratings were found in [41, 43]. The information gain algorithm was also used in [42] to determine the contextual dimensions that affect the user's interests. Some studies [49, 55] mainly focused on the approach's research area and assumed that common contextual dimensions could be selected as relevant in compliance with their application domain.

To present the contextual effects on recommendations, the mentioned approaches focused on identifying the most important individual contextual dimensions ignoring the challenge of discovering significant correlations between them. When treating individual contextual dimensions, some valuable information may be lost since the recommender system is prevented from taking advantage of useful relationships that might exist between contextual dimensions.

To recognize contextual correlations, some approaches [36, 44, 45, 61] relied on measuring the similarity between contexts. Others adopted different methods such as applying learning algorithms [60] or probabilistic methods [68].

We note that both contextual dimensions relevancy and correlation seem to be two closely linked topics and are of paramount importance in enhancing context-aware recommendations. Although plenty of solutions have been proposed for the problems in the area of context-aware recommendation, the majority of them represents distinct methods for discovering relevant or correlated contextual dimensions. The lack of methods that deal with both contextual information relevancy and correlation is a quite challenging process. We believe that it's essential to combine these two topics to be handled by one method for mitigating the computation complexity and the dimensionality of context representation.

2.3 Situation Awareness

We start with some fundamental descriptions of contextual situation. Then, earlier recommendation approaches based on situation awareness are analyzed.

2.3.1 Contextual Situation Definition

The context concept is viewed as the data provided by the surrounding environment related to several contextual dimensions like locations, time, etc. The use of these contextual dimensions is very crucial to raise the performance of recommendation systems. However, they only form a low-level context acquired from the environment that needs to be interpreted into a high-level abstraction of the context determining a contextual situation. Being on a higher-level, fewer contextual situations than contextual dimensions may exist and relationships between contextual situations are less complex than relationships between the contextual dimensions they consist of.
In the computer world and specifically in recommendation systems, the concept of situation is multifaceted and the term situation can be used with different meanings. In the recommendation systems literature, there seems to be no general definition of a situation. Thus, defining a situation is a challenging task particularly related to the extraction of human knowledge and interpretation. For instance, in [69] user preferences on multiple item criteria are viewed as contextual situations in which user will make a final decision. However, multiple works [70–74] explored the relationship between context and situation by associating the situation notion with context and its instantiated dimensions. As expressed by [71] situations are viewed as *logically aggregated pieces of context*. In this definition, authors mention that to recognize a situation multiple pieces of contextual dimensions have to be combined and that a specific situation is derived from these dimensions. For instance, being `"in a meeting"` constitutes a situation as well as someone `"traveling in the morning by a car from home to office"`. In [72], the situation is described as *a stable interpretation (snapshot) of the context of an object at a specific time t*. This definition considers that the interpretation is essential to transform pieces of context into a situation. This interpretation process is based on the fusion of different contextual dimensions during a certain period of time. Similarly, Lin et al. [73] agreed with [72] and stated that : *a situation is the description of contexts that are logically combined during a certain period of time*. In this case, the situation `"in a meeting"` could be derived from the fact that `"it is 9:00 a.m. and a meeting is scheduled in the user's calendar for 9:00 a.m"`. According to [70], the situation can also be viewed as *the set of all known context information*. Here, all the contextual dimensions are considered without selecting the relevant ones. Recently, authors in [74] reported

that *users' current situation could autonomously be identified by blending together various kinds of contextual information*.

According to the above presented content, we use the following definition for a contextual situation: *A combination of potentially many contextual conditions*.

Besides context awareness that focuses on the system's state of being aware of single pieces of contextual information presenting the user's environment. We go one step further to determine the user's current situation by blending a multitude of contextual dimensions. Thus, situation awareness represents the system's state of being able to actually comprehend the user's current contextual situation. It is considered as the main precursor to decision making by providing paramount information for making satisfying decisions during a task. Lin et al. [73] described situation awareness as *the ability of using situations to provide domain specific abilities to users*. In short, situation awareness is about knowing what is going on around the decision maker and richer situation awareness is more likely to lead to good decisions and then to good performance [75, 76].

To recognize contextual situations, various techniques can be found. They may be either manually methods, or logically based ones, or those focusing on classifying human activities considering contextual description.

2.3.2 From Context-Aware to Situation-Aware Recommender Systems

A large number of approaches exist to recognize a contextual situation from the observed contextual information about a given user, e.g. probabilistic graphical approaches, fuzzy logic approaches, knowledge-based approaches, etc.

Probabilistic graphical approaches

Bayesian network is a type of probabilistic graphical models that can be used to generate predictive models, thanks to its ability to deal with the uncertainty inherent in every facet of human life. In this respect, authors in [77], suggested a context-aware music recommender system, where various contextual dimensions were gathered together to infer user's contextual situation by adopting Bayesian Networks (BN). Then, music recommendations were provided according to users preferences by situation. In [78], a system is introduced to generate recommendations to mobile phone users. It acquired user's context through mobile phone sensors. Then, user's contextual situation was predicted by employing dynamic bayesian networks which is the extended version of BN.

Fuzzy logic approaches

Fuzzy logic has been extensively used in many areas to handle the uncertainty and vagueness in user's behaviour. For instance, in [79], Thyagaraju et al. proposed an intelligent service recommendation model based on fuzzy logic and rule based reasoning. In fact, fuzzy linguistic variables were applied to define contextual situations from multiple contextual dimensions and the rules for adopting the policies of implementing a service recommendation following users' situations. Sen et al. [80] put forward a music recommendation system exploiting contextual signals gathered from phone sensors. The contextual dimensions were passed through a fuzzy logic model to predict user's contextual situation, which was then considered to suggest music from an online music streaming service.

knowledge-based approaches

To identify situation awareness in recommender systems, knowledge-based solutions are often adopted using domain-specific rules operating on ontologies.
Particularly, in [81], Hermoso et al. developed a software architecture for a proactive situation-aware mobile recommender system. The authors suggested to integrate different reasoning approaches. In fact, complex event processing was used to identify situation awareness by taking the current contexts of all users into account and ontologies were applied for defining structural domain knowledge and semantic rules for specifying personalized recommendations. Jung et al. [82] suggested a situation-aware framework for hospital recommendation services. To realize the situation-awareness supporting framework, a domain-specific knowledge model was adopted for knowledge extraction from online health communities and text-mining techniques were used for aggregating raw data to high-level information analysis. While these approaches neglect the recommendation techniques such as content-based and collaborative filtering. Other knowledge-based approaches combine both worlds. In this respect, a software architecture was proposed in [83] to offer recommendations adapted to the user's interests and current contextual situation. The mobile devices interpreted low-level data obtained from their embedded sensors and provided the infrastructure to infer high-level contextual situations. Then, the recommender system computed situation-aware recommendations by combining knowledge-based with content-based and collaborative filtering recommendation techniques.

2.3.3 Discussion

Another way to integrate the context in the recommendation process is through incorporating contextual situations. Few works [36, 44, 61] have looked into producing ratings prediction or recommendations depending on a specific contextual situation. These studies viewed a contextual situation as a set of various contextual conditions without checking whether these conditions are suitable to be adapted together.
Furthermore, other researches such as [77–80] tended to focus on estimating the current contextual situation in which the user is involved depending on his actual context. For this purpose, these studies aim at gathering the different contextual dimensions and conditions to infer a current situation by employing various inference models like fuzzy logic, probabilistic graphical and knowledge-based models.

2.4 Multi-criteria Decision Making

2.4.1 Multi-criteria Decision Making & Recommender Systems

Before discussing the recommendation problem under the prism of Multi-Criteria Decision Making (MCDM), we should examine the relationship between the recommendation process and the MCDM process. From the MCDM perspective, generic recommender systems can be appeared as a simplified MCDM problem. In fact, RS are considered as tools that support users making decisions by selecting a choice from a multitude of alternatives.
To address the recommendation problem from the MCDM perspective, these two fields have gradually evolved into an overlap in the field of multi-criteria recommendation.
In the following, we present the steps of decision making methodology process proposed by one of the pioneers in MCDM methods [84] to understand how the MCDM can be adopted when building a recommender system:

1. *Defining the object of decision*. In other words, defining the set of items on which the decision must be taken as well as the rationale of the recommendation decision.

2. *Defining the family of criteria*. The criteria refer to the multiple item features that can express the preferences of the decision maker (targeted user) over various items influencing the recommendation decision.

3. *Developing the global preference model*. The model aggregates the user preferences on the different criteria into an overall preference about an item.

4. *Selecting the decision support process.* In this step, a final decision is made by designing and developing the appropriate procedure, methods, or software systems that will assist a decision maker in choosing the best item according to the results of the previous steps.

In multi-criteria recommender systems, the object of decision (i.e., corresponding to step (1)) refers to an item included in the set of all the candidate items. According to [85], there are four decision problematics. Some research work has focused on the decision problematic of items *ranking* (i.e., ranking items from the most to the least in accordance with their relevance). Other studies support the *sorting* of items into several categories pursuant to their pertinence for the user (i.e., recommended vs. non-recommended items). There are systems that support the *choice* (i.e., selecting the most convenient item for a given user) and *description* (i.e., presenting the appropriateness of a particular item to a user). In multi-criteria recommender systems, the criteria (i.e., corresponding to step (2)) are considered as items features. In MCDM, four types of criteria are generally used [84], and they are measurable, ordinal, probabilistic and fuzzy. A global preference model (i.e., corresponding to step (3)) affords a way of aggregating the partial preferences upon each criterion to present the total preference of a user regarding an item. In the MCDM literature, various categories of global preference modeling have been developed [85]: value-focused models, outranking relations models, multi-objective optimization models and preference disaggregation models. Finally, referring to the decision support process (i.e., corresponding to step (4)), a final decision for a given MCDM problem is made using recommendation systems fitted directly into the MCDM category to support a decision maker when taking a decision about an items set.

2.4.2 Multi-criteria Recommender Systems Approaches

Nearly the vast majority of the existing recommendation systems allow users to evaluate items using a single rating value that expresses user's opinion about an item as a whole. Traditional recommender system usually defines a single-criterion based utility function $f_R(u, i)$ measuring the appropriateness of recommending an item $i \in Items$ to a user $u \in Users$ as follows:

$$f_R : Users \times Items \rightarrow R_0 \tag{2.1}$$

This utility function considers two types of entities to generate single criterion ratings (R_0) representing the overall impression of an item in the two-dimensional $Users \times Items$ space. Nevertheless, this kind of single-criterion recommendation are not sufficient to meet users' personalized needs. When multiple criteria are being adopted in items evaluation, their ratings lead to more accurate recommendation than single criterion based ones and therefore, multi-criteria ratings are considered [86].

Multi-Criteria Recommender Systems (MCRS) take advantage of providing more information about users tastes from different aspects than traditional recommenders. Besides the overall rating, MCRS focus on describing users opinions through multiple criteria and considering their feedback on them. Therefore, MCRS have an important impact on the accuracy of the prediction of consumer preferences in many applications.

Example (Multi-criteria hotel recommendation). Considering a booking platform through which tourists can evaluate hotels differently from several aspects, such as the quality of the food, cleanliness and value for money. Some customers would be willing to sacrifice the cleanliness for great food quality. While, others may desire the value for money to food quality. Considering ratings on each of these criteria instead of a single rating over the hotel can assist providing appropriate recommendations, which make the process more customized. Accordingly, the utility function $f_R(u, i)$ of multi-criteria recommender systems is no longer with an only single overall rating (R_0). It additionally takes under consideration user's ratings on item criteria (R_1, R_2, ...,R_k):

$$f_R : Users \times Items \rightarrow R_0 \times R_1 \times R_2 \times ... \times R_k \tag{2.2}$$

Following the formation of the utility function, MCRS approaches applied in ratings prediction can be grouped into two categories [5]: memory-based and model-based.

2.4.2.1 Memory-based approaches

Like its implication in traditional recommendation approaches, memory-based techniques depend on the user's observed data and certain heuristic assumptions to compute the utility of an item for a user. One possible way of reflecting multi-criteria ratings in memory-based approaches is to adopt similarity computation. For example, among memory-based recommendation approaches, the ones based on nearest-neighbors methods assuming that similar like-minded users show similar patterns of rating behavior and similar items get similar ratings. There has been some multi-criteria recommenders extended from the traditional memory-based recommendation approaches through leveraging multi-criteria ratings within the similarity computation. Among these approaches, the earliest contribution on multi-criteria recommendation systems introduced in [86]. In this approach the multi-criteria ratings were predicted by applying traditional memory-based recommendation techniques. To obtain the overall rating, the learned aggregation function was used based on the predicted criteria ratings. In [87], authors presented a memory-based recommendation approach based on identifying the neighbors of an active user and a target item by ranking the criteria preferences of each user and item. The neighbors were further exploited to predict the overall rating of an item by adapting traditional CF methods through the modification of the aggregation functions they used. Wasid and Ali [88] proposed a clustering approach to integrate multi-criteria rating into traditional CF recommender system by using k-means algorithm. To achieve that, initially users with similar criteria preferences were clustered. Then, Mahalanobis distance was used to compute the most close neighbors for a user within the same user's cluster. After that, the predicted rating of an item was computed based on the neighbors similarities. Koudaria et. al. [89] proposed a hybrid ranking system to obtain the top-N list for MCRS. To reflect the interest of the user for each criterion, partial-ranked lists were found for each item using a learning-to-rank method. Next, global ranking list was computed by aggregating obtained partial ranked lists using ranking aggregation method that represents the user preference for each criterion.

Furthermore, in memory-based approaches, studies on weighted similarities aggregation and criteria ratings have been performed. Since aggregating criteria ratings is one of the important concern in MCRS, Gupta and Kant [90] used genetic programming to aggregate the users similarities and find the overall ratings by determining weights for each criterion. These weights were then integrated in CF process to provide recommendations.

2.4.2.2 Model-based approaches

In contradiction with memory-based approaches, model-based approaches learn a predictive model to predict the utility of items for the user. There are several existing multi-criteria rating recommenders that fall into this category.

In this regard, authors in [91] developed a fuzzy bayesian multi-criteria approach that deals with the uncertainty associated with user preferences and correlation based similarity problems. To generate recommendations, the most favorite criterion was predicted for each user on the basis of previously rated items and then recommendations were provided based on the predicted preferred criteria. Zheng [65] created an utility-based multi-criteria recommendation algorithm. The main idea of this approach is to recommend items to a user based on the utility function built using the multi-criteria ratings. More precisely, the utility is defined as the similarity between the vector of user expectations and the vector of user evaluations in terms of the predicted multi-criteria ratings. The research in [92] proposed to integrate the multi-criteria ratings of travellers extracted from social media networking for building an hotel recommender system. For this task, clustering and prediction machine learning techniques were used with the aid of fuzzy logic for finding the similarities between the travellers based on their ratings. Batmaz et al. [93] propounded a CF based multi-criteria RS presenting an aggregation function using autoencoders and neural networks. Criteria ratings prediction were produced based on the relations among users preferences extracted by the autoencoder. In the second part, the aggregation function for each user was learned by neural network. At last, overall ratings prediction were generated using criteria predicted rating and aggregation function.

2.4.3 Discussion

After the extensive review of the literature describing various researches on multi-criteria recommendation systems, we analyze and classify the discussed approaches (Table 2.2) according to different features:

- **Decision problematic**: this feature specifies which problematic that the system aims to support (Section 2.4.1). It can take four values: Choice, Sorting, Ranking and Description.
- **Family of criteria**: this feature identifies the items criteria type (Measurable, Ordinal, Probabilistic and Fuzzy);
- **Global preference model**: this feature indicates the way of aggregating criteria partial preferences into the overall preference of the decision maker regarding an item. There are four categories of preference modeling approaches: value-focused models, outranking relations models, multi-objective optimization models and preference disaggregation models;
- **Recommendation Approach**: this feature defines the category of the presented multi-criteria recommendation approach;
- **Recommendation Strategy**: this feature shows the technique used for recommendation;

TABLE 2.2: Synthetic overview of surveyed works on multi-criteria recommendation

Ref.	Decision problematic	Family of criteria	Global preference model	Approach	Strategy
[93]	Choice	Measurable	Value-Focused	CF	Model-based: autoencoders, neural networks
[94]	Choice	Measurable	Multi-Objective Optimization	CF	Model-based: neural networks
[92]	Ranking	Fuzzy	Value-Focused	CF	Model-based: clustering, regression, classification, fuzzy
[65]	Ranking	Measurable	Multi-Objective Optimization	CF	Model-based: matrix factorization
[91]	Choice	Fuzzy	Preference Disaggregation	CF	Model-based: bayesian classifier, fuzzy
[90]	Ranking	Measurable	Value-Focused	CF	Memory-based: neighbors-based (user-based)
[89]	Ranking	Measurable	Value-Focused	Hybrid	Memory-based: listwise CF
[88]	Choice	Measurable	Preference Disaggregation	CF	Memory-based: neighbors-based (user-based)
[87]	Choice	Measurable	Value-Focused	CF	Memory-based: neighbors-based (user-based, item-based)

The classification of the studies on multi-criteria recommender systems in Table 2.2 revealed some important observations. In fact, many clues about the actual findings and the further advances of MCRS approaches are given. Concerning the decision problematic, Table 2.2 shows that the primary task that multi-criteria recommender systems [87, 88, 91, 93, 94] aim to support is "choosing the best item from a set of candidates" which refers to the "Choice" problematic. This has been somewhat expected, as the issue that led to recommender systems was an overload issue, and most RS focus on finding some good items by screening out lots of bad ones [95]. There are also many MCRS approaches [65, 89, 90, 92] that support the "Ranking" problematic, which presents a ranked list of proposed items from the best one to the worst one. This task recurs in many recommender systems especially the commercial ones. Typical examples are the top-N recommendation algorithms [96].

For the criteria type, in commonly investigated studies [65, 87–90, 93, 94] the employed criteria are mainly "Measurable", where users rate items on a measurable evaluation scale for each criterion. Whereas, few others [91, 92] engage "Fuzzy" criteria.

Furthermore, as Table 2.2 reports, a small minority of multi-criteria recommenders engage in building the global preference model preference disaggregation methods

based on past decisions [88, 91] or multi-objective optimization methods [65, 94]. Contrariwise, the vast majority of the methods are based on "Value-focused models" [87, 89, 90, 92, 93]. These methods are simple to implement by calculating overall user preferences in the form of an additive utility function.
It is also apparent from Table 2.2 that nearly all surveyed MCRS adopt the collaborative filtering approach as traditional RS, but taking into consideration multi-criteria ratings. The techniques that use the majority of these systems [65, 91–94] to predict multi-criteria ratings or item overall rating (or both) fall within model-based category including bayesian classifiers and various machine learning techniques. While some other systems [87, 88, 90] rather use memory-based approaches to apply neighborhood-based collaborative filtering recommendation techniques.

2.5 Jointly Leveraging Context-based and Multi-criteria Aspects

Only very few studies have focused on combining both context information and multi-criteria ratings within a single recommender [97, 98]. The main idea in [97] is extending the dimensionality of the recommendation space to provide personalized recommendations, where the contextual information and the multi-criteria ratings were considered besides the users and items. Closest user's neighbors were found using the multilinear singular value decomposition (MSVD) technique under pertinent contextual information to be integrated in the recommendation process. Recently, Zheng et al. [98] incorporated contextual information into MCRS baselines. For multi-criteria rating predictions step, they used independent and dependent methods, and for the rating aggregations step they employed linear and conditional aggregation methods.

2.6 Synthesis

Traditionally, RS inputs comprise the users opinions or the interaction histories with the systems to provide suggestions that satisfy users requirements.
In the light of recent research efforts, there is now considerable concern about developing more complex recommendation approaches. Therefore, it has been suggested to explore extensions of recommender systems from the aspect of their input data type. With this in mind, an important leap can be taken by integrating richer information in the recommendation process, be it in terms of item criteria

feedback or users' contexts [58, 86]. Accordingly, classical recommendation approaches could be extended through two directions. The first one is dedicated to context-aware recommender systems [58], where useful contextual dimensions affecting users interests are integrated into the recommendation. The second direction is devoted to multi-criteria recommender systems [86, 99], which consider item description through multiple criteria and take into account the users feedback on each of them.

It has been shown that it is advantageous to incorporate properly users contextual information in the recommendation process to enhance recommendation accuracy [58]. Furthermore, it has been proven that it is also useful to additionally incorporate multi-criteria feedback to provide better recommendation results [100]. Multiple criteria and context-aware directions have a better ability in recommending relevant items compared to traditional approaches. In extant literature, there is much research on both context-aware and multi-criteria recommender systems but separately. As far as we know, only very limited work [97, 98] tackled the problem of combining both two directions within one recommender system. In [97], a multi-dimensional recommendation space is defined to carried out personalized services in mobile commerce depending on users neighbors' feedback. To find users' neighbors, the MSVD is adopted based on users multi-criteria ratings under relevant context information. More recently, in the contribution proposed by Zheng [98], context-awareness and multi-criteria decision making are both addressed in the area of educational learning. This approach points towards the idea of incorporating contextual dimensions into different previous multi-criteria baselines.

2.7 Conclusion

This chapter provides a review of several recommender systems to investigate the current state of the art of these systems. Accordingly, we expose several works dealing with the various types of context, ranging from the user's contextual dimensions to the user's current contextual situation. Along with that, we present multi-criteria recommender systems which were developed to deliver better performance in several recommendation scenarios. Thereafter, we focus on the few studies that deal with the multi-dimensional available data by associating both contextual and multi-criteria aspects in a unified recommendation model.

In the following chapter, we will concentrate on context awareness and situation awareness in RS and extend existing knowledge in the recommendation field to put forward effective approaches exploiting the contextual information.

Chapter 3

A Context-based Recommendation Approach

3.1 Introduction

There are various recommendation systems that attempted to meet the challenge of suggesting the suitable information according to users context information. Under this consideration, we studied existing CARS in the previous chapter. However, crucial questions still arise here: How to identify the contextual information on which users interests depend ? How to use such information for providing accurate predicted items ratings ? This chapter attempts to answer these questions by introducing the reader to our proposed context-based approach and its key phases used for improving the prediction of items ratings. The contextual information considered in this chapter is obtained from context-aware real-world datasets and is directly related to the RS entity `"Users"`. The proposed contextual recommendation approach addresses the weak points in the previous studies carried out in this area. Therefore, we focused on determining which contextual dimensions are relevant for a given recommendation system and how these dimensions interact with each other through a contextual dimensions weighting process. Thereafter, we aimed at finding out what is the contextual situation of a user, based on the combination of its contextual dimensions values by employing a rule-based inference engine using fuzzy logic. Finally, we predicted items ratings in two alternative ways using a neighborhood-based method and a matrix factorization-based method.
The remainder of this chapter is organized as follows. In the first section, we start by formalizing the problems related to the shortcomings of current context-aware recommendation approaches. Next, we define the multifaceted concepts of context

and then introduce the proposed context-based approach. In the end of this chapter, we summarize our work.

3.2 Problem Formulation and Positioning

The task of finding convenient items in multiple contextual dimensions has received much attention in previous researches and thus, different well working CARS in providing pleasing contextual recommendations exist. Nevertheless, existing context-aware recommendation systems are facing several challenges when generating recommendations. Most of these challenges pertain to how to deal with the different types of contextual dimensions. Given the potential abundance of contextual information that can be acquired, there are obviously more than one contextual dimension which can influence a user's decision. However, it is presumed that not all the obtained contextual dimensions are equally pertinent in affecting the user's ratings about an item in a significant way [16]. Generally, the different contextual dimensions do not possess similar degree of importance. Consequently, they impact differently on the items ratings and thus on the recommendations results. In this respect, Adomavicius et al. [16] showed that the degree to which the contextual dimension is integrated into the recommendation process influences the quality of the prediction of user's interests. Accordingly, if all the contextual dimensions are incorporated in the same manner into a recommendation approach without identifying the appropriate dimensions to be used, the prediction of the consumer preferences could not be accurate. A fundamental task in context representation is identifying appropriate contextual dimensions, since contextual dimensions that does not have a beneficial contribution to explain the variance of users ratings may deteriorate the prediction accuracy by adding noise [101].

Example 3.1. *As an example in TV recommendation, the user's current viewing context is exploited for personalized recommendations. For instance, a user might prefer to watch world news every day in the morning, the sports programs on weekends, and movies on friday night with friends at the cinema.*

Therefore, there could be several contextual dimensions that may affect the viewer interests such as:
- Time : represents the time of the day when watching TV.
- Location : indicates the actual location of the viewer.
- Occasion : represents the existing event in the viewer's calendar.
- Day of the week : represents the 7-day-week: Monday, .. , Sunday.
- Mood: indicates the state of mind of the viewer (e.g. happy, bored, unhappy)

- Companion: indicates the social company of the viewer.
In order to effectively incorporate the convenient contextual dimensions in the recommendation process, we have to consider two properties:

*- **Contextual relevancy**:*
In this case, the issue of identifying the contextual dimensions that perfectly influence the decision-making process is considerably important. This task makes it possible to characterize each contextual dimensions by a degree of importance to distinguish relevant contextual dimensions.

*- **Contextual correlation**:*
In the presented example, what users watch, when, and with whom may be correlated. Therefore, the interactions that may exist between the contextual dimensions must be considered in the recommender process.

As a result, the question that arises here and still considered as an open issue is: how to decide which relevant and correlated contextual dimensions to integrate in the recommendation process?

To accomplish this task, we need to examine whether an acquired contextual dimension should be used or not. Weighting and selecting the appropriate contextual dimensions that effectively impact the user's preferences is rarely studied in the past literature. Moreover, in many CARS researches, relationships between contextual dimensions are often unknown and uncertain which can negatively impact the ratings prediction quality when independent contextual dimensions are embedded.

Accordingly, obtaining accurate prediction of user's preferences undoubtedly counts on the degree to which we have integrated relevant and correlated contextual dimensions into the recommendation approach.

3.3 Context-based Predictive Model

3.3.1 User's Context

The large amount of the available items often compromises the users ability to select the content that best matches their preferences and that are perfectly adapted to their contexts. As previously shown, CARS seem to be the solution for this problem. In CARS field, the context is generally described as the circumstances effecting the user's decision in order to enhance items recommendation [102].

We remind the reader of the contextual terms employed throughout this manuscript, the term *contextual dimension* is generally understood to mean a contextual factor related to the current user's environment and that may effect his/her preferences, such as location, time, weather, etc. Each contextual dimension has different variables called *contextual conditions*. For example, the contextual conditions of weather dimension are: sunny, rainy, cloudy, etc.

Surveying previous context-aware recommender systems approaches allowed us to extract the most employed contexts types (see Table 3.1).

Temporal Context

Thanks to the development of smartphones, the time dimension is considered among the easiest collected contextual dimensions. The flexibility of time measurement implies various representations of temporal context. Therefore, to define the user's temporal context, time can be summarized into specific time periods across three layers: day, week and month. The days of the week can have two main classes of contextual conditions: work days (Monday to Friday), weekend or day off. Regarding the time of the day, we can use the five main day's parts, i.e., morning, midday, afternoon, evening and night as contextual conditions. When considering the months of the year, we can obtain many contextual conditions such as (month names, semesters, seasons, etc).

Locational Context

Different means could be employed to determine users' locations, such as the Global Positioning System (GPS) that expresses their exact position defined by an address or geographic coordinates. The location dimension of a given user that have an effect on his/her behaviour is more related to the semantic signification of physical entities than their names. For instance, it does not matter to know if the user is actually situated in "Olympico studium" or "Flaminio studium" but it is of importance to know that the user is located in a "studium" and more generally in an "entertainment location".

Physical Context

The physical context refers to the surrounding environmental conditions where the user is involved in, such as weather, lighting, temperature, sound, etc. In CARS, the most used physical contextual dimensions are related to the climate change. This is typical, for example, in the tourism area to guarantee tourists safety and satisfaction when visiting tourist sites. In this respect, we could find various contextual conditions of the weather dimension like sunny, windy, rainy, cloudy and snowy.

Emotional Context

The emotional context represents the special feeling that describes the user's state of mind. Human's emotional state could be used to upgrade the precision of user's decision making since it may impact concentration and issue resolution. The emotion contextual dimension could present several contextual conditions included in different classes: sadness, anger, joy, fear and surprise.

Social Context

The social context contains personal information like social relationships, interlocutors, tags, social explanations (likes or dislikes), etc. In the majority of CARS, social context is represented by the people around the user. Thus, the possible contextual conditions are: alone, with friends/family, with girlfriend/boyfriend, etc.

TABLE 3.1: Contextual dimensions and their possible contextual conditions

Context	Contextual dimensions	Contextual conditions
Temporal Context	- Part of the day - Day of the week - Month	morning, afternoon, evening and night work day, weekend/day-off january,..,december; spring or fall; winter,..,summer
Locational Context	Type of location	home, work/school, hotel
Physical Context	Weather	sunny, windy, rainy, cloudy, snowy
Emotional Context	Emotions	joy, sad, angry, happy, scared, surprised
Social information	Companion	alone, with friends/family, with girlfriend/boyfriend, etc.

3.3.2 User's Contextual Situation

Contextual situations are able to afford a higher-level specification of human behaviour compared to the contextual dimensions by presenting a projection on the multi-dimensional context space. Derived from the content provided in the previous chapter in Section 2.3.1, we adopt the following definition for a contextual situation: *A combination of potentially many contextual conditions*.

More formally, we define in the following the concerning sets starting from the contextual dimensions set to the contextual situations set.

- $Cd = \{cd_1, .., cd_k\}$ denotes the set of contextual dimensions, where k is the number of contextual dimensions:
  Example. In tourism area, the set Cd could contain the following contextual dimensions:
  Cd={season, trip type, weather}.

- $Cc_i = \{cc_{i1}, .., cc_{il}\}$ denotes the contextual conditions set concerning a specified contextual dimension Cd_i, where l is the number of the contextual conditions with respect to the contextual dimension Cd_i with $1 \leq i \leq k$.
  Example. The possible values of the contextual dimension (Cd_1: season) are represented by the following contextual conditions: Cc_1={summer, spring, autumn, winter}.

- $S = \{s_1, .., s_m\}$ denotes the set of contextual situations where $s_j \in S$ is a built up entity describing a contextual situation as a combination of the contextual conditions of k contextual dimensions $s_j = \{cc_{1j}, .., cc_{kj}\}$, with $1 \leq j \leq m$.
  Example. A contextual situation s_1 can be described by the contextual conditions of the three contextual dimensions season, trip type and weather as: s_1={summer, family trip, sunny}.

3.3.3 Model Structure

One of the assumptions underpinning our proposal is that the user's context when consuming an item is crucial for generating efficient recommendation. Therefore, the recommendation approach that we propose aims to exploit the user's contextual information, and to take advantage of this knowledge to do the rating prediction task. In fact, our main objective is about estimating the relevance (i.e. the rating) of items in accordance with the context of the user.

The motivations of the different elements of the context-aware recommendation approach that make up our contributions are based on the interpretations drawn from the literature review. In fact, from the insights gained from CARS literature, the studied CARS concerns and the properties of users context, the following three major key requirements can be derived:

- **Contextual dimensions weighting**: we address the thriving challenge of distinguishing the contextual dimensions that effectively impact the decision-making process. This task is achieved by employing a weighting method that provides the degree of importance of each contextual dimension and of subsets of dimensions. This method is able to identify not only contextual dimensions relevancy but also the correlations that may exist between them.

- **Contextual situation inference**: this step pertains to figuring out the user's current contextual situation from user's current context. It is particularly related to the relevant contextual dimensions having correlations between them included in the user's current context.

- **Contextual ratings prediction**: two ratings prediction models are suggested. The first one integrates the inferred contextual situation from the previous step in the rating prediction computing process. The second model makes ratings prediction according to the relevant and correlated contextual dimensions obtained via the weighting method from the first step.

More details on this will be given below.

3.3.3.1 Contextual Dimensions Weighting

A primary step that requires to be performed is to determine the degree of importance that needs to be attributed to each contextual dimension and each subset of contextual dimensions. More precisely, the aim of this step is to identify the fuzzy measure value of each contextual dimension that reflects its weight of importance. For a subset of contextual dimensions, the fuzzy measure value can be interpreted as the importance weight of the relationship that can exist between the dimensions involved in that subset. The major appeal of defining contextual dimensions fuzzy measures is their ability to model the relative importance of these dimensions as well as the interactions that may exist between them.
We present in what follows, how to define a fuzzy measure noted μ.

Definition 3.1 (Fuzzy measure [103]). Let $Cd = \{cd_1, .., cd_k\}$ denotes the contextual dimensions set and I_{Cd} represents the set of all subsets of contextual dimensions from Cd. A fuzzy measure is defined as a normalized monotone function $\mu : I_{Cd} \rightarrow [0, 1]$ such that: $\forall I_{Cd_1}, I_{Cd_2} \in I_{Cd}$ If $(I_{Cd_1} \subset I_{Cd_2})$ Then $\mu(I_{Cd_1}) \leq \mu(I_{Cd_2})$, with $\mu(I_{\emptyset}) = 0$ and $\mu(I_{Cd}) = 1$.

With the purpose of alleviating the heavy notations, $\mu(I_{Cd_i})$ will be denoted by μ_{Cd_i} in the remainder. More specifically, the value of μ_{Cd_i} specifies the weight of importance corresponding to the subset of contextual dimensions involved in Cd_i. To identify contextual dimensions fuzzy measures, we have chosen the least square method [104] in the KAPPALAB[1] package based on the R language which has the ability of tuning the best combination of fuzzy measures that should be assigned to the contextual dimensions. The least square method is recognized as being widely considered in the literature [105].

For the task of tuning the contextual dimensions weights, we begin by generating all possible weight combinations. For the sake of simplicity, a weight combination denoted as $\mu^{(.)}$ consists of the weight values affected to each individual contextual dimension as well as each subset of contextual dimensions. For k contextual dimensions, each weight combination comprises $(2^k - 1)$ weight values. For example, each weight combination μ^i for three given contextual dimensions dim_1, dim_2 and dim_3 can be written as: μ^i=$\{\mu_{dim_1}, \mu_{dim_2}, \mu_{dim_3}, \mu_{\{dim_1,dim_2\}}, \mu_{\{dim_1,dim_3\}}, \mu_{\{dim_2,dim_3\}}, \mu_{\{dim_1,dim_2,dim_3\}}\}$, where μ_{dim_i} is the weight value of the contextual dimension dim_i. The different weight values $\mu_{\{.\}}$ fall within the interval $[0, 1]$ and are obtained with a step of 0.1 such that the sum of the three contextual dimensions weights equals 1.

In the following, we describe the process for setting the initial weight values of contextual dimensions for initializing all possible weight combinations. For simplicity, we apply this process on three contextual dimensions (dim_1, dim_2 and dim_3).

- Step 1: at first, we attribute to the first contextual dimension dim_1 a higher weight value starting by 0.8. We tune the two remaining contextual dimensions dim_2 and dim_3 by a weight value equal to 0.1 for each, that is, the sum of the three weights of contextual dimensions equals 1. The weight of each subset of contextual dimensions ($\mu_{\{dim_1,dim_2\}}, \mu_{\{dim_1,dim_3\}}, \mu_{\{dim_2,dim_3\}}$}) is computed as the sum of the single dimensions weights involved in each subset. Afterwards, the weight value of dim_1 dimension is decremented by 0.1 and the weight value of dim_2 dimension is incremented, with the same pitch. We repeat this process until reaching 0.1 for the weight of dim_1 and 0.8 for the weight of dim_2.

- Step 2: we attribute to the second contextual dimension dim_2 a high weight value corresponding to 0.8. Then, we decrement the weight value of dim_2 and we increment the weight value of dim_3 with a step of 0.1 until it attains 0.8.

- Step 3: we set for dim_3 a high weight corresponding to 0.8. Then, we decrement the weight value of dim_3 and we increment dim_1 weight value with a step of 0.1 until it attains 0.8.

[1]https://cran.r-project.org/web/packages/kappalab/index.html

After initializing all possible weight combinations of contextual dimensions, we can then select the best one by computing an accuracy metric for each combination. Precisely, the best combination denoted by $\mu^{(*)}$ is the one that minimizes the rating prediction errors during the learning phase. Following this, we proceed to the application of the least squares method [104] which considers the best combination $\mu^{(*)}$ from the training set to provide an optimal weight combination that we denote $\mu^{(**)}$. In fact, $\mu^{(**)}$ comprises the fuzzy measures representing the final weights that should be assigned to each contextual dimension and each subset of dimensions. At this point, it became possible to identify the relevant and correlated contextual dimensions on the basis of their corresponding fuzzy measures.

3.3.3.2 Contextual Situation Inference

The basic idea behind contextual situation inference is that the user's context defined by a set of contextual dimensions may be modeled by a contextual situation issued from the relevant and dependent contextual dimensions. For instance, the contextual situation "meeting" can be inferred from the relevant and dependent contextual dimensions contained in user's context information such as "user is stationary", "user is located in the scheduled place at the scheduled time", "user is close to the meeting organizer". Therefore, step 1's outputs are required as the inputs of the current step.

In the literature, different ways could be used for identifying users contextual situations by employing: (1) fuzzy rules and reasoning engines (e.g., fuzzy logic, logic programming); (2) graphical inference tools (e.g., bayesian networks, belief propagation); (3) knowledge-based solutions operating on ontologies.

We choose to apply the first category using fuzzy logic for many reasons. First, fuzzy logic utilizes rules that can be created by non-experts, where human understandable notions with natural language terms are used. Consequently, fuzzy logic is considered promising as a description language. Another reason for employing fuzzy logic is that a training phase that could be very time consuming is no more required as would be needed if any supervised machine learning technique was employed. Added to that, fuzzy logic is considered as an efficient technique for solving the computational challenges that involve the manipulation of several variables [106]. Differently from other techniques such as bayesian networks, fuzzy logic finds its strength in providing lightweight solutions when merging different contextual dimensions. Indeed, bayesian networks cannot deal with diverse combined data information effectively since it only requires the discrete input data. In that case, the loss of information might happen and the contextual situation inference cannot be done appropriately. Besides, a contextual

situation is regarded as a reality that people live, perceive and reason about. Yet, this reasoning is naturally uncertain and ambiguous. To face these issues, fuzzy logic models human reasoning by mapping a user's experience and decision making to computer systems by adapting fuzzy rules.
More specifically, the fuzzy logic concept arises from the fuzzy set theory, which is a generalization of the traditional set theory proposed by Zadeh et al. in [107]. This tool could offer an effective flexibility for reasoning by designing approximate inferences on the basis of a set of supplied human language rules. Thus, the fuzzy inference has become one of the popular applications of fuzzy logic due to its ability of integrating human knowledge with its nuances.

To make contextual situation inference, we represent the input and output fuzzy variables by linguistic variables whose values are natural language words. Each linguistic variable is described by various linguistic terms, where each term has a name and a membership function. The role of the membership functions is to map the non-fuzzy inputs to fuzzy linguistic terms and vice versa. These functions can be modeled in mathematical forms. The most widely used types of membership functions are triangular, trapezoidal, and gaussian shapes. In general, the choice of the membership function type is made arbitrarily depending on the user experience [108].
To understand the process of fuzzy logic inference, we study its main components (fuzzifier, rules, inference engine, defuzzifier) and present the required steps based on these components:

1. **Fuzzification**: represents the initial step in the fuzzy inference process, in which the crisp inputs are switched to fuzzy inputs by employing fuzzy linguistic variables, fuzzy linguistic terms and membership functions. In our case, the system's inputs comprehend the relevant correlated contextual dimensions that sufficiently defines a situation. These important contextual dimensions are obtained from the weighting process described in Section 3.3.3.1. We also apply the interaction index [109] to interpret the most correlated contextual dimensions to consider. Their corresponding crisp values are extracted from users profiles gathered from real-world contextual datasets. Each input possesses a set of membership functions which contain all relevant values that the input can owns. As an example, we consider that V is a linguistic input variable corresponding to a contextual dimension from user's profile such as time of day and A refers to the range of V values. A fuzzy subset T of V depicts its corresponding linguistic terms representing the contextual conditions in our example (i.e.,morning, afternoon, evening, night). Thus, the linguistic input variable V is described by the triplet (V, A, T).

- V = time of day
- $A = R_+$
- T = {morning, afternoon, evening, night}
  - morning: if $V \in$[6am,12am]
  - afternoon: if $V \in$[12pm,5pm]
  - evening: if $V \in$[5pm,7pm]
  - night: if $V \in$ [7pm,12am]

Figure 3.1 illustrates the triangular membership function for the input variable time of day.

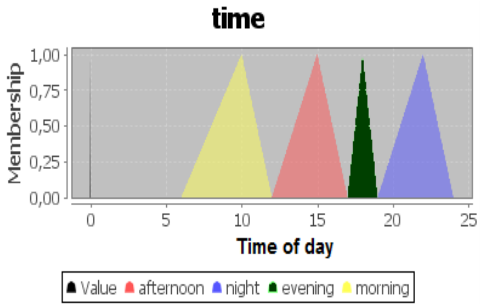


FIG. 3.1.: Fuzzy membership function for time of day

2. **Inference:** is based on the fuzzification step output containing fuzzy values of relevant and correlated contextual dimensions. The contextual situation inference step also requires a rule base involving a set of rules to perform the reasoning process and control the output variable. Ordinarily, the rules have the standard form of "IF condition THEN conclusion" clauses, where the condition is the antecedent part constituted by operators and terms and the conclusion is the consequence of inference. For instance, a fuzzy rule can be written as: If $a \in A$ and $b \in B$ then $c \in C$, where A, B and C are fuzzy sets. Precisely, the rules link the linguistic variables with their corresponding linguistic terms in the inference mechanism. For instance, if we would like to recognize the contextual situation of a music listener, we create the conclusion part by selecting some of the common contextual situations in which users could listen to music such as working, relaxing and exercising. Then, we create the IF-THEN rules according to the selected situations. In each rule, the antecedent part holds relevant and correlated contextual dimensions as well as their corresponding contextual conditions. In the following, an example of created IF-THEN rules by assuming that "location", "time" and "activity" are the relevant correlated contextual dimensions:

RULE 1 : IF time IS night AND location IS home THEN situation IS relaxing.
RULE 2 : IF time IS morning AND location IS office THEN situation IS working.
RULE 3 : IF activity IS sport AND location IS sport club THEN situation IS exercising.

In order to evaluate the presented fuzzy rules, we perform a set of fuzzy operations by applying the most common operators that examine how each rule's antecedent modifies the consequent. To accomplish this task, we use the minimum operator as an aggregation method for AND operator and maximum operator as an accumulation method to satisfy DeMorgan's Law [110].

3. **Defuzzification**: is the final step in the fuzzy logic inference process. Having completed the inference step, the obtained result is represented by a fuzzy value. This result needs to be defuzzified depending on the membership function of the output variable to get a final crisp situation value. The situations values and the degree of support for each rule can be computed through a defuzzification method. There are various methods for this purpose [111] such as the mean of maxima (MeOM) and the center of gravity (COG) defuzzification methods. The COG method is used here, since it is commonly used as a defuzzification method [112] thanks to its ability to avoid the discontinuities issue that could happen when using the MeOM method.

   To sum up, in our case:

   - the input for the fuzzy logic inference process corresponds to the relevant correlated contextual dimensions,
   - the fuzzification module transforms the crisp system inputs into fuzzy sets,
   - the fuzzy knowledge base stores the set of IF-THEN rules,
   - the inference engine simulates the human reasoning process by making fuzzy inference on the inputs and IF-THEN rules,
   - the defuzzifier is the part where a method of defuzzification is applied (COG) to obtain a non-fuzzy situation value,
   - the output represents the final decision corresponding to the inferred contextual situation.

3.3.3.3 Contextual Rating Prediction

The most fundamental step in a collaborative filtering recommendation system is to generate the output in terms of items ratings prediction. Therefore, in this step, we tackle the rating prediction problem that aims to estimate how much a user likes a particular item by counting his/her current context. We obtain the items predicted ratings through two proposed prediction methods based on the most widely used CF-based methods: neighborhood and factorization based methods.

Neighborhood-based method

Neighborhood-based methods have been originally tasked for providing items predicted ratings to distinguish what items a user is interested in based on like-minded users. Therefore, we take as our starting point the popular neighborhood based Resnick's algorithm [113] which treated user-based CF recommendation as a prediction problem to provide non-contextual items ratings prediction. Inspired by the idea of the paper [114], we adapt the rating prediction formula of Resnick's algorithm [113] to generate contextual ratings prediction through a novel proposal called Fuzzy Weighting Recommender (FWR). In other words, we aim to predict the rating for an item of a target user according to the user's contextual situation. Therefore, we additionally take into consideration users inferred contextual situations as well as the similarity between neighbors contextual situations to provide more reliable predictions. In this prediction process, the notion of contextual situations similarity is introduced, where the more close the contextual situations of two ratings were given, the more reliable those ratings for further predictions. Nevertheless, this effect should be restricted since integrating contexts with low similarity can lead to adding noise to the predictions. Thus, a set of similarity thresholds are introduced to filter ratings, for the each component.

According to FWR, the predicted rating $P_{a,i,\sigma}$ that a given user a is expected to attribute to the item i depending to his contextual situation is computed as follows:

$$P_{a,i,\sigma} = \bar{\rho}(a, \sigma_3, \epsilon_3) + \frac{\sum\limits_{n \in N_{a,\sigma_1,\epsilon_1}} (\rho(n, i, \sigma_2, \epsilon_2) - \bar{\rho}(n, \sigma_2, \epsilon_2)) \times sim_w(a, n, \sigma_4, \epsilon_4)}{\sum\limits_{n \in N_{a,\sigma_1,\epsilon_1}} sim_w(a, n, \sigma_4, \epsilon_4)} \quad (3.1)$$

The key parameters that the presented prediction formula 3.1 contains are the weight vectors σ of the contribution of the contextual situations in each component, and the ϵ values that fix the threshold of the similarity between the contextual situations in each component. The different components of formula 3.1 are described as follows:

- **Neighborhood Selection**. The first step that needs to be conducted in the prediction process is to select the neighbors who are similar to the user a. To identify the closest neighbors, we compare their corresponding contextual situations for rating a given item i with the target contextual situation sit of user a and considering them as similar users if the contextual situation similarity measure exceeds the threshold value ϵ_1. In some cases, a neighbor n may rate an item in various contextual situations, so we select the maximally-similar contextual situation when using the threshold. In what follows, we define this operation designated by N_{a,σ,ϵ_1}, where $SitSim$ represents the semantic similarity between the current contextual situation sit and a different contextual situation sit_n.

$$N_{a,\sigma,\epsilon_1} = \{n : \max_{r_{n,i,sit_n}} (\mathit{SitSim}(sit, sit_n, \sigma)) > \epsilon_1\} \tag{3.2}$$

- **Neighbor Contribution**. In some cases, the user may assign various ratings for the item i in several contextual situations that match the current situation sit to different degrees. These ratings require to be merged in order to subtract an overall weighted average of all ratings issued in similar contextual situations. The weighted average function based on contextual similarity is defined as:

$$\rho_{(n,i,\sigma,\epsilon_2)} = \frac{\sum\limits_{r_{n,i,sit_n} \ni SitSim(sit,sit_n,\sigma) > \epsilon_2} r_{n,i,sit_n} \times SitSim(sit, sit_n, \sigma)}{\sum\limits_{r_{n,i,sit_n} \ni SitSim(sit,sit_n,\sigma) > \epsilon_2} SitSim(sit, sit_n, \sigma)} \tag{3.3}$$

The selected ratings by this function are those given by users n for item i involved in contextual situations at least ϵ_2 similar to sit. We nominate by I_n the set of items given by user n. The global average of all items assessed in similar contextual situations is given by:

$$\bar{\rho}_{(n,\sigma,\epsilon_2)} = \frac{\sum\limits_{i \in I_n} \rho_{(n,i,\sigma,\epsilon_2)}}{|I_n|} \tag{3.4}$$

- **User Baseline**. User baseline depicts the global ratings average of the target user for items in similar contextual situations, which is $\bar{\rho}_{(a,\sigma,\epsilon_3)}$.

- **User Similarity**. We can determine the similar users set U_{ϵ_4} by gathering all rated items i by the users a and n in their corresponding contextual situations sit and sit_n, respectively, such that $SitSim(sit, sit_n, \sigma) > \epsilon_4$

$$U_{\epsilon_4} = \{(i, sit, sit_n) \ni \exists r_{a,i,sit}, r_{n,i,sit_n} \wedge SitSim(sit, sit_n, \sigma) > \epsilon_4\}.$$

Having obtained the ratings and their contextual situations, we are able to get the weighted version of the correlation function by the following equation:

$$sim_w(a,n,\sigma,\epsilon_4) = \frac{\sum\limits_{(i,sit,sit_n)\in U_{\epsilon_4}} (r_{a,i,sit} - \bar{r_a})(r_{n,i,sit_n} - \bar{r_n}) \times SitSim(sit, sit_n, \sigma)}{\sqrt{\sum(r_{a,i,sit} - \bar{r_a})^2 \sum(r_{n,i,sit_n} - \bar{r_n})^2 \sum\limits_{(i,sit,sit_n)\in U_{\epsilon_4}} SitSim(sit, sit_n, \sigma)}} \tag{3.5}$$

Matrix factorization-based method

Matrix factorization plays an important role in recommendations being among the most efficient algorithms for ratings prediction. Furthermore, in the area of contextual recommendation, MF could be applied to a wide variety of contexts allowing easy integration of multiple contextual dimensions unlike other context-aware ratings predictors.

However, the majority of the surveyed CAMF recommendation methods in the previous chapter (Section 2.2.5), cannot fully capture the impact of the relevant contextual dimensions as well as their associations on the predicted ratings. To tackle this shortcoming, we put forward an improved CAMF recommendation model on the basis of the fuzzy measures of contextual dimensions. This proposal consists of two strategies extended from the correlation based CAMF-MCS model suggested in [44]. Both of the two proposed strategies apply a common rating prediction formula (Equation 3.6) highlighting the notion of "contextual correlation". The underlying assumption behind that notion is that, the more correlated two contexts are, the more two recommendation lists for a same user in those two contexts are similar too.

$$\hat{r}_{u,i,s_t} = \vec{q_i}.\vec{p_u}.Corr(s_t, s_E) \tag{3.6}$$

In the rating formula 3.6, both items and users are characterized by vectors. In fact, each item i is associated with an item vector denoted $\vec{q_i}$ and each user u is associated with a user vector denoted $\vec{p_u}$. Those vectors values are the weights on different latent factors. Precisely, the elements in $\vec{q_i}$ indicate the extent to which the item i obtains those latent factors. For the vector $\vec{p_u}$, its elements indicate how much users like those latent factors. The function denoted ($Corr(s_t, s_E)$) predicts the correlation or the similarity between a current contextual situation s_t in which the user u consume the item i and an empty contextual situation s_E.

We adjust this function for each strategy of our proposed CAMF recommendation model. In fact, according to our contributions, we aim to offer:

(i) A weighting strategy (named WCAMF-MCS) to integrate the weight of relevant contextual dimensions in the correlation function ($Corr(s_t, s_E)$);

(ii) An interaction strategy (named ICAMF-MCS) to incorporate relatedness measurement between interacted contextual dimensions in the correlation function ($Corr(s_t, s_E)$).

In our proposed CAMF recommendation model, we assume that the contextual dimensions form a multidimensional coordinate system. In this system, a real value is assigned for each contextual condition belonging to those dimensions, such that each contextual condition can locate a position in the corresponding axis. Therefore, a contextual situation can be represented by a point in the multidimensional space. As a result, the distance separating two such points can serve as the basis for a correlation measure. We follow the standard optimization in matrix factorization for parameters learning. Thus, we use the Stochastic Gradient Descent (SGD) method to learn the user and item vectors, besides the real values representing the contextual conditions positions by minimizing the ratings prediction errors. In Equation 3.8, we present the general form of the loss function, where e denotes the prediction error computed in Equation 3.7 based on the real and predicted ratings r_{u,i,s_t} and $\hat{r}_{u,i,s_t}$ respectively. The parameter α represents the regularization term that deals with the overfitting problem.

$$e = r_{u,i,s_t} - \hat{r}_{u,i,s_t} \tag{3.7}$$

$$_{p,q,Corr}\frac{1}{2}(e)^2 + \frac{\alpha}{2}(||\overrightarrow{p_u}||^2 + ||\overrightarrow{q_i}||^2 + Corr^2) \tag{3.8}$$

The update of the user and item vectors can be viewed in the following equations, where β represents the learning rate.

$$\overrightarrow{p_u} = \overrightarrow{p_u} + \beta.(e.\overrightarrow{q_i}.Corr(s_t, s_E) - \alpha.\overrightarrow{p_u}) \tag{3.9}$$

$$\overrightarrow{q_i} = \overrightarrow{q_i} + \beta.(e.\overrightarrow{p_u}.Corr(s_t, s_E) - \alpha.\overrightarrow{q_i}) \tag{3.10}$$

We present also the update of the contextual conditions positions $P_{cc_{j,t}}$ and $P_{cc_{j,E}}$ of a particular contextual dimension cd_j, where the distance between the two contextual situations s_t and s_E is referred by $Dist$.

$$P_{cc_{j,t}} = P_{cc_{j,t}} + \beta(e.(\overrightarrow{p_u}.\overrightarrow{q_i})\frac{P_{cc_{j,t}} - P_{cc_{j,E}}}{Dist} - \alpha P_{cc_{j,t}}) \tag{3.11}$$

$$P_{cc_{j,E}} = P_{cc_{j,E}} + \beta(e(\overrightarrow{p_u}.\overrightarrow{q_i})\frac{P_{cc_{j,t}} - P_{cc_{j,E}}}{Dist} - \alpha P_{cc_{j,E}}) \tag{3.12}$$

In what follows, we describe each strategy of the proposed CAMF recommendation model:

1. **WCAMF-MCS strategy**

   In accordance with their different importance degrees, the contextual dimensions impact differently the prediction of ratings. Accordingly, we integrate the obtained weights of relevant contextual dimensions into the correlation function ($Corr(s_t, s_E)$) included in the rating prediction formula (3.6). Since the correlation function can estimate the similarities between contextual situations, it can be measured as the inverse of the distance separating two data points corresponding to two contextual situations. Thus, we choose to use a weighted euclidean distance measure to compute the contextual situations distances while incorporating the weight of importance associated to each contextual dimension in these situations. For this task, the real values corresponding to the contextual conditions positions should fall into the interval [0, $\frac{1}{\sqrt{k}}$] (k is the number of contextual dimensions) to make sure that the obtained distances are within the interval [0, 1].

   Therefore, for the WCAMF-MCS, the correlation function of Equation 3.6 can be computed as follows:

   $$Corr(s_t, s_E) = 1 - Dist \tag{3.13}$$

   $$Dist = \sqrt{\sum_{j=1}^{k} \mu_{cd_j} (P_{cc_{j,t}} - P_{cc_{j,E}})^2} \tag{3.14}$$

   Where:

   - μ_{cd_j}: the importance weight of the contextual dimension cd_j.
   - $P_{cc_{j,t}}$: the position of the contextual condition $cc_{j,t}$ of the contextual dimension cd_j in s_t.
   - $P_{cc_{j,E}}$: the position of the contextual condition $cc_{j,E}$ of the contextual dimension cd_j in s_E.

2. **ICAMF-MCS strategy**

   There are useful interactions that may exist between contextual dimensions and it is of great importance to take these interactions into account when predicting items ratings to improve the prediction accuracy. For this reason, we decide to incorporate the obtained weights of interacted contextual dimensions in the rating prediction process. For this task, we compute the correlation function ($Corr(s_t, s_E)$) by employing a similarity measure in which we can integrate interaction measurement among correlated contextual dimensions. In order to do so, we apply a similarity measure based on Choquet integral model [115] that considers the synergy among criteria represented by fuzzy measures. In our case, we use the fuzzy measures to represent the importance degree of each subset of correlated contextual dimensions, which gives rise to a more flexible representation of interaction between dimensions. Instead of being a weighted average model, the Choquet integral model is used, this time, to compute a similarity measure by integrating the weights of interacted contextual dimensions. Therefore, for the ICAMF-MCS strategy, we formulate the correlation function of Equation 3.6 as follows:

$$\begin{aligned} Corr(s_t, s_E) &= 1 - (\int_{Cd} f \, d\mu)^{1/p} \\ &= 1 - (\sum_{j=1}^{k} (\mu_{Cd_j} - \mu_{Cd_{j+1}}) f(cd_j))^{1/p} \end{aligned} \tag{3.15}$$

Where:

- $Cd_j = \{cd_j, ..., cd_k\}$: a correlated contextual dimensions set.
- μ_{Cd_j} : the weight of importance of the set $\{cd_j, ..., cd_k\}$
- $f(cd_j)$=$|P_{cc_{j,t}} - P_{cc_{j,E}}|^p$, $p \in [1, +\infty]$.

In our experiments, the similarity measure is computed using $p = 2$ which is commonly employed as the suitable value ([115, 116]).

3.4 Conclusion

In this chapter, we introduce the proposed context-based recommendation approach that estimates items ratings in respect with the contextual information of the user. In our work, we exploit and integrate user's contextual information in two manners. On one hand, we employ a weighting method based on fuzzy measures to identify not only contextual dimensions relevancy but also contextual dimensions correlation. Unlike other research carried out in this area which deal with these two topics separately, the proposed method has the role of determining the importance weight of each contextual dimension and also of each subset of dimensions to distinguish the most relevant and correlated ones. On the other hand, we use an efficient technique based on fuzzy logic to infer user's contextual situation by fusing relevant and correlated contextual dimensions. The outputs from both mentioned ways of exploiting the contextual information are then integrated for producing items predicted ratings. This task is performed by using two commonly used CF-based methods, which gave rise to two novel prediction models: a neighborhood-based model and matrix factorization-based model.

We will show, in the next chapter, how to implicate user's contextual information with items criteria information in one recommender.

Chapter 4

Engaging Context and Criteria Information in Recommendation

4.1 Introduction

In seeking to increase the recommendation accuracy, many researches have looked for extending traditional recommendation approaches by including useful additional information. In this respect, significant research efforts have been devoted to Context-Aware Recommender Systems (CARS) including our proposal described in the previous chapter, where context-specific dimensions of information (e.g., weather, time, location, etc.) are incorporated in the recommendation process. However, nearly all existing CARS are still based on a single criterion, a numerical rating, representing how a user likes a specific item. This kind of single-criterion recommender systems could not always meet users personalized requirements and thus, Multi-Criteria Recommender Systems (MCRS) have been emerged. In fact, MCRS take advantage of describing an item across multiple criteria representing its properties and considering the user's evaluation on each individual criterion to provide more personalized recommendations. Both CARS and MCRS are well addressed but separately in the extant literature. In this chapter, we introduce a context-aware multi-criteria recommendation approach that attempts to improve the recommendation quality by considering users' multi-criteria ratings under specific contexts.

4.2 Problem Formulation and Positioning

Several new recommender systems were developed to enhance the recommendation results and adapt to novel applications by integrating richer information in the recommendation process. In this respect, standard recommendation approaches are extended through different directions to provide more personalized recommendations by making use of the available information. One of these directions is dedicated to context-aware recommender systems. Generally, CARS extend traditional recommendation approaches by integrating contextual information in the utility function that predicts the user's preferences with accordance to his/her context. We have already discussed in Chapter 2 (Section 2.2.5) the different existing CARS approaches [34–45] and their ability to provide high level of recommendation accuracy outperforming the traditional ones. Albeit CARS exploit potentially relevant data for producing recommendations, they are still based on a single-criterion utility function. The main underlying assumption is that users rate the extent to which a user is interested in an item with respect to one single objective criterion. However, this assumption is limited [86], because in real-life applications, users express their preferences about item-related facets encompassing their subjective preferences. Besides, in CARS, the item criteria and their strengths might evolve while context evolves. Therefore, single-criterion based CARS may not be sufficient to express user's interest about an item whenever the item inherently has multiple features or criteria, especially in domains like hotels, restaurants, movies, etc. This perspective raised the importance of modeling the user's utility of an item as a vector of ratings along several criteria and has been the focus of a separate research direction dedicated to multi-criteria recommender systems. Roughly speaking, MCRS learn the utility function from a number of utility clues which impact the user's perception of item usefulness and interact each other as ratings are made. The research experiments that have been conducted on MCRS, proved that these approaches perform significantly better than the single-criterion ones [86]. Nevertheless, the existing multi-criteria models doesn't integrate contextual information into the recommendation process. Both context and criteria information play a crucial role in recommendation since the information they present can impact users decisions. Accordingly, much research work has been carried out on multi-criteria and context-aware recommendation directions applied in the real-world applications, such as movies [66, 67, 117], tourism [10, 38, 43] and music [77, 80]. However, these two directions have been considered separately from each other forming two distinct directions.

In our work, we attempt to contribute to this under-explored research area where the overall goal is to take advantage of combining the benefits of the two mentioned directions. Specifically, we explore the idea of capturing more fine grained preferences by using more detailed items feedback based on their criteria ratings under the specific contexts in which these items are consumed.

As an illustration, in the well known website of hotel reviews (TripAdvisor[1]), many comments are given by users describing their stay experiences by highlighting their preferences with ratings upon hotel criteria (e.g., location, service *etc.*) and their contexts (e.g., the season of traveling, travelling companion *etc.*). Certainly, a seaside hotel would be convenient during a summer vacation, where the importance of considering both the location criterion with the temporal contextual dimension for selecting a suitable hotel.

Despite this interest, nearly all the recommendation approaches have considered context-awareness or multi-criteria decision making independently. As far as we know, only a small minority of recommendation studies [97, 98] (Section 2.5) addressed the problem of adding contextual information and multi-criteria ratings into a single recommender.

Having different multi-dimensional data to be considered in the recommendation process, a crucial question arises here : How accurate items recommendations and ratings prediction can be produced by modeling all the available data ?

In contradiction with earlier closest studies [97, 98], we highlight the multi-dimensional data modeling by proposing an efficient representation of the available data. In fact, we define a new representation to model the dimensions of the context and the users feedback on the items criteria. Then, when it comes to the task of ratings prediction, we design a novel strategy for this step, contrary to what was previously done by closely related work [97, 98] that applied classical existing strategies to provide ratings prediction or recommendations for items.

4.3 Contextual Multi-criteria Recommendation Via Bipartite Graph Modeling

In this section, we present the proposed recommendation approach which aims to integrate the context and criteria information in the recommendation process. To handle all the available input data, we examine the recommender's data from the graph theory based perspective by modeling the associations between two types of entities (users contexts and criteria) as a bipartite graph. Starting from the assumption that contextually similar users tend to provide similar preferences for item criteria, we rely on the joint clustering of users contexts in the one hand and items criteria in the

[1] http://www.tripadvisor.co.uk/

other hand. Accordingly, we conjecture that clustering users involved within similar contexts induces the clustering of their preferred criteria and inversely. Following this assumption, we obtain two types of entities simultaneously assigned to clusters resulting from the joint clustering of contextual recommendations embedding users with similar criteria interests for items under similar contexts. When it comes to the rating prediction step, we assume that the importance of criteria might vary among users even among those with similar criteria preferences. Based on this idea, we suggest to aggregate users' predicted criteria ratings from the co-clusters according to their particular preferences.
In summary, we turn our recommendation problem into two main sub-problems:

- The prediction of item multi-criteria ratings based on the co-clusters of contextual ratings : starting from a bipartite graph modeling the users contexts and the criteria entities, we try in the first sub-problem to solve the bipartite graph partitioning problem by a simultaneous clustering technique. Then, we apply a rating prediction algorithm for predicting the criteria ratings issued from the obtained co-clusters.
- Computing the overall user's assessment of an item : the key issue within this sub-problem is designing a suitable aggregation strategy for the predicted multi-criteria ratings to obtain the overall item rating. Contrary to previous aggregation strategies, we underline the context-dependent importance of each criterion for each user to provide more flexible customization of the overall assessment results.

4.3.1 Context-Aware Multi-criteria Recommendation Framework

4.3.1.1 Basic notation and problem definition

We first present the basic terminologies we are going to use throughout this chapter in a clear way. Then, we formulate the context-aware multi-criteria based recommendation problem.
User's situational context. In recommendation systems area [44], a contextual situation can be viewed as a set of contextual dimensions values reflecting user's state. In our bipartite graph, the user's situational context entity will be used to refer to the distinct pair (user, context) representing the user involved in a specific surrounding contextual dimension (e.g., location, time of day, *etc.*). More formally, we denote by U the set of users $U = \{u_1, .., u_k\}$, where k represents the total number of users, and we denote by Co the set of contexts represented as $Co = \{co_1, .., co_l\}$, where l is the total number of contexts. A user's situational context is built up as an entity denoted as s_{ij}, that

refers to the pair consisting of the user u_i involved in the context co_j. For the sake of simplicity, we denote the user's situational context s_{ij} as s_i where $1 \leq i \leq m$ resulting the whole situational contexts set formed as $S = \{s_1, .., s_m\}$.

Criteria. The set of criteria refers to the rated item aspects representing users criteria preferences in different situational contexts. This set is noted $C = \{c_1, .., c_n\}$, where n is the total number of criteria of a given item.

Example. For the item hotel, the set of criteria can be expressed as $C = \{$cleanliness, service, room quality$\}$.

Bipartite graph. The bipartite graph is a special model in graph theory, which is very significant in the research field of many practical applications [118–120]. In the field of recommendation systems, the most used bipartite graphs model the connections from one part of the graph, users, to the other part, items. Generally, a bipartite graph is defined by the triplet G=(X,Y,E) composed of two vertex sets X and Y and an edges set E connecting only nodes from different vertex sets (E=$< i, j >$ | i ∈ X, j ∈Y). In our bipartite graph, X and Y represent situational contexts and criteria vertex sets respectively. For care of the simplicity of the notations, X and Y are noted as S and C respectively and thus we obtain the triplet G_{SC} = (S,C,E^{SC}) where E^{SC} is the edges set of ratings connecting nodes from vertex S to vertex C such as (E^{SC}=$<s_i$, $c_j>$ | $s_i \in$ S, c_j ∈C). In this graph, there are no intra-relationships. In other words, there are no edges that connect between situational contexts nor between criteria.

Context-aware multi-criteria recommendation problem. The pending problem consists in predicting items ratings for users according to their contexts and criteria preferences. We turn this problem into two sub-problems: (1) identifying co-clusters in the bipartite graph G_{SC}; (2) predicting users preferences for items through predicting users preferences over the items multiple criteria C.

4.3.1.2 Bipartite graph co-clustering

Once the bipartite graph is formed based on a rating matrix composed of users situational contexts as rows and criteria as columns, the next step includes its partitioning to generate criteria ratings prediction. To solve the bipartite graph partitioning problem, we count on the following hypothesis:

H: "Users in similar contexts tend to have similar interests for similar criteria".

Therefore, we aim to identify clusters including like-minded users in similar situational contexts and criteria that these users are particularly interested in. As a result, we attempt to simultaneously partitioning the bipartite graph entities containing users situational contexts and criteria into sub-groups. For this task, we apply the spectral co-clustering algorithm [118] that has achieved efficient performance in different tasks on graph theory [118, 121]. The general principle of spectral co-clustering is to approximate the normalized cut of the bipartite graph to identify the co-clusters. An approximate solution to obtain the optimal normalized cut can be achieved by decomposing the normalized $m \times n$ rating matrix R whose rows correspond to users situational contexts and columns to criteria such that R_{ij} is equal to the graph edge E_{ij}^{SC}. Thus, the matrix R is first normalized as follows: R_n= $D_1^{-1/2}$ R $D_2^{-1/2}$, where D_1 represents the diagonal matrix with entry i defined by $\sum_j R_{ij}$ and D_2 represents the diagonal matrix with entry j defined by $\sum_i R_{ij}$. After that, the desired partitions of the rows and columns of R are provided via the Singular Value Decomposition (SVD) of the resulting matrix $R_n = U\Sigma V^\top$. More precisely, the factorization via SVD consists of finding the $m \times m$ matrix U, the $m \times n$ diagonal matrix Σ, and the transpose of the $n \times n$ matrix V. The columns of U and the columns of V represent the left and right singular vectors of R_n, respectively. The users situational contexts partitions will be given by a subset of the left singular vectors, and the criteria partitions will be given by a subset of the right singular vectors. These singular vectors are then employed for building the matrix Z as follows:

$$Z = D_1^{-1/2} U D_2^{-1/2} V \tag{4.1}$$

Lastly, the desired co-clusters are obtained via the decomposition of the resulting matrix Z by k-means++ algorithm. The identified co-clusters are then adopted in the next step of rating prediction detailed below.

4.3.1.3 Rating prediction algorithm

Our rating prediction process is based on users multi-criteria ratings under specific contexts. Generally, there are two ways to enhance the multi-criteria recommendation algorithms. On one hand, it is useful to improve the rating prediction on each criterion. On the other hand, it is important to improve the overall rating prediction based on the aggregation of the predicted multi-criteria ratings. In our approach, we draw our attention especially to the second work by exploring new manners to aggregate the predicted multi-criteria ratings to obtain overall ratings.

Criteria Ratings Prediction

The first stage in our rating prediction process consists in predicting the multi-criteria ratings. In this step, we rely on the co-clustering results obtained from the previous step with a rating prediction algorithm. What differentiates our work from previous prediction methods is that we make personalized criteria ratings prediction from co-clusters of like-minded users in similar contexts and criteria in which these users share their interests.

We present in what follows the rating prediction algorithm (Algorithm 1) that outputs the predicted criteria ratings for each co-cluster. It takes as input parameters the rating matrix R containing criteria as columns and users situational contexts as rows, the co-clusters number L and the factors number F. The number of co-clusters may significantly impacts the rating prediction accuracy. Thus, we experimentally tune it in Chapter 6.

Algorithm 1 Criteria Ratings Prediction for each Co-cluster

Input: Rating matrix with multi-criteria: $R \in \mathbb{R}^{m \times n}$, the number of co-clusters: L, and the number of factors: F.
Output: Criteria predicted ratings in each co-cluster
Begin

1. **For** each co-cluster $k \in \{1, .., L\}$ **do**
2. R_k=ExtractSub-matrix (R,co-$cluster_k$)
3. P_k,Q_k=MatrixFactorization(R_k, F)
4. **For** each $i \in P_k$ **do**
5. **For** each $j \in Q_k$ **do**
6. **For** each $t \in \{1, .., F\}$ **do**
7. $\hat{r}_{ij}$= $p_{i,t} \times q_{j,t}$

End.

At first, we begin by extracting for each co-$cluster_k$ a rating sub-matrix $R_k \in \mathbb{R}^{m_k \times n_k}$ from the original rating matrix $R \in \mathbb{R}^{m \times n}$, where m_k and n_k represent the users situational contexts number and the criteria number in co-$cluster_k$. Then, we apply the Matrix Factorization (MF) algorithm [122] for predicting the criteria ratings in each obtained sub-matrix R_k. Our choice for the MF algorithm is motivated by its significant effectiveness in solving the rating prediction problem [58, 122, 123]. To apply the MF algorithm, the $MatrixFactorization$ function is called (line 3), where we suppose that there are F hidden factors capturing users situational contexts features as well as criteria features to model users preferences. More specifically, each rating sub-matrix R_k is decomposed into the product of two matrices P_k and Q_k with lower dimensionality. The matrix P_k represents users situational contexts, where each row of P_k would describe the strength of the associations between a user's situational context and the features. While the matrix Q_k represents the criteria, where each row of Q_k would describe the strength of the associations between a criterion and the features. We learn both matrices P_k and Q_k with the Stochastic Gradient Descent (SGD) method which minimizes the rating prediction errors. Finally, the predicted rating $\hat{r}_{ij}$ of the user's situational context s_i for the criterion c_j is calculated as described below:

$$\hat{r}_{ij} = p_i {q_j}^T \tag{4.2}$$

Overall Rating Prediction

Finding the aggregation function is crucial in multi-criteria recommender systems. In fact, the multiple criteria combination plays an important role to serve for overall item rating prediction. However, this task has not raised the attention it deserves. There are various ways for gathering the multi-criteria ratings. Yet, the most widely used forms of aggregation are the average aggregation and its variations because of their simplicity [124]. But these functions are not always suitable as they do not reveal the optimal weights of the various criteria based on specific users preferences. Contrary to conventional aggregation forms, we propose a novel strategy to combine the predicted multi-criteria ratings into an overall rating. In fact, to tackle the limit of the compensatory property of previous aggregation forms, we rather exploit prioritized aggregation operators [125] which are relevant in situations where we consider that criteria are not of equal strengths. The used priority-based aggregation operators are the `"Scoring"` and `"And"` operators [126, 127]. They were applied in a number of real applications in general [128, 129], and in particular in the information retrieval field [126, 127, 130]. However, as far as we know there are no attempts to apply these prioritized operators in the recommendation systems field. As it will be seen, these operators apply a weighted aggregation, where the weights of criteria are based on the users preference order of criteria estimated from their expressed criteria ratings. Furthermore, with regard to the problem of context-aware based recommendation at hand, we conjecture that the strength of each criterion also depends on users contexts. Thus, applying the prioritized operators offers a high personalization of the overall rating prediction results according to the criteria weights which are based on users preferences on multiple criteria under specific contexts.

We assume that the preference order over the criteria is user-dependent. This is due to the fact that each user may express his specific preference on each criterion in a given context. Therefore, this aspect induces different importance weights associated to each criterion for the same item based on the user's preferences under different contexts.

Therefore, finding the importance weight of a given criterion c_i , with i $\neq 1$, is closely related to users preferences order over the considered criteria, and also relies on both the weight of the criterion c_{i-1} (of greater priority with respect to c_i), and the preference of c_{i-1}. To find the user preference order of the considered criteria, we rely on computing an average score for each criterion based on the user expressed criteria ratings. More formally, we define by $C = \{c_1, ..., c_n\}$ a finite set of ordered criteria, where c_1 is the most preferred criterion with the highest order and thus the criterion c_n presents the least preferred one. We denote by w_p the importance weight associated to the criterion $c_p \in C$ for a given item and in a particular context. The weights computation process of the ordered criteria can be formalized as follows:

- The weight attributed to the most preferred criterion c_1 is set to be 1 ($w_1 = 1$).
- The weights of the remainder criteria c_p ($p \in$ [2, n]), are computed according to the following formula:

$$w_p = w_{p-1}.r_{p-1} \tag{4.3}$$

Where r_{p-1} represents the preference rating given to the criterion c_{p-1} and w_{p-1} denotes its corresponding weight.
In the following, we take advantage of the prioritized "Scoring" and "And" aggregation operators to define novel aggregation functions that represent the relationship between the overall rating and the individual criteria ratings:

1. **Prioritized "Scoring" operator (F_s)**: this operator computes the overall item rating denoted as r_0 from the item criteria evaluations by considering each criterion weight. Therefore, the aggregation of these criteria ratings with the "Scoring" operator (F_s) is defined as:
   $F_s : [0,1]^n \longrightarrow$ [0, n]
   $$r_0 = F_s(r_1, .., r_n) = \sum_{p=1}^{n} w_p.r_p \tag{4.4}$$
   Intuitively speaking, the more higher the satisfaction degree of a more important criterion, the more the satisfaction degree of less important criterion effects the item overall rating.

2. **Prioritized "And" operator (F_a)**: this operator computes the overall item rating r_0 in accordance with the importance of just one criterion, that is, the least satisfied criterion. The aggregation with the "And" operator (F_a) is defined as follow:
   $F_a : [0,1]^n \longrightarrow$ [0, 1]
   $$r_0 = F_a(r_1, .., r_n) = \min_{p \in [1,n]}(\{r_p\}^{w_p}) \tag{4.5}$$
   For F_a operator, the extent to which the least satisfied criterion is considered depends on its importance for the user. When the least satisfied criterion is considered as the most important criterion, its value is viewed as the overall item rating merely. Otherwise, the values of the other criteria are considered, depending on their importance.

4.4 Contextual Multi-criteria Recommendation Via Tripartite Graph Modeling

In the present section, we aim to extend our prior context-aware multi-criteria proposal (Section 4.3) that introduced a recommendation model based on bipartite graph partitioning. In fact, in the previous model, we only considered a single contextual dimension associated with each user, while real-life applications give rise to multidimensional contexts with interactive criteria. Therefore, we intend to work toward enriching our previous model by extending the bipartite graph to deal with different context nodes and the relevant interactions between contexts and item criteria. This idea leads us to propose new research directions that we will detail in the following key contributions:

- For modeling the multi-dimensional input data, we explore the idea of presenting the pertinent entities coming from the heterogeneous related recommendation data as a tripartite graph with three types of linked entities (users, contextual situations and criteria). More precisely, we extend our previous bipartite graph based-model to handle more entities including the criteria entities, users entities and contexts entities comprising a set of contextual dimensions values representing the contextual situations in which these users are involved. Furthermore, we emphasize a new challenge through the tripartite graph modeling, involving the weighting of the graph entities links on the one hand, and the simultaneous clustering of these entities on the other hand:
  - To reveal the latent connections between the different types of entities, we weight the relationships between users and their contextual situations as well as between users and their evaluated criteria.
  - To solve the tripartite graph partitioning problem, we create a research hypothesis by considering the graph entities and their relationships to afford insights about the suitable co-clustering structure. According to the posed hypothesis, we substitute the two-order co-clustering used in our previous work by a high-order co-clustering represented by a consistent fusion of two bipartite graphs co-clustering sub-problems.
- For the rating prediction step, we replace the algorithm used in our previous work by a novel rating prediction process that runs in two stages. In the initial stage of the process, the dependency between contexts and users is considered in a particular low dimensional recommendation space. Then, the correlation between criteria is taken into account to be integrated in the prediction process.

4.4.1 Situational Multi-criteria Recommendation Framework

4.4.1.1 Basic notation and problem definition

We first give the definition of the used basic terms. These terms are principally the elements included in the multi-dimensional matrix that we denote M of the $Users \times Items \times ContextualSituations \times Criteria$ recommendation space. We then present the context-aware multi-criteria based recommendation problem.

Users. A given user u may provide ratings for one or more criteria of a target item in a particular contextual situation. We denote by U the set of users $U = \{u_1, .., u_p\}$, where p represents the users number.

Contextual situations. According to the previous presented definitions about contextual situations, we consider the combination of different contextual conditions as a contextual situation. Formally, let Cd be the set of contextual dimensions, represented as $Cd = \{cd_1, .., cd_k\}$, where k is the contextual dimensions total number.

Example. In the tourism field, the set Cd could be composed of these contextual dimensions: Cd={companion, weather, season}.

The contextual dimensions values are represented by a set of contextual conditions. Therefore, the contextual conditions set noted as Cc_i of a particular contextual dimension Cd_i could be represented by $Cc_i = \{cc_{i1}, .., cc_{il}\}$, where l represents the number of the contextual conditions of the particular contextual dimension Cd_i with $1 \leq i \leq k$.

Example. In the set Cd, the first contextual dimension (Cd_1: companion) could be described by the following contextual conditions Cc_1={partner, family, friends}.

Therefore, a contextual situation that we denote by s_j can be formed by the contextual conditions of k contextual dimensions $s_j = \{cc_{1j}, .., cc_{kj}\}$, where $1 \leq j \leq m$ leading to obtain the total set of contextual situations noted by $S = \{s_1, .., s_m\}$.

Example. A particular contextual situation s_1 can be formed by the first contextual conditions of each contextual dimension from the Cd set. Thus, we could obtain the following situation s_1={ partner, sunny, summer}.

Criteria. The criteria set $C = \{c_1, .., c_n\}$ is the one defined in Section 4.3.1.1.

Tripartite Graph. In terms of graph theory, a tripartite graph is the k=3 case of a k-partite graph where the graph vertices are decomposed into three disjoint sets. We model the context-aware multi-criteria network by a weighted tripartite graph represented as G_{SUC}=(S, U, C, E^{SU}, E^{UC}) where S, U, C depict the sets of contextual situations, users and criteria vertices; E^{SU} and E^{UC} stand for the two edges types that indicate the relatedness between users-contextual situations and users-criteria respectively. We project the tripartite graph G_{SUC} into two bipartite graphs: the contextual situation-user bipartite graph defined as G_{SU}=(S, U, E^{SU}) and the user-criteria bipartite graph represented by G_{UC}=(U, C, E^{UC}). More specifically, the edge (s_i, u_j)$\in$ E^{SU} in the first graph G_{SU}, represents the undirected link connecting the contextual situation $s_i \in S$ and the user $u_j \in U$ where the corresponding weight edge is denoted by $w_{ij}^{(su)}$. For the second graph G_{UC}, a relationship can be between a user u_j and a criterion c_o if u_j has provided a rating for c_o. This relationship is modeled by the edge (u_j, c_o) $\in E^{UC}$ and established in one direction, where $w_{jo}^{(uc)}$ indicates its weight. The two defined types of edges representing the two graph entities relationships are weighted as follows:

Contextual situation-user association weighting: to attribute the weight of importance corresponding to each edge (s_i, u_j)$\in$ E^{SU} connecting a user to the contextual situation in which this user is involved in, we employ the weighing scheme TF-IDF (Term Frequency Inverse Document Frequency) [131], which to date has tended to point on weighting bipartite graphs edges rather than tripartite graphs edges. Our choice of using TF-IDF is motivated by its ability to discover latent associations that may exist between users and their contextual situations while consuming items. To apply TF-IDF in our model, a document is viewed as a user and a term is considered as a contextual situation. Thus, we obtain a SF-IUF (Situation Frequency Inverse User Frequency) described as follows:

$w_{ij}^{(su)}$= SF(i,j) $\times IUF(i)$.

$$SF(i,j) = \frac{freq(i,j)}{\max[freq(j)]} \tag{4.6}$$

$freq(i,j)$ represents the frequency of a contextual situation s_i in which the user u_j provide an item rating and $max[freq(j)]$ represents the maximum frequency over the frequencies of all the contextual situations in which the user u_j is involved in.

$$IUF(i) = \frac{|U|}{|U_i|} \tag{4.7}$$

To compute the Inverse User Frequency IUF(i), we divide the total number of users $|U|$ by the number of users in the contextual situation s_i denoted by $|U_i|$.

Afterwards, we build a weighted adjacent matrix A representing the first bipartite graph G_{SU}, where the matrix A elements can be defined as:

$$a_{ij} = \begin{cases} w_{ij}^{(su)} & \text{if } (s_i, u_j) \in E^{SU} \\ 0 & \text{otherwise.} \end{cases}$$

User-criteria association weighting: the edge weight $w_{jo}^{(uc)}$ of the link connecting a user u_j by a criterion c_o is calculated as an average rating $\bar{r}_{jo}$ of u_j for c_o across all the items set denoted by I:

$$w_{jo}^{(uc)} = \bar{r}_{jo} = \sum\nolimits_{t \in I} (\frac{r(j,t,o)}{I}) \tag{4.8}$$

Here, $r(j, t, o)$ represents the rating provided by the user u_j for the criterion c_o of the item t.

The weighted adjacent matrix $B = \{b_{jo}\}$ representing the second bipartite graph G_{UC} can be defined as follows:

$$b_{jo} = \begin{cases} w_{jo}^{(uc)} & \text{if } (u_j, c_o) \in E^{UC} \\ 0 & \text{otherwise.} \end{cases}$$

Example *(Hotel recommendation)*

Consider there are three travellers: Steven, Alice and Paul, who attempt to choose a suitable hotel by taking into account their contextual situations as well as the hotel criteria: service, location and quality of room. Steven and Alice are a couple going on summer vacation. Paul is going on a business trip that will take place from 2-9 November. In this example, we can define the users set $U = \{Steven, Alice, Paul\}$, the contextual situations set $S = \{\{couple\ trip, summer\}, \{business\ trip, winter\}\}$ and the criteria set $C = \{service, location, quality\ of\ room\}$. Figure 4.1 shows the tripartite graph constructed based on this example.

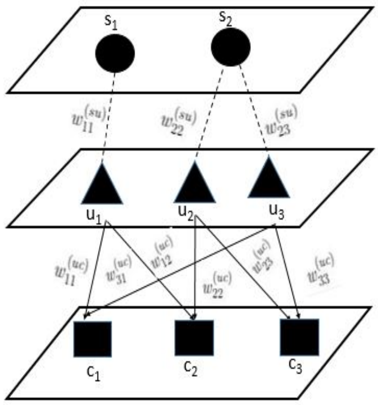


FIG. 4.1.: Example of tripartite graph structure

Context-aware multi-criteria recommendation problem. The outstanding issue we are addressing is estimating users preferences for items by taking into account their contextual situations as well as their feedback on items criteria. In particular, this problem is posed in terms of two intriguing sub-problems:

(1) Determining co-clusters from the tripartite graph G_{SUC} based on the graph edges weights. This first sub-problem handles the tripartite graph partitioning issue that

consists in clustering simultaneously three different types of entities.
(2) Estimating users overall items ratings from predicting the partial users ratings upon criteria by considering the criteria correlations which was often neglected in the previous MCRS.

4.4.1.2 Tripartite graph co-clustering

To get insights about the co-clustering structure, we design the following research hypothesis H_1 that guides our investigation:
H_1: **"Users in similar contextual situations tend to have similar preferences for similar criteria"**.
As outlined by our hypothesis H_1, the concerned triplet data is composed of users U, contextual situations S and criteria C which are the three entity types in the tripartite graph G_{SUC} defined in Section 4.4.1.1. Therefore, we aim to partition G_{SUC} into co-clusters of users in similar contextual situations evaluating similar items criteria. Traditional clustering approaches have failed to accurately provide meaningful co-clusters simultaneously from high-order graphs. In fact, it was revealed that applying an extended spectral co-clustering to a high-order graph might not provide the adequate co-clustering results [132]. To perform the tripartite graph partitioning, we point towards the idea of modeling the high-order co-clustering problem in the form of two pair-wise co-clustering sub-problems with the constraint of the triplet structure. To accomplish this task, the tripartite graph G_{SUC} will be modeled as two dependent bipartite graphs that we denote by G_{SU} and G_{UC} sharing the central entity type U. In this regard, we use the Collective Matrix Factorization (CMF) [133] as a co-clustering algorithm. It is known to be effective in mining multi-relational data compared to other co-clustering algorithms like the spectral relational clustering [134]. In fact, the CMF tackles the problem of jointly clustering graphs by factorizing their corresponding connected matrices describing the relationships between the graphs entities. In our case, we consider the two related bipartite graphs G_{SU} and G_{UC} represented by the corresponding adjacency matrices $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{p\times n}$ respectively, where m represents the contextual situations number, p is the users number and n is the criteria number (Section 4.4.1.1). The data matrices A and B include the three considered graph entities (S, U, C) as well as their weighted links. In the CMF co-clustering process, the building block handles low-rank matrix factorization, which extends factorizing one matrix to factorizing a set of related matrices with shared entities. Consequently, using CMF implies the low-rank decomposition of every adjacency matrix into the product of two matrices to simultaneously identify subgroups of its rows and columns entities. As a result, we begin by jointly factorizing the two input

matrices A and B into three low-dimensional matrices that we denote by T, W, and Z. More specifically, each matrix is approximated by the product of low rank-K factors forming the representation of the associated row and column entities.
The contextual situations-users matrix $A \in \mathbb{R}^{m \times p}$ is represented as the product of lower-rank factor matrices: $m \times k$ matrix T and $p \times k$ matrix W. That is, $A \approx f(TW^T)$ for an element-wise transformation $f\colon \mathbb{R} \to \mathbb{R}$ and $k < \{m, p\}$.
Since the entity type U participates in the both matrices A and B, we use its corresponding factor W in both approximations and thus we have:
The users-criteria matrix $B \in \mathbb{R}^{p \times n}$ is represented as the product of two lower-rank factor matrices: $p \times k$ matrix W and $n \times k$ matrix Z. That is, $B \approx f(WZ^T)$ for an element-wise transformation $f\colon \mathbb{R} \to \mathbb{R}$ and $k < \{p, n\}$.
As a result we obtain a low-rank representation for each entity-type, where $T \in \mathbb{R}^{m \times k}$ contains the contextual situations factors, $W \in \mathbb{R}^{p \times k}$ contains the users factors, $Z \in \mathbb{R}^{n \times k}$ contains the criteria factors and f represents the link function.
We will then use the obtained factors as the losses arguments to compute the closeness between the input matrices (A and B) and their corresponding reconstructions: $A \approx f(TW^T)$ and $B \approx f(WZ^T)$ respectively. Therefore, we mathematically model the co-clustering problem as an optimization problem handling an additive objective function (Equation 4.9) to be minimized. This function is formed by the sum of the two reconstruction losses of the two related input matrices A and B. In fact, we provide a specific objective function for sharing information between A and B matrices by tying their corresponding losses. To optimize the objective function accurately, we rely on constraints involving the stochasticity of rows and non-negativity, i.e., the factors are normalized and non-negative. A further important implication of a regularization term $R(.)$ in the following objective function is useful to help optimizing the collective factorization model.

$$\underset{T,W \geq 0}{\arg\min} f = \alpha D(A, TW^T) + (1 - \alpha) D(B, WZ^T) + R(T, W, Z) \tag{4.9}$$

More precisely, $\alpha \in [0, 1]$ acts as the indicating trade-off parameter that weights the relative importance between two relations, $D(A, TW^T)$ and $D(B, WZ^T)$ represent the loss functions quantifying the cost of the approximations. It is assumed that the loss is decomposable. For example, for the model $A \approx f(TW^T)$, the corresponding loss $D(A, TW^T)$ decomposes over the elements of A into a weighted sum. Accordingly, the loss for weighted singular value decomposition can be written as: $D(A, TW^T) = ||W_e \odot (A - TW^T)||^2$, where W_e denotes the loss argument representing the data weights and $\odot$ represents the element-wise product of matrices.
The added regularization penalty $R(T, W, Z)$ in the objective is employed to mitigate overfitting. There are standard regularizers for linear models that can be adapted, such

as the l_p norms of the factors. We employ in our experiments the robust decomposable l_1 norm based regularization: $R = \gamma_1||T||_1 + \gamma_2||W||_1 + \gamma_3||Z||_1$, where $\gamma_1, \gamma_2, \gamma_3$ represent the regularization control parameters.

Having defined the objective function, the next step consists in solving the optimization problem. Here, the optimization process gives the solution of finding optimal co-clusters of the graphs entities. To accomplish this task, we need to differentiate the objective function with respect to each of the factors T, W and Z. In our case, the loss is represented as a linear function composed of individual losses. So, to determine the roots of the differentiable functions, we derive an efficient Newton update by using stochastic constraints cyclically till convergence. Therefore, the optimisation process applied on (4.9) drives us to identify optimal clustering of data leading to simultaneously partitioning the connected graphs entities into T co-clusters $C_l = \{c_{l_1}, .., c_{l_T}\}$.

4.4.1.3 Rating prediction algorithm

In this work, we mainly focus on enhancing the criteria ratings prediction, since the overall user preference is estimated based on these predicted criteria ratings. Our improved rating prediction process runs in two steps: (1) predicting users criteria ratings involved in similar contextual situations; (2) computing users overall items ratings.

Criteria Ratings Prediction

In this stage, we present an improved rating prediction method to generate, as output, the predicted ratings of criteria included in each co-cluster $c_{l_k}, k \in \{1, .., T\}$. Therefore, we make use of the co-clusters set C_l of contextually similar users with preferred criteria derived from the tripartite graph G_{SUC} co-clustering process (Section 4.4.1.2). More specifically, the object of this stage consists of two parts:

(a) First, we point towards the idea of considering the dependency between users and their contextual situations. For this purpose, the contextual dimensions values forming the contextual situations are fused to users offering a new reduced recommendation space. Consequently, we obtain a lower dimensional matrix that we denote R representing the $Users \times Items \times Criteria$ recommendation space after excluding the contextual situations from the original multi-dimensional matrix M representing the $Users \times Items \times ContextualSituations \times Criteria$ recommendation space. To deal with such problem, we employ the user splitting approach [135], where a user with significantly various preferences in several contexts may be viewed as multiple users.

More details on this will be given in the example below:

TABLE 4.1: A partition of the Educational context-aware multi-criteria rating dataset

User	*Item*	*Rating*	*Application*	*Data*	*Ease*	*Class*	*Year*	*Semester*
st_1	tp_1	4	4	4	4	DA	2018	Spring
st_1	tp_2	2	2	2	2	DM	2018	Fall
st_1	tp_3	4	4	5	4	DM	2017	Spring

Table 4.1 illustrates a partition of the Educational dataset [136] on which we will apply an example of the user splitting process. The Educational dataset will be employed for the experimental investigations in Chapter 6. It can be seen from the Table 4.1 that our example presents one user representing the student (st_1), three different items representing the topics of projects (tp_1, tp_2 and tp_3), three criteria of the topics of projects (application, data and ease) and three contextual situations constituted by the values of three contextual dimensions representing the type of the class (Data Mining (DM) or Data Analytics (DA)), year of the course (2017 or 2018) and semester (Fall or Spring). In the initial step of the user splitting process, we have to determine the contextual condition in which users give significantly different ratings. This task is performed by using the impurity criteria [137] measuring how much the student st_1 has rated the criteria differently across the different contextual conditions. If we assume that the best split for the student st_1 is the semester contextual conditions (Fall vs Spring). Thus, st_1 can be splitted into the two new following students: st_{11} representing st_1 choosing a project topic for the spring semester and st_{12} representing st_1 choosing a project topic for the fall semester. Accordingly, the original matrix shown in Table 4.1 is converted into a reduced one presented in Table 4.2 after eliminating the contexts through applying the user splitting process.

TABLE 4.2: The transformed partition of the Educational context-aware multi-criteria rating dataset

User	*Item*	*Rating*	*Application*	*Data*	*Ease*
st_{11}	tp_1	4	4	4	4
st_{12}	tp_2	2	2	2	2
st_{11}	tp_3	4	4	5	4

(b) Second, we carry out for this part a prediction algorithm for the criteria-related ratings that underlines the interactions existing among criteria. Many earlier recommendation studies ruled out the dependency aspect when generating ratings prediction, not to mention the studies in the field of multi-criteria recommendation. As far as we aware, the approach in [98] is the only one that takes into account the correlation among the multiple criteria to generate ratings prediction for these criteria. However, instead of increasing the performance results, using the dependent-based criteria method in [98] could degrade the performance results in some cases.

We introduce the proposed criteria rating prediction process in Algorithm 2 that uses as input parameters the converted multi-criteria rating matrix R, the co-clusters set C_l including T co-clusters and the number of factors F. Algorithm 2 attempts to generate, as output, the users predicted criteria ratings in each co-cluster $c_{l_k} \in C_l$ by taking into account the interactions between criteria.

Algorithm 2 Criteria Ratings Prediction for each Co-cluster

Input: Converted multi-criteria rating matrix R, the number of co-clusters: T , and the number of factors F.
Output: Criteria predicted ratings in each co-cluster.
Begin
1. **For** each co-cluster $k \in \{1, .., T\}$ **do**
2. R_k=ExtractSub-matrix (R, c_{l_k})
3. P_k,Q_k=MatrixFactorization(R_k,F)
4. **For** each $j \in P_k$ **do**
5. **For** each $t \in Q_k$ **do**
6. **For** each $f \in \{1, .., F\}$ **do**
7. $Corr(c_k, c_E)$=1-$EuclideanDist(c_k, c_E)$
8. $\hat{r}_{j,t,c_k}$= $p_{j,f} \times q_{t,f} \times Corr(c_k, c_E)$

End.

As shown in Algorithm 2, we start by extracting from the converted multi-criteria rating matrix R more personalized sub-matrices depending on the obtained clustering results. Precisely, for each co-cluster k (c_{l_k}) including users with similar contextual situations assessing preferred items criteria, we can obtain from the matrix R a sub-matrix that we denote R_k containing the users and items criteria included in that co-cluster. Thereafter, we turn to apply the Matrix Factorization (MF) [122] to decompose each sub-matrix R_k into the product of two reduced matrices. We call the first low dimensional matrix by P_k, where each row would represent the associations strength between a user and the features. The second low dimensional matrix is denoted as Q_k, where each row would represent the associations strength between an item and the features. To learn the two matrices P_k and Q_k, we use the SGD method minimizing the rating prediction errors. Following this, we look then at the second part which concerns taking into account the correlation aspect in the prediction process. For this purpose, we apply an effective correlation-based rating prediction algorithm [44]. The presented approach highlights the "contextual correlation" which concerns the correlation between contexts by measuring the similarity between the contextual situations. The main idea behind the concept of "contextual correlation" is that more the contextual situations are correlated or similar, the more the recommendation lists in these situations are similar. Following our contribution, we adapt the correlation function $Corr(c_k, c_E)$ to be used for estimating the correlation between two criteria: the current criterion c_k corresponding to the user j and the item t and an another criterion c_E. We assume in our model that the criteria could constitute a multidimensional coordinate system, where, each criterion can locate a position in the corresponding axis. As a result, the correlation between two criteria can be measured as the distance between their corresponding points included in the multidimensional coordinate system. For this task, we opt for the Euclidean distance to compute the distances between the points representing the criteria (line 7). Afterwards, we apply the correlation-based prediction formula (line 8):

$$\hat{r}_{j,t,c_k} = \vec{p_j}.\vec{q_t}.Corr(c_k, c_E) \tag{4.10}$$

Where $\vec{p_j}$ is the user vector, and $\vec{q_t}$ represents the item vector. We learn the parameters including the user and item vectors as well the criteria positions by SGD method by minimizing the rating prediction errors.

Overall Rating Prediction

After the multi-criteria ratings prediction step, we need to aggregate these ratings into a single output representing the overall item rating. Thus, the overall item rating is not just another separate rating, but rather serves as a multi-criteria ratings aggregation of this item. Despite the fact that predicting criteria ratings is beneficial in several situations, but it is not sufficient, as one of the primary targets of recommendation systems is to predict the overall item rating for each user. Besides, to make pertinent items recommendations, MCRS eventually need to compare the items in terms of their corresponding overall ratings. Indeed, if the suitable items are selected in the absence of the overall items ratings, MCRS could be confronted to complicated optimization issues [138]. Thus, defining the aggregation function is significant for multi-criteria recommendation. To predict how much a user will prefer an item, we present the aggregation function F_r (Equation 4.11) that represents the relation between the criteria ratings $(r_1, r_2, ..., r_n)$ and the overall item rating r_0.

$$r_0 = F_r(r_1, r_2, ..., r_n) \tag{4.11}$$

In the field of MCRS, several aggregation functions can be found. The most common ones could not be convenient such as the average function since the optimal weights of the item criteria are neglected. In our case, we assume that there is a linear relationship between the multiple criteria and the item overall rating. Therefore, we opt for adopting the linear aggregation [86] to generate the function F_r as follows:

$$r_0 = w_1 * r_1 + w_2 * r_2 + ... + w_n * r_n + c \tag{4.12}$$

One promising technique in our case would be the multiple linear regression-based technique. More precisely, we employ the linear-aggregation based multi-criteria recommendation method Support Vector Regression (SVR) [139] to find the overall assessment of an item r_0, where w_n is the weight of importance associated to the criterion c_n. This choice is motivated by the ability of SVR in enhancing the predictive accuracy of the aggregation-based approaches and handling the sparse datasets. This is typical, for example, in the tourism area.

We learn the linear aggregation parameters (viewed in Equation 4.12) including the weights of criteria $(w_1, w_2, ..., w_n)$ and the constant c by minimizing the squared prediction errors during the training.

4.5 Conclusion

The researches on recommendations systems show performance improvement in the quality of recommendations, when making use of the relevant available information as input data. In fact, it is revealed that there is a compelling need to explore the extensions from the aspect of the input data type of recommenders, where the significant inputs are users explicit feedbacks such as ratings on items criteria, and the context information when users are selecting an item.

From this respect, we provided in this chapter various contributions starting from modeling the relevant multi-dimensional input data up to generating items ratings prediction. To achieve our goals, we propose a set of techniques that take advantage of the positive effect of incorporating both context-awareness and multi-criteria decision making directions into the recommendation process.

The next chapter is dedicated to the experimental evaluation of our context-based proposals.

Chapter 5

Evaluation of the Context-based Recommendation Approach

5.1 Introduction

The recommender system is a complex system. Most of the effort made when developing our work was experimenting novel solutions to upgrade the system performance results in the rating prediction and recommendation tasks. We will present in this chapter, the experiments conducted to evaluate the effectiveness of the context-aware recommendation proposals.

5.2 Experimental Evaluation Setting

In this section, we start by describing the used datasets. Then, we introduce the evaluation protocol. Finally, we present the comparative baseline algorithms and the evaluation metrics.

5.2.1 Datasets

A dataset is a major component consists of a collection of data table objects related to each other to support the research evaluation. In the world of recommender systems, it is a common practice to use public available datasets from different application environments in order to evaluate and compare the performance of recommendation algorithms. We have proposed in this dissertation context-aware recommendation models. Therefore, their evaluation performance should be on context-aware recommendation datasets. We conduct our experiments on four popular contextual real-world datasets from various domains: music, food and movie. This variation enables us to assess the performance of the proposed models across a range of different datasets, each with different characteristics. We provide in the following more details about these datasets.

- **Music dataset** [58] is collected from a mobile application recommending music tracks to the passengers involved in various driving and traffic conditions. The dataset contains 8 contextual dimensions and 34 contextual conditions in total.
- **Food dataset** [140] represents a contextual food preference dataset collected from a survey containing users ratings on the food menu in the context of different degrees of hunger.
- **Movie dataset** [141] is collected from surveys in which students were asked to give ratings for movies in various contexts. In this dataset, three contextual dimensions were captured: location (home, cinema), time (weekday, weekend) and companion (alone, partner, family).
- **LDOS-CoMoDa dataset** [142] is a movie-rating dataset collected from a web application that enables rating movies in different contextual situations. It contains various variables among which a set of different contextual conditions coming from 12 various contextual dimensions describing the situations in which the movies were watched.

The properties of these datasets are summarized in Table 5.1 (first four rows).

TABLE 5.1: Description of the used datasets

Dataset	**# of users**	**# of items**	**# of ratings**	**# of contextual dimensions**	**# of contextual conditions**
Music	41	139	3940	8	34
Food	212	20	6360	6	8
Movie	97	79	5035	3	12
LDOS-CoMoDa	185	4138	2297	12	49
Contextual MovieLens	**2648**	**2758**	**2758**	**12**	**49**
Contextual Movie & TV	**9660**	**1196**	**1196**	**12**	**49**

5.2.1.1 Large contextual datasets: the enrichment methodology

Despite the fact that datasets play a fundamental role in performance comparison and evaluation, the number of the available contextual datasets is quite limited and those datasets that incorporate context are usually small or sparse. In fact, a challenging problem which arises in context-aware recommendation field is the relative rarity of

large contextual datasets. Part of the reason is the difficulty of collecting contextual data explicitly unlike users choices that can be collected more simply. Specifically, in CARS research area, the majority of publicly available datasets are recorded from surveys, leading to obtain small and sparse datasets. Furthermore, collecting ratings in multiple contexts is a hard task and the privacy of each user is often a concern. That's why, in literature, we can only find small contextual datasets with a reduced number of users or large no-contextual datasets. For instance, we can notice from Table 5.1, that the number of users in the majority of the publicly available contextual datasets is under 250. This small number can lead to serious problems for most collaborative filtering systems like cold-start problem [143–145], scalability [146] and sparsity [147].
It is inherently difficult to produce relevant contextual recommendations, as the few users can not rate all the items of the overall dataset in the various contexts and therefore it is difficult to build rich user profiles.
That is why finding valuable contextual datasets is considered as a hard task. Therefore, in seeking to help improving contextual recommendations and to alleviate the faced problems such as the cold-start users problem in small and sparse datasets, we point towards the idea of building large-scale contextual datasets to gain sufficient users contextual ratings to be considered by the rating prediction algorithms.
Our idea is motivated by the recommendation accuracy obtained by collaborative filtering approaches in large-scale recommendation engines. For instance, in [148], the cold start problem has been effectively solved on a large scale while maintaining a high level of reliability of the recommendations.
Our work aims to extend existing large recommenation datasets by including contextual information to describe users expressed preferences linked to their corresponding contexts. More specifically, the original no-contextual datasets will undergone changes in their content. These changes necessarily impact the ratings prediction accuracy, since users ratings are influenced by the current context in which these users are involved.
The dataset enrichment process operates through the following three steps sketched by Algorithm 3:

- Extracting categories.
- Similarity between categories.
- Creating large contextual datasets.

1. **Extracting categories:**
   In the initial step, we select a contextual dataset with different contextual dimensions and a large no-contextual dataset within the same domain.

Algorithm 3 The dataset enrichment process

Input:
CD: a contextual dataset
LND: a large non-contextual dataset
t: a similarity threshold
Output:
LCD : a large contextual dataset
Begin
1. C_{CD}= $ExtractCategories$ (CD)
2. C_{LND}= $ExtractCategories$ (LND)
3. s= $Similarity$ (C_{CD},C_{LND})
4. If (s $\geq t$) $Then$
5. LCD=AddContext(CD,LND)
6. $End\ If$
7. $Return$ LCD
End.

Therefore, these datasets share the same item type. After selecting the convenient datasets to be used for the enrichment process, we need to retrieve the items categories of each dataset. In line 1, the algorithm 3 calls the *ExtractCategories* function. This routine returns the items categories of a contextual dataset by different ways according to the given dataset. In fact, we could find that besides users feedback on items, some datasets may comprise other side information such as items category information. In this case, the items categories could be directly extracted form the dataset. In other datasets, users reviews are given in the form of a natural language text along with data signals that could help in finding the items categories. Thus, for this latter kind of datasets, the *ExtractCategories* function explores the reviews content to extract items categories. For this task, we first identify the keywords by removing all the existing stop words in the review text. Then, a request is submitted with the obtained keywords to the Open Directory Project (ODP), also known as dmoz[1] which is an open directory that lists categories based on search terms. Finally, a list of categories matching the keywords is obtained by ODP. In line 2 of Algorithm 3, the same described extracting process is applied for a large non-contextual dataset.

2. **Similarity between categories:**

Once we have extracted the items categories from the considered datasets, turning now to compute the similarity between the contextual dataset items categories and the no-contextual dataset items categories (line 3 of Algorithm 3) by using the library WordNet Similarity for Java (WS4J) [2]. The word similarity

[1]http://www.dmoz.org/search/q?
[2]http://wn-similarity.sourceforge.net

library used in this paper supports several similarity algorithms. Among theses algorithms, we choose Lin similarity since it was considered as the more recent measure computing the words pairs similarities by using an information content scheme [149]. It also has the advantage of providing an easier implementation with less complexity. Specifically, the principle of Lin similarity is the more common information two concepts share, the more similar these concepts are, where its main benefit is taking the information content of the compared concepts into considerate. Besides, it has been proven that Lin similarity is able to show significant results over other similarity measures when measuring accuracy and also when correlating the similarity scores with that of human judgments [150].

3. **Creating large contextual datasets:**

   Depending on the obtained similarity results between the items categories of a contextual dataset and a no-contextual dataset, we define a threshold to identify only the most likely similar categories. If the obtained similarity value is greater than the fixed threshold, the Algorithm 3 invokes *AddContext* function (line 5). This function takes from the contextual dataset the context in which the current item corresponding to the selected category is consumed. Then, it adds this context to the current item that corresponds to the selected category of the no-contextual dataset. We obtain finally a novel large contextual dataset from a large no-contextual dataset and a contextual dataset.

Application Example

After describing the enrichment process, the following example outlines an application on real-world publicly available datasets. For the contextual dataset, we choose from Table 5.1 the LDOS-CoMoDa dataset which is a well-known rich contextual dataset with multiple contextual dimensions and conditions. Also, we use two popular large no-contextual datasets :

MovieLens-latest dataset[3] defines tagging activity from MovieLens, as a movie recommendation service with tag applications across movies.

Movie & TV dataset[4] contains items metadata (price, category information, brand, and image features) and reviews from Amazon, including item and user information as well as their ratings.

The main reason for this choice is that these three considered datasets (Movie & TV, MovieLens-latest and LDOS-CoMoDa) represent movie rating datasets with the same shared item type "Movie". LDOS-CoMoDa and MovieLens-latest datasets specify a list of categories for each movie ID. While in Movie & TV dataset, users provide some

[3] https://grouplens.org/datasets/movielens/latest/

[4] https://snap.stanford.edu/data/web-Amazon.html

cues in their reviews by expressing their opinions about the movie category.
Our main objective is to combine each large no-contextual dataset with LDOS-CoMoDa dataset to make two novel large contextual datasets which we call (Contextual MovieLens and Contextual Movie & TV). The characteristics of these datasets are summarized in the last two rows of Table 5.1.

TABLE 5.2: Example of Movie & TV dataset enrichment

The enrichment process steps	**Example**
1. Extracting categories	MovieId : 2 ReviewText: "eat movie all kids. It is entertaining, nicely done and keeps the kids entertained." keywords: movie, kids, entertaining, nicely, done, keeps, kids, entertained. Categories: Kids and teens, dance and art.
2. Similarity between categories	Similarity(Kids and teens, Comedy)= 0.0806 Similarity(Kids and teens, Music)= 0.0944 Similarity(Dance and art, Comdey)= 0.0872 Similarity(Dance and art, Music)= 0.0908
3. Creating large contextual dataset	Threshold (t)= 0.08 Movie & TV dataset+context: MovieId : 2 Categories: kids and teens, dance and art, comedy and music Context: night; weekend; summer; home; clear and parents.

TABLE 5.3: Example of MovieLens dataset enrichment

The enrichment process steps	**Example**
1. Extracting categories	MovieId : 1 Categories: adventure; animation and action.
2. Similarity between categories	Similarity(Action, Adventure)= 0.090 Similarity(Action, Animation)= 0.097 Similarity(Action, Action)= 1
3. Creating large contextual dataset	Threshold (t)= 0.09 MovieLens+context: MovieId : 1 Categories: adventure, animation and action. Context: evening; working day; autumn; home; cloudy and alone.

More specifically, Table 5.2 shows an example of Movie & TV dataset enrichment by considering from the contextual dataset (LDOS-CoMoDa) the MovieId : 2, its categories are comedy and music under the following contextual conditions: night; weekend; summer; home; clear and parents.
Table 5.3 illustrates an example to establish the contextual Movielens dataset by applying the enrichment process through taking from the contextual dataset (LDOS-CoMoDa) the MovieId : 1, belonging to the action category in the contextual conditions: evening; working day; autumn; home; cloudy and alone.

5.2.2 Evaluation Protocol

We present in the evaluation protocol how the datasets are handled to obtain the required training and testing sets.
It consists in dividing each entire dataset into k equally sized folds, to perform a cross-validation evaluation process. The value of the parameter k shouldn't be too small or too high, it is often chosen to be 5 to 10 depending on the data size. The partitioning process is repeated K times, each time changing the fold used as testing set. On the above datasets, we conduct a 5-fold cross validation. Here, the objective of training is to learn the fuzzy measures viewed as the importance of each contextual dimension and each subset of dimensions and to predict unknown ratings, while, the testing set is used to assess the accuracy of the predictions. More precisely, each dataset has been divided into five folds. Four folds of data are viewed as the training set (TR.S) and the remaining fold is used as the testing set (TE.S).

5.2.3 Baselines

In this section, the recommendation algorithms that we use for comparison are summed up, where the chosen algorithms are the renowned traditional and contextual ones implemented in the java based context-aware recommendation engine CARSKit [141]. The compared algorithms are described below:

1. **User-oriented K-Nearest Neighbors (UserKNN)** [151] represents a neighborhood collaborative filtering algorithm on the basis of users similarity.
2. **Item-oriented K-Nearest Neighbors (ItemKNN)** [151] represents a neighborhood collaborative filtering algorithm on the basis of items similarity.
3. **Differential Context Weighting (DCW)** [114] introduces the contextual weighting in the rating prediction process through a weighted similarity measure.

4. **Singular Value Decomposition model based on implicit feedback (SVD++)** [152] represents a matrix factorization model using users history information.

5. **List-Rank Matrix Factorization (LRMF)** [153] refers to a matrix factorization ranking model that joins the list-wise learning with MF.

6. **Context-Aware Matrix Factorization (CAMF)** [58] represents an extended MF model that integrates contextual information in the rating prediction process. We tried its three variants (CAMF-C, CAMF-CI and CAMF-CU) and we only present the best performing one, denoted by CAMF-Dev.

7. **Multidimensional Context Similarity (CAMF-MCS) model** [44] refers to a CAMF algorithm considering the contextual correlation aspect using a multidimensional space.

5.2.4 Evaluation Metrics

The recommender system's output can be viewed as a recommendation items list for each user produced through ranking items in accordance with their corresponding predicted ratings. Thus, the obtained ratings from the rating prediction process are used to offer effective recommendations of items corresponding to the current needs of each user. Therefore, the evaluation will concern two tasks: the rating prediction task and the top-N recommendation task. There are various suggested metrics for evaluating recommender systems. According to the tasks we are evaluating, we consider different metrics grouped into two main classes: rating metrics for evaluating the rating prediction task and ranking metrics for evaluating the top-N recommendation task.
We also tried to measure some alternative performance metrics described in 1.4.2.3.

5.2.4.1 Rating metrics

To assess the rating prediction accuracy, we choose the most widely used metric, Mean Absolute Error (MAE) [154] defined in 1.4.2.1. This metric is useful when the evaluated method is based on ratings prediction, since it can estimate how accurate the predicted ratings are, and in turn it gives an idea about the recommendations accuracy.

5.2.4.2 Ranking metrics

Evaluating the top-N recommendations is another way of evaluating the quality recommendations, because a high-accuracy in rating prediction can not necessary translate to improvements in user recommendations [155]. The most common way to assess the quality of the top-N recommendation is to measure whether pertinent items are in the top positions of a recommendation items list. Therefore, the items ranking offers more results about the performance of the recommender. For this purpose, we compute the Recall@N, Precision@N and NDCG@N metrics defined in 1.4.2.2.

5.3 Offline Experiments

A number of parameters explained when the algorithms were introduced, have an influence on the results produced by the recommender system. Therefore, before conducting the experimental evaluation for our proposals, we begin by performing preliminary experiments by presenting a parameter sensitivity analysis in order to set the optimum values of these parameters to be used for the further evaluation experiments. Moreover, we perform an accuracy analysis by varying the contextual dimensions number to select the relevant ones according to their corresponding fuzzy measures. Then, we introduce in the second part the evaluation effectiveness of our proposals and carry out the comparison with popular recommendation baselines.

5.3.1 Analyzing parameter sensitivity and relevant contextual dimensions importance

5.3.1.1 Impact of the number of iterations:

First, we examine the number of iterations required in our proposals which are the Fuzzy Weighting Recommender approach (FWR) and the two proposed CAMF-MCS strategies: the weighting strategy (WCAMF-MCS) and the interaction strategy (ICAMF-MCS). Figure. 5.5 reports for each dataset the prediction accuracy measured in compliance with the number of iterations. It is apparent from Figure. 5.5, that on Music and Food datasets, FWR requires 20 iterations to get a peak prediction accuracy. While WCAMF-MCS and ICAMF-MCS show that a good prediction is obtained in the fortieth iteration but more iterations will adversely hurt the prediction accuracy. When it comes to the Movie dataset, both methods indicate reduced prediction accuracy when the iterations number goes beyond 60. For the LDOS-CoMoDa dataset, the best performance is achieved by FWR at 50 iterations. For WCAMF-MCS and ICAMF-MCS, we can note that the prediction accuracy progresses when the iterations number reaches 100. We set the suitable iterations number for each method when the best prediction accuracy is achieved.

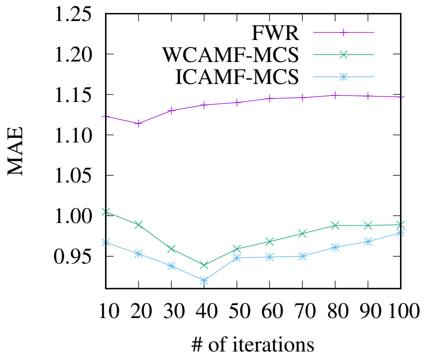


FIG. 5.1.: Food dataset

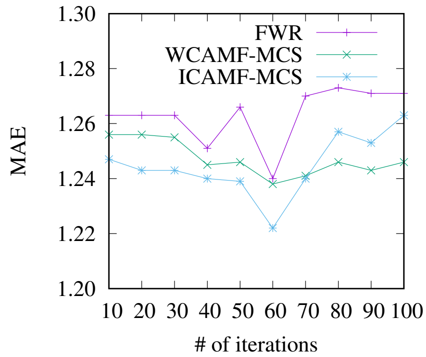


FIG. 5.2.: Movie dataset

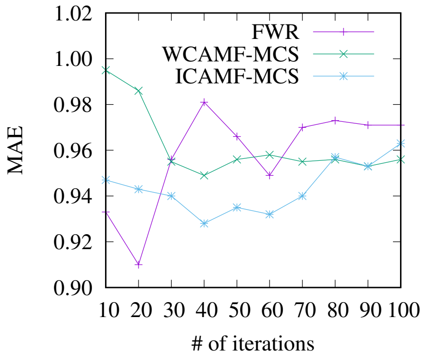


FIG. 5.3.: Music dataset

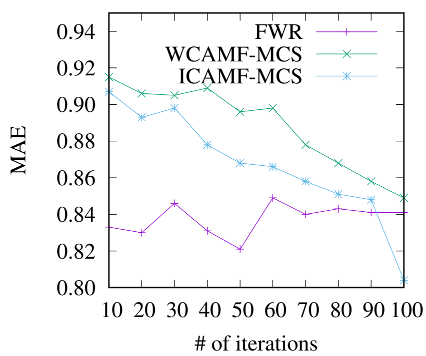


FIG. 5.4.: LDOS-CoMoDa dataset

FIG. 5.5.: MAE variation in different iterations numbers

5.3.1.2 Impact of the number of latent factors:

Selecting a proper number of latent factors is one of the paramount parameters in matrix factorization approaches. Therefore, we investigate the impact of various values of latent factors K on the accuracy of our proposed CAMF-based strategies.

The Figure. 5.10 pinpoints that on LDOS-CoMoDa dataset, the MAE of WCAMF-MCS and ICAMF-MCS strategies declines to the lowest when the factors number reaches 30. Consequently, for LDOS-CoMoDa dataset, setting k as 30 is considered as the best value for both WCAMF-MCS and ICAMF-MCS strategies. For the Movie dataset, both strategies achieve good prediction accuracy when K = 10. While on the Music dataset, good MAE results can be obtained when K = 12. Thus, the fixed value of k for both strategies is 12 on the Music dataset and 10 on Movie dataset. Lastly, to obtain the best prediction accuracy on Food dataset, we observe that WCAMF-MCS and ICAMF-MCS need 6 and 12 latent factors respectively. As a result, the required number of latent factors is set to be 6 for WCAMF-MCS strategy and 12 for ICAMF-MCS strategy on Food dataset.

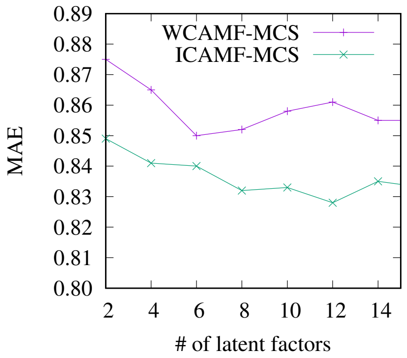


FIG. 5.6.: Food dataset

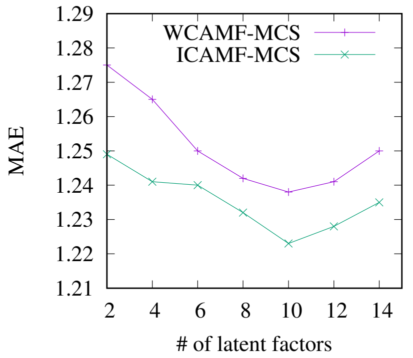


FIG. 5.7.: Movie dataset

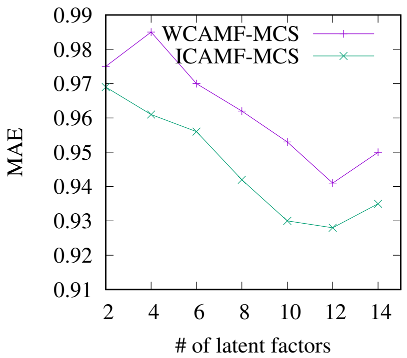


FIG. 5.8.: Music dataset

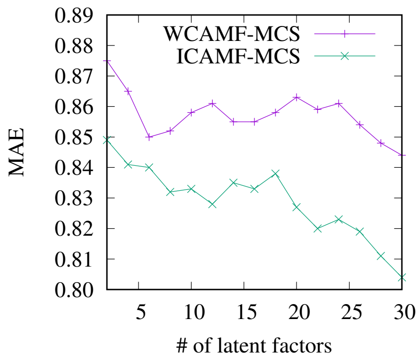


FIG. 5.9.: LDOS-CoMoDa dataset

FIG. 5.10.: MAE variation in different latent factors values

5.3.1.3 Impact of neighbors size:

In collaborative filtering recommendations approaches based on neighborhood methods, the right number of neighbors has to be chosen when computing ratings prediction since it has a major effect on the prediction quality. Thus, we focus on studying the variation of the accuracy of our neighborhood based method FWR with respect to various numbers of neighbors. Figure. 5.11 illustrates the experimental results on four datasets when the neighbors size increases from 10 to 150. It can be seen from Figure. 5.11, that when the neighbors size increases from 10 to 60 the MAE values slightly decrease. It seems that the prediction accuracy gets better when the neighbors number increases since the similar users might help to provide more convenient information for prediction. Nevertheless, we note that the performance of FWR tends to be steady when the neighbors number continues to raise, since many less similar neighbors could be introduced, which causes a downside impact on the predictions. We thus set as the neighbors size, the value 30 for the Music dataset, 60 for the Movie dataset, 40 for the Food dataset and 20 for the LDOS-CoMoDa dataset.

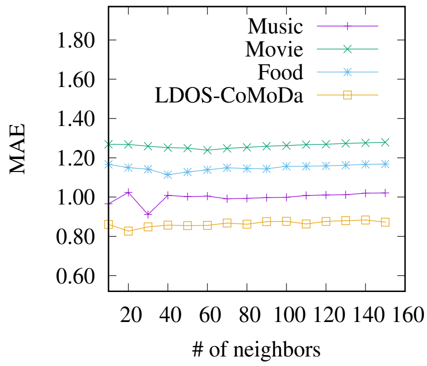


FIG. 5.11.: The impact of neighbors size

5.3.1.4 Impact of relevant contextual dimensions:

Even though we have associated weights for the contextual dimensions through defining their corresponding fuzzy measures, it is also interesting to select the most relevant ones particularly for a rich contextual dataset. Besides, the computational cost is related to the contextual dimensions number in the dataset, where more the contextual dimensions are, more weights must be learned. Therefore, performing a pre-selection process for contextual dimensions would have an important implication for solving this issue. For this task, we make use of the fuzzy measures attributed to each contextual dimension to pick out the most influential ones. The greater the fuzzy measure value of a contextual dimension, the more that dimension tends to be pertinent. More specifically, if the fuzzy measure value of a contextual dimension is better than a given threshold, then that dimension is viewed as pertinent. To get the convenient threshold, we study the effect of the contextual dimensions number on the rating prediction accuracy. Figure. 5.12 shows a contextual dimensions relevancy study on LDOS-CoMoDa dataset that includes the greatest number of contextual dimensions and conditions. The figure is revealing that the two proposed neighborhood-based (FWR) and matrix factorization-based (WCMAF-MCS ICMAF-MCS) methods can reach the best MAE value when the pertinent contextual dimensions corresponding to the threshold are the first four. A greater number of contextual dimensions exceeding 4 may constrict the performance improvement. The four contextual dimensions that are selected as relevant in LDOS-CoMoDa dataset depending on their fuzzy measures values are: Social, Mood, Day-type and Location. Only the detected relevant contextual dimensions will be considered for further experiments.

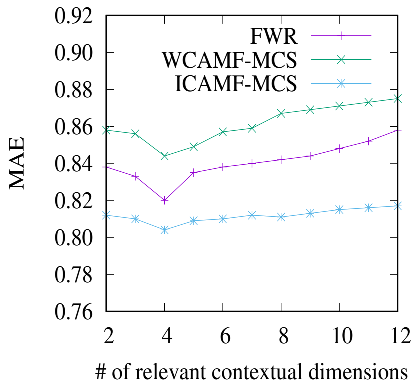


FIG. 5.12.: The impact of relevant contextual dimensions on LDOS-CoMoDa dataset

5.3.2 Results and Discussion

After determining the optimal parameter values, we examine through empirical experiments whether our proposals can exhibit substantial improvements over state-of-the-art recommendation approaches in the rating prediction and top-N recommendation tasks. We start by tuning the fuzzy measures values of contextual dimensions then we conduct two experiments. In the first one, we evaluate the performance of our proposals on the publicly real-world available contextual datasets. In the second experiment, the evaluation performance is conducted on the large created contextual datasets.

5.3.2.1 Tuning the fuzzy measures

TABLE 5.4: An example of the selected fuzzy measures values during the learning phase on LDOS-CoMoDa dataset

	μ_M	μ_D	μ_S	μ_L	$\mu_{\{S,M\}}$	$\mu_{\{S,D\}}$	$\mu_{\{S,L\}}$	$\mu_{\{M,D\}}$	$\mu_{\{M,L\}}$	$\mu_{\{D,L\}}$	$\mu_{\{S,M,D\}}$	$\mu_{\{S,M,L\}}$	$\mu_{\{S,D,L\}}$	$\mu_{\{M,D,L\}}$	**MAE**
μ^1	0.1	0.1	0.1	0.7	0.2	0.2	0.8	0.2	0.8	0.8	0.3	0.9	0.9	0.9	0.924
μ^2	0.4	0.3	0.2	0.1	0.6	0.5	0.3	0.7	0.5	0.4	0.9	0.7	0.9	0.8	0.922
μ^3	0.1	0.7	0.1	0.1	0.2	0.8	0.2	0.8	0.2	0.8	0.9	0.3	0.9	0.9	0.918
μ^4	0.1	0.1	0.7	0.1	0.8	0.8	0.2	0.2	0.2	0.2	0.9	0.9	0.9	0.3	0.889
μ^5	0.7	0.1	0.1	0.1	0.8	0.2	0.2	0.8	0.8	0.2	0.9	0.9	0.3	0.9	0.911
μ^6	0.2	0.3	0.3	0.2	0.5	0.6	0.5	0.5	0.4	0.5	0.8	0.7	0.8	0.7	0.878
$\mu^{(*)}$	**0.1**	**0.4**	**0.4**	**0.1**	**0.5**	**0.8**	**0.5**	**0.5**	**0.2**	**0.2**	**0.9**	**0.6**	**0.9**	**0.6**	**0.845**
$\mu^{(**)}$	**0.17**	**0.25**	**0.47**	**0.12**	**0.23**	**0.10**	**0.22**	**-0.15**	**-0.11**	**-0.06**	**0.65**	**0.52**	**0.55**	**0.68**	**0.825**

The purpose of this step is to generate and set the fuzzy measures values that should be assigned to each individual contextual dimension and each subset of dimensions. In the following, we introduce an example that determine the fuzzy measures corresponding to the relevant contextual dimensions in "LDOS-CoMoDa" dataset. For this task, we have tested 63 weight combinations of contextual dimensions fuzzy measures. In these combinations, the initial weight of each contextual dimension is obtained with a pitch of 0.1 such that the sum of all contextual dimensions weights is equal to 1. The weight of each subset of contextual dimensions is computed as the the sum of the individual contextual dimensions contained in that subset. In the Table 5.4, we present in the first column 6 weight combinations that we denote by μ^i, where M, D, S and L refer to the Mood, Day-type, Social and Location contextual dimensions, respectively. $\mu^{(*)}$ stands for the best obtained combination, which reaches the best MAE value in the learning phase. To obtain the optimal weight combination that contains the final weights attributed to the relevant contextual dimensions, we apply the least squares based optimization method. This combination is denoted by $\mu^{(**)}$ and it is situated in the last row of Table 5.4. More precisely, the least squares method based on quadratic programming uses the best combination returned by the training step $\mu^{(*)}$ to provide an optimal solution represented by the combination $\mu^{(**)}$. We thus obtain the final weights of each contextual dimension and of each subset of dimensions through $\mu^{(**)}$. We note from Table 5.4, that the highest weight assigned to the individual contextual dimensions corresponds to the Social dimension which represents user's companion (friends, partner, colleagues, etc.). This result is not surprising since a user often makes different choices when selecting movies in case he intends to watch the movie with a girlfriend, or with children. The proposed weighting method responded well to this preference by attributing a high weight to the Social contextual dimension in recommending movies. It also can be seen that the fuzzy measures corresponding to the subsets {S,M}, {S,D} and {S,L} are represented by positives values which means that there is a positive interaction between the contextual dimensions included in each of these subsets. For the subsets containing three contextual dimensions, it is apparent from Table 5.4 that their obtained values are positive and important offering a good interaction between them.

To provide a deep understanding of the interaction phenomenon that may exist between the contextual dimensions, we compute the interaction index [109] based on the obtained fuzzy measures.

Formally, let $D = \{dim_1, .., dim_n\}$ be a set of contextual dimensions and $d \subset D$ is a subset of dimensions. For a given contextual dimension dim_i, its interaction with a contextual dimension dim_j is denoted by $I_{dim_i dim_j}$. We thus define the interaction index between two contextual dimensions as follows:

$$I_{dim_i dim_j} = \sum_{d \subset D \setminus \{dim_i, dim_j\}} \frac{(n-|d|-2)!|d|!}{(n-1)!} [\mu(d \cup \{dim_i, dim_j\}) - \mu(d \cup \{dim_i\}) - \mu(d \cup \{dim_j\}) + \mu(d)] \tag{5.1}$$

TABLE 5.5: Interaction indices of contextual dimensions on LDOS-CoMoDa dataset

Interaction index	$I_{\{S,M\}}$	$I_{\{S,D\}}$	$I_{\{S,L\}}$	$I_{\{D,M\}}$	$I_{\{L,M\}}$	$I_{\{D,L\}}$
Value	0.12	0.15	0.07	-0.5	-0.06	-0.02

Table 5.5 reports the computed interaction indices between the contextual dimensions (Social (S), Mood (M), Day-type (D) and Location (L)). The obtained interaction value of each subset within the interval [-1..1], is zero when the contextual dimensions in that subset are independent and it is positive (respectively negative) whenever the interaction between these dimensions is positive (respectively negative). We can note that Table 5.5 is revealing the presence of positive interactions among the contextual dimensions included in these subsets {S,M}, {S,D} and {S,L}. This explains the strong contribution of the contextual dimensions contained each subset when they are present together. These results are consistent with those obtained by least squares method. From the Table 5.5, we can also note that the highest interaction index value is achieved by Social and Day-type dimensions which seem significantly correlated. They can be considered as two complementary contextual dimensions impacting the user's decision of watching a movie. For instance, if we consider that a user often watches movies in weekend with partner, thus, the contextual dimensions Social (partners) and Day-type (weekend) may be significantly correlated as a result. We can also remark that even the Day-type dimension has an important individual weight, it does not bring any contribution when it is combined with Mood or Location dimensions. Only the selected correlated subsets of contextual dimensions are worth considering when producing rating predictions and recommendations by our proposals.

An application example of Music Recommendation

Let consider four contextual dimensions: Weather (We), Mood (M), Time (T) and Activity (A). We begin by determining the contextual dimensions weights using the proposed weighting method based on fuzzy measures. We obtain in a first step the best combination that includes the following fuzzy measures values: μ_{We} = 0.1, μ_M = 0.4, μ_T = 0.4, μ_A =0.1, $\mu_{\{We,M\}}$ = 0.5, $\mu_{\{We,T\}}$ =0.5, $\mu_{\{We,A\}}$ =0.2, $\mu_{\{M,T\}}$ =0.8, $\mu_{\{M,A\}}$ =0.5, $\mu_{\{T,A\}}$ =0.5, $\mu_{\{We,M,T\}}$ =0.9, $\mu_{\{We,M,A\}}$ =0.6, $\mu_{\{We,T,A\}}$ =0.6, $\mu_{\{M,T,A\}}$ =0.9. In a second step, we obtain the optimal combination of final weights that should be assigned to the contextual dimensions and subsets of dimensions. This latter combination contains the following fuzzy measures values: μ_{We} = 0.12, μ_M = 0.36, μ_T = 0.27, μ_A =0.26,

$\mu_{\{We,M\}}$ = 0.10, $\mu_{\{We,T\}}$ =0.09, $\mu_{\{We,A\}}$ =0.12, $\mu_{\{M,T\}}$ = -0.10, $\mu_{\{M,A\}}$ =-0.09, $\mu_{\{T,A\}}$ = -0.02, $\mu_{\{We,M,T\}}$ = 0.78, $\mu_{\{We,M,A\}}$ =0.80, $\mu_{\{We,T,A\}}$ =0.75, $\mu_{\{M,T,A\}}$ = 0.78. It can be seen that Mood dimension has the highest weight among individual contextual dimensions weights. Hence, the Mood dimension is deemed as an influential contextual dimension in the music domain. This is not surprising because a user often seeks for music that match his/her current mood. For instance, after a long tiring working day, Jack certainly prefers to listen to soft smooth relaxing music for relieving everyday stress rather than hard music. Although, Mood has an important weight, we can notice that the obtained weights of the subsets {M,T} and {M,A} are negative. This means that the Mood dimension does not appear to be a good dimension when combined with Time or Activity. However, the fuzzy measures attributed to {We,M}, {We,T} and {We,A} indicate a positive interaction between both dimensions of each subset. We conclude that our approach gives more importance to these correlated dimensions {We,M}, {We,T} and {We,A}. These results are consistent with those obtained from the correlation analysis through the computation of the interaction index in Table 5.6.

TABLE 5.6: Interaction indices of contextual dimensions

Interaction index	$I_{\{We,M\}}$	$I_{\{We,T\}}$	$I_{\{We,A\}}$	$I_{\{T,M\}}$	$I_{\{A,M\}}$	$I_{\{T,A\}}$
Value	0.05	0.07	0.08	-0.07	-0.04	-0.4

The correlated contextual dimensions are worth considering for generating contextual situations. For this task, we employ the open source Java library *jFuzzyLogic* [156] that uses the correlated contextual dimensions in the IF-THEN rules to recognize the common contextual situations in which users listen to music:

RULE 1 : IF weather IS sunny AND time IS morning THEN situation IS commuting;
RULE 2 : IF weather IS sunny AND mood IS active THEN situation IS studying;
RULE 3 : IF activity IS sport AND weather IS cloudy THEN situation IS exercising;

In this stage, we aim to recommend songs that match the inferred contextual situations. Hence, we use the online music streaming service Musicovery [5] which can provide dynamically personalized playlists and their metadata based on the listener's current contextual situation. The returned songs from Musicovery are novel, popular and of different genres matching the listener's current contextual situation which help improving important aspects of recommendation quality, such as the novelty and diversity of recommendations. For example, if we request music tracks where the listener's contextual situation is commuting we obtain the recommendations in the table below.

The obtained songs can be filtered according to their levels of popularity (popularity beyond 50).

[5]http://musicovery.com

TABLE 5.7: Example of music tracks for commuting contextual situation

Listener situation	**Track ID**	**Track title**	**Track genre**	**Track popularity**	**Artist**
Commuting	91743	Love Is A Cigarette	rock	35	Nuno Bettencourt
	7512	Running water	Jazz	60	Stan Getz
	49861	Money	pop	60	John Lennon

5.3.2.2 *First experiment:* experimental results on the available contextual datasets:

We begin by evaluating our proposed models: the neighborhood based model (FWR) and CAMF based model (WCAMF-MCS and ICAMF-MCS strategies) according to MAE, Precision@N (Prec@N), Recall@N (Rec@N) and NDCG@N with N $\in$ {5,10}. We present in Tables 5.8 and 5.9 the obtained experimental results versus seven baselines on Music, Movie, LDOS-CoMoDa and Food datasets. We indicate by the bold numbers the best results on each dataset. We can observe from the two tables below, that in the most of the cases the proposed CAMF based model is able to outperform the proposed neighborhood based model. Take ICAMF-MCS strategy for example, we can find that, it gives an improvement of the Prec@5 value by 28.1%, 16.4%, 45.9% and 14.5% over FWR, on Music, Movie, LDOS-CoMoDa and Food datasets respectively.

TABLE 5.8: Comparison results on the Music and Movie datasets

Dataset	Algorithm	MAE	Prec@5	Prec@10	Rec@5	Rec@10	NDCG@5	NDCG@10
Music	ItemKNN	0.983	0.015	0.014	0.043	0.079	0.040	0.045
	UserKNN	1.087	0.013	0.015	0.038	0.091	0.043	0.042
	DCW	1.064	0.058	0.052	0.090	0.144	0.121	0.123
	FWR	**0.911**	0.064	0.070	0.106	0.161	0.143	**0.148**
	SVD++	0.965	0.036	0.025	0.183	0.179	0.117	0.110
	LRMF	1.270	0.024	0.017	0.186	0.134	0.077	0.075
	CAMF-Dev	1.001	0.014	0.018	0.142	0.150	0.042	0.037
	CAMF-MCS	0.998	0.033	0.031	0.118	0.166	0.112	0.092
	WCAMF-MCS	0.939	0.078	0.071	0.191	0.172	0.128	0.129
	ICAMF-MCS	0.920	**0.082**	**0.079**	**0.198**	**0.195**	**0.151**	0.141
Movie	ItemKNN	1.229	0.052	0.044	0.263	0.248	0.210	0.231
	UserKNN	1.242	0.055	0.045	0.275	0.285	0.202	0.201
	DCW	1.248	0.046	0.052	0.295	0.302	0.261	0.266
	FWR	1.240	0.061	0.062	0.302	0.322	**0.346**	**0.283**
	SVD++	1.688	0.057	0.028	0.268	0.104	0.222	0.105
	LRMF	1.395	0.053	0.042	0.276	0.251	0.224	0.136
	CAMF-Dev	1.229	0.048	0.045	0.281	0.311	0.226	0.119
	CAMF-MCS	1.529	0.052	0.049	0.391	0.351	0.245	0.123
	WCAMF-MCS	1.238	0.069	0.064	0.404	0.396	0.246	0.142
	ICAMF-MCS	**1.223**	**0.071**	**0.065**	**0.496**	**0.403**	0.248	0.184

Given the fact that the neighborhood based model can suffer from low accuracy problem due the absence of the knowledge learned about item aspects to produce

TABLE 5.9: Comparison results on the LDOS-CoMoDa and Food datasets

Dataset	Algorithm	MAE	Prec@5	Prec@10	Rec@5	Rec@10	NDCG@5	NDCG@10
LDOS-CoMoDa	ItemKNN	0.973	0.007	0.006	0.024	0.020	0.024	0.030
	UserKNN	0.952	0.004	0.005	0.019	0.025	0.021	0.027
	DCW	0.830	0.002	0.005	0.017	0.026	0.019	0.022
	FWR	0.821	0.037	0.025	0.026	0.028	0.031	0.033
	SVD++	0.871	0.024	0.025	0.018	0.017	0.021	0.040
	LRMF	2.004	0.011	0.012	0.012	0.009	0.007	0.005
	CAMF-Dev	0.867	0.022	0.020	0.027	0.022	0.006	0.006
	CAMF-MCS	1.021	0.042	0.032	0.016	0.010	0.008	0.005
	WCAMF-MCS	0.848	0.048	0.039	0.033	0.026	0.018	0.059
	ICAMF-MCS	**0.804**	**0.054**	**0.048**	**0.043**	**0.047**	**0.033**	**0.061**
Food	ItemKNN	1.183	0.060	0.065	0.106	0.146	0.119	0.118
	UserKNN	1.214	0.038	0.046	0.099	0.120	0.118	0.120
	DCW	1.206	0.069	0.032	0.105	0.118	0.101	0.107
	FWR	1.114	0.076	0.040	0.145	0.151	0.125	0.122
	SVD++	1.119	0.062	0.050	0.131	0.150	0.123	0.128
	LRMF	1.270	0.065	0.047	0.172	0.147	0.128	0.118
	CAMF-Dev	1.007	0.040	0.048	0.122	0.146	0.138	0.117
	CAMF-MCS	1.529	0.080	0.072	0.144	0.172	0.137	0.123
	WCAMF-MCS	0.938	0.086	0.073	0.190	0.177	0.139	**0.143**
	ICAMF-MCS	**0.927**	**0.087**	**0.082**	**0.199**	**0.197**	**0.154**	0.134

accurate top-N recommendations. In addition, the neighborhood formation process, especially the user-user similarity computation step requires the calculation of user's interest similarity with all other neighbors to make predictions or recommendations which may increase the computation complexity. While, matrix factorization is simply a mathematical tool that identify the latent features underlying the relations between the rating matrix entities, as a result it tends to produce much faster and satisfactory recommendations. However, in the case of having a sufficiently small number of users, neighborhood based model can outperform matrix factorization based model. For example, FWR improves the best performing strategy of CAMF based model by 5% and 53.8% in terms of NDCG@10 on Music and Movie datasets respectively. For the proposed CAMF based model, we can observe a little difference between the two strategies ICAMF-MCS and WCAMF-MCS. Most commonly, the ICAMF-MCS strategy gives a better performance than WCAMF-MCS strategy. In this respect, we can note that, ICAMF-MCS slightly improves the MAE value over WCAMF-MCS by 2%, 1.2% and 1.2% on Music, Movie and Food datasets respectively. ICAMF-MCS strategy is also able to beat WCAMF-MCS strategy in terms of Prec@10 and Rec@10 on LDOS-CoMoDa dataset by an improvement of 23.1% and 80.7% respectively. The obtained experimental results show the superior performance of the ICAMF-MCS strategy especially on rich contextual datasets. In fact, this latter strategy takes into account the interaction that may exist between the relevant contextual dimensions according to their fuzzy measures. Therefore, the strategy that considers correlated contextual dimensions outperforms the one considering independent contextual dimensions. As a result, the interaction among the relevant contextual dimensions may be considered as a better framework to understand and represent the contextual effects on recommendation. For instance, a user may more precisely decide a movie if the time contextual dimension is correlated with companion dimension rather than considering these contextual dimensions separately.

Turning now to study the performance quality of our proposals against each of the baselines. As expected, Tables 5.8 and 5.9 show that, in most cases, the proposed models outperform the comparative baselines in the presented datasets. We observe that the proposed neighborhood-based CF model (FWR) can significantly improves the rating accuracy metric MAE over the previous popular neighborhood-based CF approaches (ItemKNN, UserKNN and DCW). For example, FWR achieves an MAE value equals to 1.114 while the best performing neighborhood-based baseline achieves an MAE value equals to 1.183 on Food dataset. It also can be found that, on Music dataset, FWR improves the MAE value from 0.983 (the MAE of the best performing neighborhood-based baseline) to 0.911. Furthermore, when it comes to the top-N recommendation task, FWR is also able to achieve higher ranking metric values and thus beat the neighborhood-based baselines. For instance, on LDOS-CoMoDa dataset, FWR gives an improvement in terms of Rec@5 by 8.3% over ItemKNN , 36.8% over the UserKNN and 52.9% over the DCW. Therefore, the comparative neighborhood-based CF models always show lower results than our neighborhood based model FWR. A possible explanation for this is that these baselines ruled out the influence of contextual dimensions relevancy and interaction in determining suitable neighbors with similar contexts which may increase the computational complexity in the neighborhood formation process and thus decrease recommendation accuracy. On account of the fact that considering relevant and correlated contextual dimensions to infer user's situation can eliminate redundant users with possible similar items, but they may not have closer contextual situations as the active user.

Regarding the comparison between the matrix factorization-based models, we can notice that in general the comparatives CAMF based models (CAMF-Dev and CAMF-MCS) work better than MF models (SVD++ and LRMF). Given the fact that CAMF based models incorporate contextual information in the recommendation process in an advantageous way, increasing in turn the recommendations effectiveness. Nevertheless, it can be found that, in terms of MAE, MF models such as SVD++ can improve the CAMF-MCS by 36.6% on Food dataset, this may have occurred due to the small contextual conditions number in this dataset.

The two proposed strategies (WCAMF-MCS and ICAMF-MCS) can achieve a superior recommendation performance over prior CAMF models, particularly ICAMF-MCS strategy. It outperforms Rec@5 by 41.3% and 76.5% relative to CAMF-MCS and

CAMF-Dev, respectively on Movie dataset. Moreover, on LDOS-CoMoDa dataset, ICAMF-MCS makes better Rec@5 value by 59.3% and 168.7% than CAMF-Dev and CAMF-MCS respectively.

Let us note that in nearly all the cases, ICAMF-MCS obtains the preferable results consistently which prove the accuracy of the proposed Interaction based CAMF strategy and confirms the efficiency of employing weighted correlated contextual dimensions in the prediction process using factorization techniques.

Evaluation on other performance measures

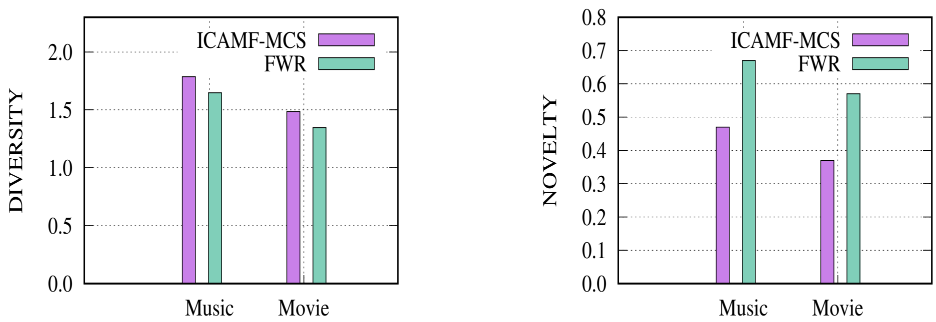


FIG. 5.13.: Comparison results in terms of novelty and diversity on Music and Movie datasets

Beyond rating metrics and ranking metrics, we point towards evaluating the proposed approaches while using other metrics like novelty and diversity (described in 1.4.2.3). We start by trying this evaluation on two different datasets. An extensive experimental evaluation with more datasets and other performance measures is under establishing by our team.

More precisely, we compute novelty and diversity for the best performing proposed CAMF-based strategy (ICAMF-MCS) and our neighbors-based approach (FWR). We choose these approaches since they manipulate the rating data in different ways and may thus produce varying recommendations. The two proposals share a common task: they both produce predicted ratings that can be used to recommend items. Thus, the idea is to use the generated predictions in order to offer personalized items recommendation. The results shown in Figure. 5.13 confirm that the diversity level in Music dataset is greater than Movie dataset. In fact, the good predictive accuracy in Music dataset, tends to upgrade the performance with respect to the contextual recommendation diversity. Moreover, there are more items in Music dataset and this factor may be responsible for this result since diversity is also related to how different the items are with respect to each other. We can also note, that the diversity results obtained by ICAMF-MCS are better than the ones generated by FWR on both Music and Movie datasets, possibly because the neighbors based model largely depends on the users profiles similarities which can negatively affect diversity; it seems that the neighbors who have already rated extensively will see the least diverse recommendations. When it comes to the novelty values, it can be seen from Figure. 5.13 that the results are lower than the average diversity values. That means that when a different recommendation appears, it is more often a recommendation that has appeared in the past, rather than something that has not appeared before. The recommendation produced by ICAMF-MCS lacks novelty, probably related to the cold-start items problems which frequently exist in the field of CAMF models.

5.3.2.3 *Second experiment:* experimental results on the created contextual datasets:

These experiments are carried out to investigate our models performance in large scale system. Table 5.10 illustrates the comparison results between the proposed models against the presented baselines on the created large contextual datasets. It is apparent from the table below that the proposed models are able to beat the presented baselines on the large datasets. For instance, the proposed FWR model achieves an important

TABLE 5.10: Comparison results on the large created datasets

Dataset	Algorithm	MAE	Prec@5	Prec@10	Rec@5	Rec@10	NDCG@5	NDCG@10
Contextual Movie&TV	ItemKNN	1.003	0.100	0.101	0.200	0.210	0.290	0.292
	UserKNN	1.009	0.100	0.100	0.201	0.202	0.292	0.298
	DCW	0.828	0.120	0.100	0.211	0.265	0.226	0.220
	FWR	0.506	0.137	0.129	0.229	0.281	0.313	0.333
	SVD++	1.023	0.162	0.121	0.178	0.278	0.228	0.311
	LRMF	1.104	0.160	0.162	0.150	0.215	0.217	0.278
	CAMF-Dev	1.114	0.178	0.171	0.179	0.287	0.223	0.353
	CAMF-MCS	1.101	0.172	0.100	0.180	0.321	0.230	0.380
	WCAMF-MCS	0.485	0.290	**0.190**	0.321	0.343	0.406	0.411
	ICAMF-MCS	**0.457**	**0.298**	0.144	**0.371**	**0.347**	**0.463**	**0.460**
Contextual MovieLens	ItemKNN	0.902	0.012	0.008	0.064	0.081	0.051	0.077
	UserKNN	0.885	0.012	0.009	0.063	0.080	0.051	0.075
	DCW	0.898	0.015	0.008	0.067	0.081	0.050	0.089
	FWR	0.561	0.036	0.040	0.145	0.171	0.125	0.122
	SVD++	0.869	0.045	0.025	0.125	0.179	0.180	0.192
	LRMF	1.110	0.012	0.015	0.064	0.153	0.063	0.098
	CAMF-Dev	0.767	0.038	0.021	0.192	0.219	0.126	0.122
	CAMF-MCS	1.035	0.047	0.033	0.193	0.327	0.193	0.198
	WCAMF-MCS	0.522	0.076	0.077	0.219	**0.377**	0.223	0.224
	ICAMF-MCS	**0.519**	**0.097**	**0.092**	**0.225**	0.321	**0.235**	**0.231**

rating prediction accuracy progress of 63.6% and 57.7% against the best performing neighborhood based baseline on the two created datasets: contextual Movie & TV and contextual MovieLens respectively. Turning now to the ranking metrics, the obtained results show a more important gap between FWR and the best performing neighborhood based baseline. For example, on the contextual MovieLens dataset, FWR

substantially outperforms the Prec@5 value by 140%, the Rec@5 value by 116.4% and the NDCG@5 value by 145%. If we look at the proposed CAMF based model, we find that ICAMF-MCS strategy is able to generate reasonably interesting rating prediction quality by enhancing the MAE value by 140.9% and 47.8% over the best performance among the presented CAMF competitors on contextual Movie & TV and contextual MovieLens datasets respectively. In addition, ICAMF-MCS outperforms the Prec@5 value by 67.4%, Rec@5 value by 106.1% and NDCG@5 value by 98.7% over the best performing CAMF based baseline on the contextual Movie & TV dataset.
Obviously the proposed FWR and CAMF-MCS models are able to make a paramount progress on the created large datasets, which demonstrates that our proposals are particularly helpful on rich contextual datasets with multiple users. The conducted experiments confirm that our proposals are scalable to large datasets, principally the CAMF based model.
The experiments findings point to the usefulness of the contextual dimensions relevancy and interaction on the rating prediction accuracy and thus on the recommendation performance, essentially on the created datasets. Imagine for example, it is snowing, the user is on lunch break, and he prefers to have lunch in an outdoor restaurant. In this situation, the importance of the mood dimension should be little. Yet, the weather and the time of day are the relevant contextual dimensions that should also be combined together to select the suitable restaurant.

The obtained results in the conducted experiments have led us to conclude that attributing an importance weight to each contextual dimension and each subset of dimensions then distinguishing the relevant and correlated contextual dimensions can significantly improve the rating prediction accuracy and as a result the recommendations performance over the conventional methods.

5.4 Online Experiments

It is increasingly known that the majority of prior recommendation approaches have focused on the accuracy of predictions or the performance of recommendations. In fact, nearly all the collaborative-filtering based approaches were developed, with one definite target in mind: to enhance the prediction quality.

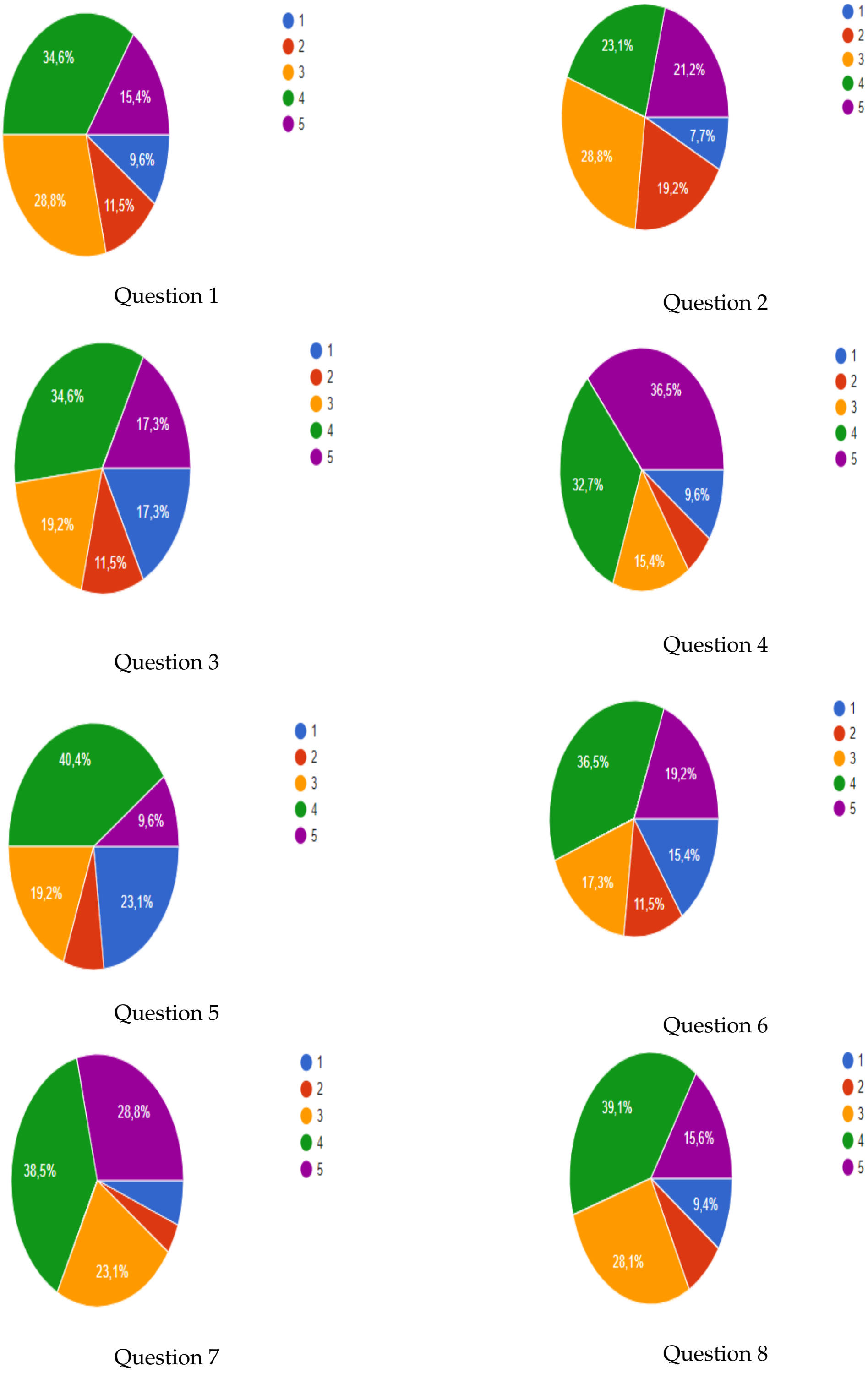


FIG. 5.14.: The percentage of participants answers per degree of satisfaction given for each question

To evaluate these approaches, offline ratings and ranking metrics are used, but these metrics can partially assess a recommender system. The underlying assumption is that the recommender ability to capture user interest will upgrade user satisfaction measuring the user acceptance of the recommendations. However, some experts have seen that user satisfaction is not always related to a valuable recommendation accuracy [157]. User satisfaction is therefore required for evaluating recommender systems. Given the fact that if the users are satisfied with the recommended resources, they will develop trust in the recommender system and keep using it. Despite this interest, most studies have focused on evaluating the recommendation accuracy and few researchers have addressed the question of investigating whether the recommended items match users satisfaction [158, 159].

Hayes et al. [160] stated that it is required to measure real-life user satisfaction for recommender systems. Consequently, we choose to use an online questionnaire to study the users satisfaction on the recommended items in accordance with users contexts. Via our five-point Likert scale questionnaire, participants were asked to provide their satisfaction degrees about the recommended music genres in a given context using 5 answer options from (1) `"I do not like very much"` to (5) `"I really like"`. We opt for eight questions so as not to overload the participants with a lot of questions which can lead to more noise data from the responses. Besides, it is more easier for the participants to provide specific information when there is a sufficient number of questions, bearing in mind that users often shy away from answering questionnaires. A total of different 52 participants volunteered to participate in this questionnaire. They are from various professions (e.g., students, teachers, software developers etc.) and their ages range from 20 to 35. The details and the posed questions can be found in Appendix A. Having collected all the participants responses on the degrees of satisfaction, we turn to compute the average results of the responses for each question. The Figure. 5.14 illustrates at a glance the percentage of these results. From the figure, we note that for the majority of questions, the degree of (3) `"I like"` or (4) `"I often like"` or (5) `"I really like"` are among the most widely chosen in the participants answers. For instance, in response to Question 4, it can be viewed that the majority of respondents (36,5%) really liked the recommended items, whereas only 5.8 % didn't like very much the recommended items. For the Question 2, approximately 28.8% who responded with the `"I like"` degree, while 7.7 % of answers belong to the `"I do not like very much"` degree. For the remainder of questions, the large majority of respondents select the `"I often like"` degree. The obtained results are then used to compute an average satisfaction score for all participants by employing the Customer Satisfaction Score (CSAT) metric [161] which directly computes customer satisfaction level. The computed CSAT score is equal to 78.6 % which means that the most of participants are satisfied with the given recommendations.

5.5 Conclusion

In this chapter, we presented the conducted offline and online experiments to assess the context-aware recommendation proposals effectiveness. Our findings, with respect to the rating prediction accuracy and recommendation performance on both real-word available and large created contextual datasets, demonstrate that our proposals yielded promising results.
Moreover, the online experiment, where real users feedback is considered, sheds further light for finer analysis capturing user satisfaction on the recommended items.
We will present, in the next chapter, the carried out experiments to evaluate the effectiveness of the proposed context-aware multi-criteria approaches.

Chapter 6

Evaluation of the Contextual Multi-criteria Recommendation Approach

6.1 Introduction

This chapter focuses on evaluating our proposals that integrate the contextual information and items criteria ratings into the recommendation process. Therefore, we present the various experiments carried out to test the effectiveness of our context-aware multi-criteria proposed models regarding a set of models discussed in the literature. For each proposed model, we introduce the two main parts of the evaluation experiments including the experimental setting and the comparison results.

6.2 Evaluation of the Bipartite Graph Based Model

For our experimental evaluation, we investigate the following research questions: is our research hypothesis **H** valid ? what is the comparative effectiveness when using prioritized operators versus average operator for overall item rating prediction ? how do our co-clustering based prioritized multi-criteria aggregation model perform in comparison to representative baselines ?

6.2.1 Experimental Evaluation Setting

To evaluate our context-aware multi-criteria model based on bipartite graph co-clustering, we start by describing the used dataset. Then, we introduce the baseline recommendation algorithms and the evaluation metrics.

6.2.1.1 Dataset

We conduct our experiments using a real-world dataset from the tourism domain: TripAdvisor data [139]. This dataset was collected by Jannach et al. using a Web crawling process which registers users ratings on hotels situated in various locations. TripAdvisor dataset is the most appropriate dataset that corresponds to our evaluation purpose since: (1) user's context is provided based on a contextual dimension referring to the *season* dimension. From the trip date information expressed in months in the dataset, we can derive the season dimension (e.g., June, July, August are the summer season months). (2) Users ratings for individual items criteria, plus an overall rating for items are available. The items criteria are: *quality of rooms*, *value for the money*, *cleanliness of the hotel*, *the hotel location*, *experience of check-in*, *overall quality of service and business services*. On average, each user has provided at least 3 ratings for the hotels in the dataset, which satisfies the experiments requirement. This dataset contains a total of 22.130 ratings provided by 1502 users on 14.300 hotels. Our created bipartite graph is built upon $m = 3916$ users situational contexts related to all the available $n = 7$ criteria. Considering the prioritized operators employed for multi-criteria ratings aggregation, we adopt the users criteria ratings in the training data to obtain an average score for each criterion to be used for building the criteria personalized ranking as required.

6.2.1.2 Baselines

For comparison purposes, we use baselines that fall in three categories of recommendation algorithms. Table 6.1 presents the three types of baselines that have been tested on the publicly available TripAdvisor dataset.

6.2.1.3 Evaluation metrics

We evaluate the effectiveness of our model and baselines using the popular accuracy metric "Mean Absolute Error (MAE)" and the popular ranking metric "F-measure" that joins both Precision and Recall metrics together to provide a more stable view of a recommender system performance.

TABLE 6.1: Baselines

Category	*Algorithms*	*Ref*	*Descriptions*
Single-rating approaches	`Biased Matrix Factorization (BiasMF)`	[122]	Adds biases values of user and item in the extension matrix decomposition model.
	`User-based K-Nearest Neighbors (UserkNN)`	[151]	Exploits users k-nearest neighbors to predict a score for each user-item pair.
Multi-criteria rating approaches	`Aggregation based approach (Agg)`	[86]	Uses traditional multi-criteria ratings prediction and linear aggregations.
	`All criteria aggregation (CluAllCrit)`	[124]	Aggregates all the criteria ratings from the same or close cluster by a linear aggregation.
	`Criteria- Independent Contextual Model (CIC)`	[99]	Uses traditional multi-criteria ratings prediction and conditional aggregations.
	`Criteria Chains: Aggregation Model (CCA)`	[99]	Uses criteria chains for multi-criteria ratings prediction and linear aggregations.
	`Criteria Chains: Contextual Model (CCC)`	[99]	Adopts criteria chains to predict criteria ratings and conditional aggregations.
Context-aware rating approach	`Context-Aware Matrix Factorization (CAMF)`	[58]	Integrates contextual information in the classical matrix factorization.

In fact, increases in the Precision results can come at the expense of decreases in the Recall results sometimes. Thus, "F-measure" can be written as follows:

$$F - measure = \frac{2.Precision.Recall}{Precision + Recall} \tag{6.1}$$

We present in the comparison results, the F-measure values on the top-10 recommendations, as the results based on the top-5 recommendations display approximately the similar patterns. We compute these metrics by adopting a training-testing methodology for both parameter tuning and evaluation. For this purpose, we fixed a splitting ratio of training/test of 80/20.

6.2.2 Results and Discussion

6.2.2.1 Research hypothesis validation

Returning to the hypothesis **H** posed in Chapter 4 which suggests that **"Users in similar contexts tend to have similar interests for similar criteria"**. To validate our research hypothesis **H**, we realize a statistical analysis to identify the relationships strength among contextually similar users depending on their criteria importance. In this respect, we perform a correlation analysis on all the users with criteria ratings of similar items in similar contexts from TripAdvisor dataset. We first compute for each user the importance of each rated criterion to determine users favorite criteria in accordance with their contexts. For this purpose, we exploit the user's tendency of providing ratings to the various criteria of a rated item in a given context. Then, we compute a weighted average of the ratings tendency across the rated items by a user in a specific context. More precisely, we integrate user's context information in the formula used for finding users preferred criteria in [87]. The revised formula is the following:

$$\Gamma^{c}_{u,i,co} = r^{c}_{u,i,co} - \bar{r}^{c}_{i,co} \tag{6.2}$$

$$Pref^{c}_{u,co} = \frac{\sum_{i \in I_{u,co}} n_{i,co} \times \Gamma^{c}_{u,i,co}}{\sum_{i \in I_{u,co}} n_{i,co}} \tag{6.3}$$

Where $\Gamma^{c}_{u,i,co}$ is the tendency of providing rating for criterion c of a rated item i by the user u in the context co with $r^{c}_{u,i,co}$ is the rating given by u for criterion c of the item i in the context co and $\bar{r}^{c}_{i,co}$ is the average rating. The term $I_{u,co}$ denotes the set of items rated by u in his context co and $n_{i,co}$ is the number of users who rated item i in the same context co. Then, we study the strength of these users relationships in compliance with their criteria importance by computing one of the most common measures of dependencies the "Spearman's rank correlation coefficient" [100] denoted r_s, and ranging in the interval $-1 \leq r_s \leq 1$. The closer r_s is to 1, the more likely the positive correlation is strong and as r_s value goes towards 0, the correlation will be weaker. For analyzing the computed correlation coefficient values, we employ the following rule of thumb:

- .00 to .19 "very weak correlation"
- .20 to .39 "weak correlation"
- .40 to .59 "moderate correlation"
- .60 to .79 "strong correlation"
- .80 to 1.0 "very strong correlation"

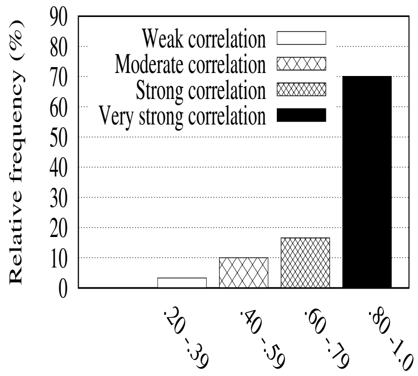


FIG. 6.1.: Distribution of the correlation coefficient values

We present in Figure. 6.1, the relative frequency of users in each correlation strength interval, which displays the percent of the total users represented by each interval. Note that relative frequencies should add up to approximately 100%, although the total might be slightly higher or lower due to rounding error. Figure. 6.1 pinpoints the elevated percentage corresponding to the very strongly related users in similar contexts. In fact, the relative frequency distribution indicates that the percentage of users with very strong relationships is around 70%. The obtained result indicates that the large majority of users in similar situational contexts attain a fairly important positive correlation coefficient according to their preferences for similar criteria. Furthermore, for better interpretation, we compute the p-value corresponding to the significance level of correlations and we obtain p-value = $0.014 \leq 0.05$ (a common threshold for statistical significance). This points that the correlation is statistically significant. Thus, there is a favorable agreement between contextually similar users on criteria importance. Consequently, according to the findings of this study, we could conjecture that the more close the users contexts, the more these users tend to have close criteria importance which lends a strong support for our research hypothesis **H**.

6.2.2.2 Comparative evaluation of the prioritized operators based models

Turning our focus to the second objective of our evaluation, we investigate in this experimental scenario the comparative effectiveness of the "Scoring" and "And" prioritized operators with the standard "Average" operator on the overall item rating prediction. Precisely, to asses the joint effect of the used aggregation operators and the co-clusters number on rating prediction accuracy, we perform various co-clusters numbers ranging from 2 to 10. Figure. 6.2 plots the MAE results by the three compared aggregation operators.

From this figure, we note that the two prioritized aggregation operators (ie., "Scoring" and "And" operators) outperform the non-prioritized aggregation operator (ie., "Average" operator). Particularly, for a number of co-clusters ranging from 5 to 8, the "Scoring" and the "And" operators attain an average improvement of 19.9% and 14.6% respectively over the "Average" aggregation operator. This finding points to the effectiveness of the prioritized combination of criteria in the co-clusters, which leads to obtain personalized overall rating prediction results in accordance with users preferences. This result corroborates our intuition behind the benefit of leveraging users' preferences in order to rank and differentiate the criteria strengths among criteria and among users. In the presented comparisons, the best performing operator in terms of predictive accuracy is the "Scoring" operator due to the appropriateness of the criteria importance order according to users contexts.

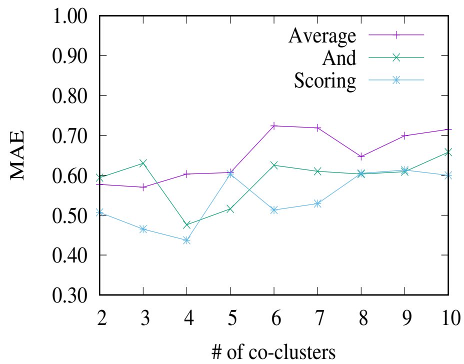


FIG. 6.2.: Comparison of the aggregation operators and impact of the co-clusters number

Figure. 6.2 also shows that the number of co-clusters is able to impact the prediction accuracy. It is apparent that when the co-clusters number increases from 2 to 4 , the prediction accuracy slightly improves. It means that the accuracy becomes higher since the information within each co-cluster is more specialized and tied to users. However, the prediction accuracy tends to be steady when the co-clusters number keeps rising. Regarding the graph partitioning method, increasing the number of co-clusters could be interpreted as being a result of several small sub-matrices derived from the rating matrix. However, a sufficient amount of data is needed for the MF algorithm to provide accurate criteria ratings prediction. Therefore, the criteria aggregation process can not obtain pleasing results under a reasonable threshold of data given by the co-clusters, which leads to a downside impact on the prediction accuracy.

As a result, we set in the remaining experiments the co-clusters number to 4 and 3 for the prioritized operators based models and the "Average" operator based model respectively.

6.2.2.3 Comparison results with baselines

Impact of the number of latent factors:

Before starting the experimental evaluation, we begin by tuning one of the important parameters for matrix factorization models which is the latent factor number that we denote by F. Figure. 6.3 plots the obtained results in terms of MAE metric with respect to the variation of F. We can observe that the MAE value of our "And" operator based model turns down to the lowest in cluster 2 and cluster 3 when F = 12. Consequently, the best choice for both cluster 2 and 3 is when F is equal to 12. If we now turn to cluster 1 and 4, the MAE value by our "And" operator based model reaches an importing prediction accuracy when F is equal to 10. On the other hand, the prediction accuracy by our model based on "Scoring" operator advances in all clusters when the latent factors number attains 10. Hence, based on these experimental results, the latent factor number when the proposed models achieve the best prediction accuracy is tuned.

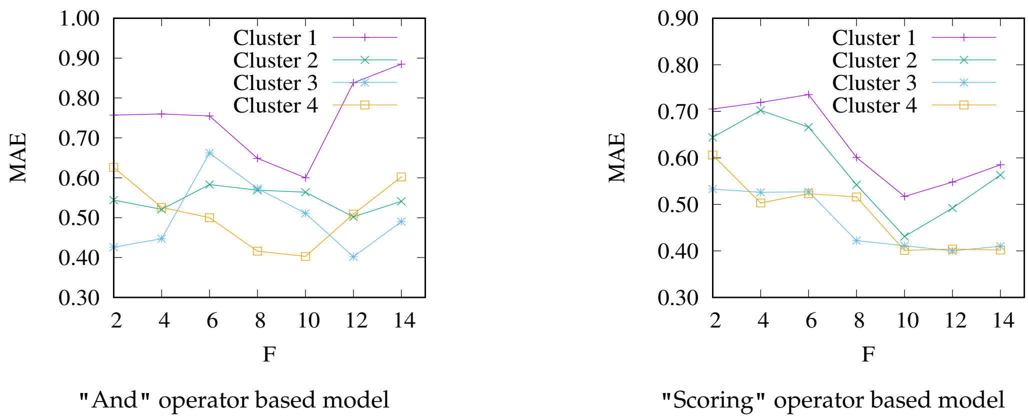


FIG. 6.3.: F variation on the prioritized operators based models

Comparison results

In this scenario, we turn to compare the effectiveness of our proposal with state-of-the-art recommendation approaches with respect to the last objective of our evaluation. The results of multi-criteria baselines are reported from the corresponding published research articles referenced in the Table 6.2 using their optimal parameters on the TripAdvisor dataset. The results of the remaining categories of baselines are computed through the toolkit CARSKit [141]. In Table 6.2, *IR Scoring* and *IR And* designate the improving rate achieved by employing the "Scoring" operator and the "And" operator respectively in comparison with each of the baselines.

TABLE 6.2: Comparison results for the rating prediction task

Category	*Algorithms*	*MAE*	*IR Scoring*	*IR And*
Traditional single rating approaches	BiasMF [122]	0.894	+104.5%	+87.8%
	UserKNN [151]	0.883	+102.1%	+85.5%
Multi-criteria rating approaches	Agg [86]	0.752	+72.1%	+57.9%
	CIC [99]	0.756	+72.9%	+58.8%
	CCA [99]	0.710	+62.4%	+49.2%
	CCC [99]	0.460	+5.3%	-3.5%
	CluAllCrit [124]	2.545	+482.4%	+434.7%
Context-aware rating approach	CAMF [58]	0.639	+46.2%	+34.2%
Our model	Average	**0.570**	-	-
	Scoring	**0.437**	-	-
	And	**0.476**	-	-

As can be seen from Table 6.2, our proposed approach is able to beat the comparative baselines by reaching higher prediction accuracy. In fact, we can find that, when applying the "Scoring" operator, our approach achieves a significant improvement of $+72.1\%$, $+72.9\%$ and $+62.4\%$ over Agg, CIC and CCA models respectively. The same improving trend holds for the "And" operator-based model. The possible explanation for this finding is that the multi-criteria baselines (Agg, CIC, CCA) employ either a traditional algorithm to predict the criteria ratings, a traditional aggregation to predict the overall rating, or both which may decrease the prediction quality. For the clustering based multi-criteria model (CluAllCrit) which employs a linear aggregation, we note that it degrades the prediction accuracy in comparison with other multi-criteria baselines. Consequently, the proposed model achieves a considerable improvement over CluAllCrit ($+482.4\%$ and $+434.7\%$ by using the "Scoring" and the "And" operators respectively). The low accuracy level of CluAllCrit may be caused by the problem of the automatic coefficients of criteria obtained using a traditional aggregation. Besides, even the use of a clustering technique to improve the prediction results, employing such coefficients to perform the aggregation may produce predicted ratings with negative values or outside the [1..5] scale. In this case, the multi-criteria based single clustering model cannot bring an improvement accuracy over multi-criteria models.

In comparison with the multi-criteria CCC model, that takes into account the criteria dependency in predicting the criteria ratings and employs conditional aggregations, we observe a little difference in the accuracy results between this latter model and our "Scoring" based model. These findings indicate that there might exist dependent criteria impacting the user's decision in choosing an item. Nevertheless, using the conditional aggregation by viewing the criteria preferences as contexts may not always be a good choice, due to the fact that CIC model performs worse than CCA model which uses a linear aggregation function.

For the contextual baseline algorithm, it can be seen that CAMF is able to make headway over the majority of baselines but still outperformed by our proposal by $+46.2\%$ with the "Scoring" operator and by $+34.2\%$ with the "And" operator. The absence of other additional information such as the multi-criteria feedback in the CAMF model may well be responsible for this result.

Generally speaking, our findings reveal that particularly in situations where there are various criteria ratings, it can be beneficial to regard the criteria strength according to user's context. However, the strength of the criteria in the majority of related work are assumed to be equal over users which could explain the superiority of our approach. In addition, this explanation is concurred with cross-comparing the obtained findings through using the prioritized operators in the one hand against the average aggregation and the CAMF model on the other hand. In fact, we can observe that the MAE value is reduced from 0.639 to 0.570 when adding the contextual information and decreased more to less 0.480 when additionally using the prioritized operators.

From another side, considering the recent best performing baselines and the best performing proposed model in terms of predictive accuracy, we evaluate here the ranking of the high predicted items as a way of evaluating top-N recommendations, because a high-accuracy doesn't necessary guarantee a good ranking. The obtained results measured according to the F-measure are reported in Table 6.3. We can observe that our model based on the "Scoring" operator shows strongly significant improvement comparing with the previous models. In fact, in our approach the F-measure is equal to 0.667 while the obtained value of the best performing baseline is equal to 0.482. Therefore, our proposal is also able to beat the presented baselines in the top-N recommendation task by achieving a higher ranking metric value.

In summary, our experimental findings demonstrate that: (1) co-clustering users situational contexts and criteria to predict multi-criteria ratings, and (2) aggregating the predicted multi-criteria ratings in a prioritized way directly influence rating prediction accuracy and recommendation performance.

TABLE 6.3: Comparison results for the top-N recommendations task

Category	*Algorithms*	*F-measure*
Multi-criteria rating approaches	CIC [99]	0.463
	CCA [99]	0.482
	CCC [99]	0.449
Our model	Scoring	**0.667**

6.3 Evaluation of the Tripartite Graph Based Model

In this section, we experimentally examine the presented research questions: is our research assumption H_1 valid? How do our context-aware multi-criteria recommendation model based on tripartite graph co-clustering perform comparing with the baselines ?

6.3.1 Experimental Evaluation Setting

To evaluate our context-aware multi-criteria model based on tripartite graph co-clustering, we start by presenting the settings needed for our experiments including the used datasets, evaluation protocol, evaluation metrics, baselines and configurations.

6.3.1.1 Datasets

One major difficulty in the recommendation research area is to find available datasets with different additional information, especially the ones with both users contextual information and items multi-criteria preferences. As far as we know, there are only two available datasets that provide multi-criteria users feedback in a given context:

- **TripAdvisor dataset** [139]: this dataset was described and used in the previous experiment. What differentiates its use in the current experiments from earlier experiments is that, we will work with three different contextual dimensions unlike working with a single contextual dimension previously. In fact, the used contextual dimensions are: the *trip type*, the *year*, and the *season* which is derived from the trip date information.
- **Educational dataset** [136]: this dataset contains approximately 3,306 ratings provided by 269 students on 70 topics of projects. It also contains ratings of 3 individual item criteria (application, data and ease), plus 1 overall rating. Averagely, every student selected around 3 topics of projects and provided overall ratings for these topics. In this dataset, 3 types of contextual dimensions are presented: the *type of the class*, the *year of the course* and the *semester*.

6.3.1.2 Evaluation protocol

For the parameter tuning and evaluation, we employ for our evaluation protocol, the 5-fold cross-validation technique. We use the MAE metric for studying the impacts of the employed parameters on the ratings prediction accuracy. Since it was proven that ranking-based metrics are better suited to evaluate the quality of the recommendations [155], we focus on evaluating the common top-N recommendation task. Therefore, to assess the quality of the top-N recommendations, the evaluation is done using the proper ranking metric "F-measure". As the previous experiment, we present the results based on the top-10 recommendations.

6.3.1.3 Baselines

Regarding baselines selection, we compare our proposal with common conventional state-of-the-art recommendations algorithms that fall in four categories. The first one considers single rating based algorithms which only employ a single criterion for item rating (`BiasMF`). The second category concerns contextual based algorithms which incorporate contextual information in the traditional recommendation process (`CAMF`). The third category includes the multi-criteria rating based algorithms which take into account the multi-criteria ratings of items (`Agg`, `CIC`, `CCA`). The last category contains context-aware multi-criteria rating based algorithms which integrate both contextual and criteria information (`DCL`, `ICL`, `ICC`, `CDL`, `DCC`). The baselines belonging to the three first categories were previously described in Table 6.1, we briefly describe in the following the remaining baselines belonging to the last category:

- `Criteria-Dependent Contextual Linear Model (DCL)` [98]: employs dependent criteria ratings prediction and contextual linear aggregations.
- `Criteria-Independent Contextual Linear Model (ICL)` [98]: employs independent contextual multi-criteria ratings prediction and linear aggregations.
- `Criteria-Independent Contextual Conditional Model (ICC)` [98]: employs independent contextual multi-criteria ratings prediction and conditional aggregations.
- `Criteria-Contextual Dependent Linear Model (CDL)` [98]: employs contextual dependent multi-criteria ratings prediction and linear aggregations.
- `Criteria-Dependent Contextual Conditional Model (DCC)` [98]: employs dependent contextual criteria ratings prediction and conditional aggregations.

Note that the used baselines results for the comparison study are generated by the toolkit CARSKit [141] for the first and the second baselines categories. Whereas the remaining baselines results are reported using their optimal parameters on the same datasets used in our experiments from the published research paper [98].

6.3.1.4 Configurations

To carry out the rating prediction task on the multi-criteria, it is first required to select the suitable contextual condition that can be used as the best split. In this respect, we employ the impurity criterion t_{mean} [137] which can estimate for each contextual condition, the statistical significance of the difference in the means of its ratings by a t-test. It is always practical to fix a threshold for the splitting process so that users can only be split when the splitting criteria meets the significance requirement. Thus, we employ a threshold that approximately reaches the 0.05 level of statistical significance. Having performed the splitting process, we apply the correlation-based CAMF presented in formula (4.10) on the resulting matrices for generating item criteria ratings in each co-cluster. To better assess the correlation-based CAMF, we have experienced a range of various latent factors F values ($5 \leq F \leq 60$, increment 5) and training iteration It ($10 \leq It \leq 100$, increment 10). We also handle others parameters such as the learning and regularization factors by CARSKit toolkit [141], where SGD is applied as the optimization method.

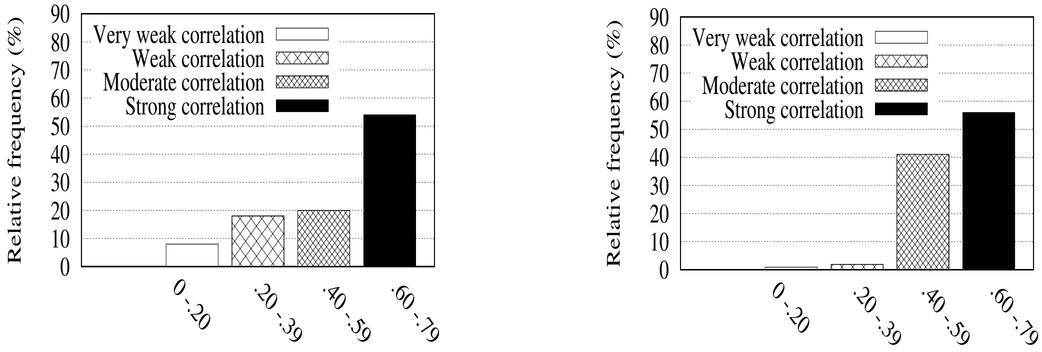
6.3.2 Results and Discussion

6.3.2.1 Research hypothesis validation

To validate the posed hypothesis H_1 (See Section 4.4.1.2), we run preliminary experiments including quantitative and qualitative analyzes.

Quantitative analysis:

We aim in this quantitative study to identify the significance and the strength of the correlations among contextually similar users according to their criteria importance. For this purpose, we start by determining users preferred criteria according to their contextual situations by computing users criteria importance using the formula 6.3 in which we integrate users contextual situations.



TripAdvisor Data Educational Data

FIG. 6.4.: Following the rule of thumb, correlations close to +0.70 or -0.70 indicate a strong relationship; correlations closer to +0.5 and -0.5 show a moderate relationship; and correlations less than +0.5 and -0.5 show a weak relationship. Student t-test significance : p-value $\leq$ 0.05.

Then, we investigate the strength of users correlations according to their criteria importance by the Spearman's rank correlation coefficient (r_s) [100]. Figure. 6.4 illustrates for each correlation strength interval the users relative frequency and the significance testing based on the Student t-test for the TripAdvisor and Educational datasets. It can be seen from the first figure that in the TripAdvisor dataset, over half of the users in similar contextual situations are strongly correlated, where the Spearman's correlation coefficient value falls within the interval [0.60, 0.79]. For the Educational dataset, we note from the figure to the right that there is a high percentage of strongly correlated users in similar contextual situations corresponding to 56%. These findings are also considered as statistically significant with p values between .01 and .05 which underlines the significant agreement between contextually similar users on criteria importance. As a result, we could conjecture that the more similar the contextual situations of the users, the more these users tend to have similar criteria importance, as underlined by our hypothesis H_1.

Qualitative analysis:

Probing for a deeper understanding of the impact of users contextual situations on criteria preferences, we further the quantitative analysis with a qualitative analysis to obtain a better insight on hypothesis H_1. We describe in Table 6.4 different users in various contextual situations rating the same item from TripAdvisor dataset where our model integrating criteria ratings is able to predict the relevant hotels to recommend.

TABLE 6.4: Example of users preferences in TripAdvisor dataset

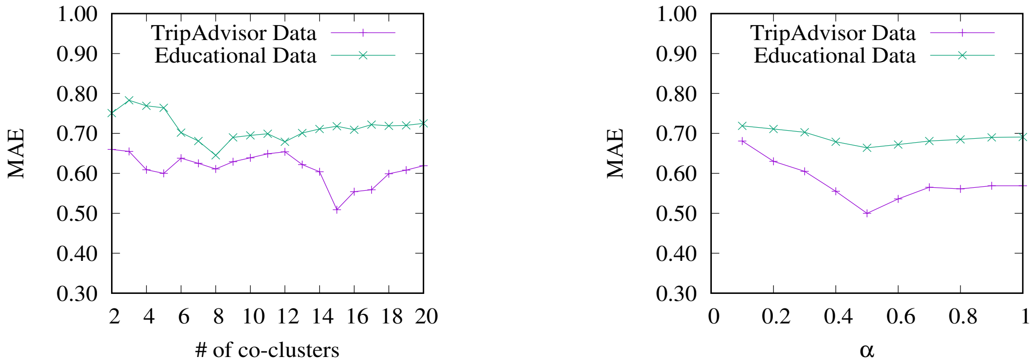
Contextual situation	**User**	**Item**	**Rated Criteria**	**Overall Rating**
Summer 2011, traveled with family	u_1	Miramonti Hotel	value: 4; sleep quality: 4; cleanliness: 4; service: 5	4.2
Summer 2011, traveled with family	u_2	Miramonti Hotel	value: 4; sleep quality: 5; cleanliness: 5; service: 5	4.9
Summer 2011, traveled with family	u_3	Miramonti Hotel	value: 4; sleep quality: 5; cleanliness: 5 ; service: 4	4.8
Winter 2011, traveled as couple	u_4	Miramonti Hotel	location: 3; value: 4; sleep quality: 5; rooms: 4	3.1
Autumn 2013, traveled as couple	u_5	Miramonti Hotel	location: 4; value: 4; sleep quality: 3 ; rooms: 4; cleanliness: 5; service: 4	3.8

Table 6.4 is revealing that the three first users in similar contextual situations (u_1, u_2 and u_3) share 4 similar criteria (*value of the money, sleep quality, cleanliness and service*) among the 7 available criteria in the TripAdvisor dataset and they also give close preferences to these criteria which can make them very strongly correlated. Given the fact that, for a summer family trip, it is obvious to give importance to the value of the money criterion since the hotels expenses get high over summer. Added to that, there might be additional charges for children. Moreover, to enjoy a comfy family vacation, it is necessary to be satisfied by the sleep quality, the cleanliness and the delivered service. For the remaining users (u_4 and u_5), it is apparent that they own various criteria preferences compared to the users in the three first rows (u_1, u_2 and u_3). In fact, they are affected by other criteria like location and room when choosing a hotel in different travel contextual situations like traveling in the low season as couple. Indeed, the location criterion can play an important role in selecting a suitable hotel for such situation, since it is preferable for a couple to select a hotel with beautiful views and also it should be appropriate for the low season climate. Besides, it is of interest to have a comfy and quiet room. To obtain criteria preferences for each co-cluster, we apply our predictive model by considering the criteria associated to each clustered contextually similar users, where the first users cluster contains u_1, u_2 and u_3, the second cluster contains u_4 and the last one contains u_5. Finally, we obtain for the item "Miramonti Hotel" the overall predicted ratings with close important values for u_1, u_2 and u_3. For the others users (u_4, u_5) involved in different contextual situations and included into different clusters, we remake that their corresponding criteria preferences lead to obtain different overall ratings compared to those obtained for the first users cluster.

As a result, these experiments findings provide concrete proof of the assumption we make and appear to provide a strong support for our research hypothesis H_1.

6.3.2.2 Parameter tuning

Before running the main experiments, we start by studying the sensitivity of some parameters on the prediction accuracy. For this purpose, we start by varying the co-cluster number L from 2 to 20 and we study the prediction accuracy of the proposed model on TripAdvisor and Educational datasets.



Impact of the co-clusters number

Impact of the parameter α

FIG. 6.5.: Parameter tuning

From the first figure, it can been seen that when L = 8 the MAE value on the Educational dataset degrades to the lowest. Thus, we reach the conclusion that 8 is the suitable co-clusters number for our model on the Educational dataset. For TripAdvisor dataset, we observe that our model needs 15 co-clusters to achieve the best prediction accuracy. We then report for each dataset the co-clusters number when our model attains the best prediction result.
The second parameter to be tuned is α that was used in formula (4.9) to trade-off between the two bipartite graphs forming our tripartite graph. For this task, we tried different values of α ranging within the interval [0, 1] and we note that the prediction accuracy results are affected by the variation of α in the above figure to the right. Indeed, when α= 0.1 our model obtains a low accuracy and when α is closer to 1 the accuracy tends to be steady. This indicates the dominance of one bipartite graph which can be the consequence of the tripartite graph partitioning failure since the used co-clustering method fails when considering only one bipartite graph. We achieve the best MAE value when α is around 0.5. Thus, we fix this value for both datasets.

6.3.2.3 Comparison results with baselines

Here, we turn to compare the effectiveness of our proposal with state-of-the-art recommendation approaches with respect to the second objective of our evaluation. We present in Table 6.5 the obtained F-measure results on TripAdvisor and Educational datasets, where "Improv" is the improvement achieved by the proposed model compared to each baseline model.

TABLE 6.5: Comparison results on the TripAdvisor and Educational datasets

		TripAdvisor dataset		*Educational dataset*	
Category	*Baselines*	*F-measure*	*Improv*	*F-measure*	*Improv*
Single-rating approach	`BiasMF`[122]	0.0007	**+67.1%**	0.0698	**+19.7%**
Context-aware rating approach	`CAMF` [58]	0.0011	**+48.4%**	0.0715	**+17.8%**
Multi-criteria rating approaches	`Agg` [86]	0.0010	**+53.1%**	0.0720	**+17.2%**
	`CIC` [99]	0.0009	**+55.4%**	0.0700	**+19.5%**
	`CCA` [99]	0.0013	**+38.9%**	0.0682	**+21.6%**
Context-aware multi-criteria rating approaches	`DCL` [98]	0.0015	**+29.6%**	0.0620	**+28.7%**
	`ICL` [98]	0.0017	**+20.2%**	0.0749	**+13.9%**
	`ICC` [98]	0.0018	**+15.5%**	0.0765	**+12.1%**
	`CDL` [98]	0.0015	**+26.8%**	0.0660	**+24.1%**
	`DCC` [98]	0.0019	**+10.8%**	0.0677	**+22.2%**
	Our Model	**0.0021**	-	**0.0870**	-

It can be seen from the table that generally in both datasets, the baseline models that take in account the contextual information only (i.e., `CAMF`) or the multi-criteria preferences only (i.e., `Agg`, `CIC`, `CCA`) can reach superior recommendation performance compared to the traditional model `BiasMF` that doesn't integrate neither context nor multi-criteria ratings. For instance, in TripAdvisor dataset, the F-measure value of the traditional `BiasMF` model is improved by $+36.4\%$ when incorporating the contextual information using `CAMF` model and by $+46.2\%$ when considering the multi-criteria ratings using `CCA` which is the best performing multi-criteria recommendation baseline. These findings confirm that extended recommendation models with context or criteria preferences information generally work better compared to traditional recommendation models without additional information. Yet, in the Educational dataset, not all the multi-criteria recommendation models can achieve better performance. For example, `CCA` model represents the worst performing multi-criteria recommendation baseline leading to a drop in the performance by -2.3% over the traditional `BiasMF` model. In fact, the multi-criteria models use different methods for rating prediction (dependent or independent) which justify the different performance results obtained by these models. The findings show that the used dependant way for criteria ratings prediction by `CCA` model may not be convenient in the Educational dataset. This may due to the nature of the considered criteria in the Educational dataset, where students may provide conflicting preferences for these criteria.

In comparison with the context-aware multi-criteria models (i.e., `DCL`, `ICL`, `ICC`, `CDL`, `DCC`) which appertain to the same category as our proposal, we observe that in the TripAdvisor dataset, the F-measure results are always significantly improved especially when integrating the contextual information in the multi-criteria ratings prediction process. The findings obtained from the Educational dataset present close patterns except for `DCL`, `CDL` and `DCC`. Despite the fact that the context is considered to upgrade the models recommendation performance based on a dependent criteria ratings prediction method (`CDL` and `DCC`), the obtained f-measure results decline from 0.0720 for the non contextual multi-criteria `Agg` model to 0.0660 and 0.0677 for `CDL` and `DCC` models respectively. Added to that, when incorporating the contextual information in the process of ratings aggregation only using `DCL`, decreased results can be seen in the Educational dataset. Consequently, even though an improved recommendation performance could be achieved in the Educational dataset by contextualizing the criteria ratings prediction process, it certainly counts on the applied criteria ratings prediction method. In particular, on the Educational dataset, our model outperforms `ICL` and `ICC` models that use independent contextual criteria ratings prediction and different rating aggregations methods by $+13.9\%$ and $+12.1\%$ respectively. As a result, our proposal improves the best performing baseline `ICC` in the Educational dataset which employs conditional aggregations. This finding underlines that improving the aggregation step only may be insufficient sometimes to significantly enhance the recommendation performance because it is also useful to develop the criteria ratings prediction step, since the overall user preference is predicted based on these criteria ratings. This result shows that our model using a dependant contextual criteria ratings prediction can beat both `ICL` and `ICC` models independently of the applied aggregation way. In comparison with the closest baseline (`CDL`) that employs dependent contextual criteria ratings prediction and linear aggregation, we obtain an important progress by our model ($+24.1\%$) on the Educational dataset. This result appears to prove that our proposal based on a dependant method is beneficial in dealing with the multi-criteria ratings prediction issue, since it underlines not only the relationship between contexts and users in a reduced recommendation space but also highlights the correlations among the different criteria in the prediction process. Nevertheless, the criteria ratings prediction dependent way used by `CDL` consists in integrating all the available criteria ratings. In this case, when the criteria are not in fact dependent, some important information may be missed which can induce lower recommendation accuracy with enhancing its complexity. More precisely, in the Educational dataset, the `"application"` criterion indicates the student's taste on the projects domain, while `"data"` and `"ease"` criteria present the projects difficulty from the students perspective.

While some students prefer to choose a simple project, others may prefer to select challenging projects which induces in some conflicting preferences that may reduce the recommendation performance when employing an inappropriate dependent way for criteria ratings prediction. In our proposed model, we have addressed this problem by taking into account only the important criteria rated by users in similar contextual situations in the same cluster which gives a reason for the superiority of our model in the Educational dataset. In the TripAdvisor dataset, our proposal is also able to significantly outperform the F-measure value against all the presented baselines. Particularly, it can even beat the closer baseline (`CDL`) by $+26.8\%$. Besides, we can find that our model slightly improves the best baseline employing a dependent contextual criteria ratings prediction and conditional aggregations (`DCC`) by $+10.8\%$. This implies that besides the performed improvement in the first step in our model, further improvement also needs to be done especially in the aggregation step.
The obtained findings have led us to conclude that the recommendation performance can be enhanced by merging context-awareness and multi-criteria directions by integrating the contextual information into the step of criteria ratings prediction. In addition, taking advantages of the relevant correlations that might exist between the criteria rated by the clustered contextually similar users might have important implications for generating more effective recommendations.

6.3.3 Conclusion

This chapter delivers several contributions starting by the multi-dimensional recommendation data modeling until reaching overall items ratings prediction. This was done by applying a set of techniques that take advantage of the positive impact of incorporating context-awareness and multi-criteria decision making in the recommendation process. We first investigate how to model the available input data including users feedbacks on items criteria, and the contextual situations in which these users are involved when selecting items. Therefore, we present the context-aware multi-criteria network by interconnected multi-type entities in the form of bipartite and tripartite graphs. Furthermore, we posed two research hypothesizes based on the modeled entities and their relationships in each graph to give insights about the desired co-clustering structure. Then, we exploit the obtained co-clusters to obtain partial user's item ratings which are then finally aggregated to estimate the overall impression of an item. The experimental evaluation undertaken on two real-world datasets (TripAdvisor and Educational datasets) demonstrate the effectiveness of our proposals in comparison to state of the art recommendation models.

Conclusion and Future Works

This last part of the document thesis briefly summarizes our research and propose potential extensions to address as future directions.

Summary and Conclusion

Recommender systems constitute a promising research area with many interesting underexplored topics and open research problems; some of them have been addressed in this thesis. The central thrust of this thesis tackles the main task in recommender systems which is the prediction of user's preferences. Therefore, we deal with the rating prediction problem to estimate how much a user likes a particular item by considering useful additional information. This dissertation delivers several contributions towards the purpose of enhancing items recommendation by predicting users preferences based on their contexts and their criteria preferences. Thus, the bibliographical study that we carried out has presented the traditional recommendation systems and a growing body of literature has examined several works related to context-based and multi-criteria recommendation approaches. At this level, we spotted various limitations challenging the recommendation domain and we thus built our contributions by extending the state of the art approaches in different ways. At the broadest level, these contributions are summarised below:

- Weighting the contextual dimensions: we deal with the problem of identifying the contextual dimensions that truly impact the decision-making process. To accomplish this task, we propose a weighting method based on identifying the fuzzy measures associated to the different contextual dimensions. The proposed method has the advantage of facilitating the task of interpreting relevant and interacted contextual dimensions. In fact, it is able to define the weight of importance not only for each individual contextual dimension but also for subsets of dimensions. This could eventually lead to determine relevant and correlated contextual dimensions.

- Inferring user's contextual situation: we make use of the relevant and correlated contextual dimensions supplied by the previous weighting task to recognize the user's contextual situation using fuzzy logic.
- Proposing two context-aware rating prediction models:
  1. A neighborhood-based model: the predicted rating is obtained by integrating the inferred user's contextual situation in the neighborhood-based rating prediction process.
  2. A matrix factorization-based model composed of two strategies: a weighting strategy that includes the relevant contextual dimensions weights in the rating prediction process and an interaction strategy that integrates the interaction measurement between correlated dimensions in the rating prediction process.
- Building new large contextual datasets: the datasets building consists of an enrichment process of large non-contextual datasets based on a contextual dataset to gain sufficient users contextual ratings to be considered for performing new comparisons and evaluations.
- Modeling the relationships between situational contexts and item criteria: starting from a rating matrix composed of users situational contexts as rows and criteria as columns, we explore a new way to integrate these two entities in the prediction process by modeling their connections by a bipartite graph.
- Modeling the relationships between users, contextual situations and criteria: to present the multi-dimensional available data, we model the entities resulting from connected heterogeneous recommendation data as a tripartite graph with three entities types (users, contextual situations and criteria). Therefore, we extend the previous bipartite graph to deal with additional nodes. We also underline a novel challenge through the tripartite graph modeling, involving weighting differently the three mentioned entities connections.
- Proposing two context-aware multi-criteria rating prediction models:
  1. A bipartite graph based model: the proposed model mainly relies on the bipartite graph spectral co-clustering for simultaneously partitioning users situational contexts and the items rated criteria. The obtained co-clusters are used to predict users criteria ratings that are then aggregated to compute the overall items ratings by applying prioritized aggregation operators which allow tailoring the criteria strengths to the users preferences.

2. A tripartite graph based model: we start by posing a research hypothesis to give insights about handling the tripartite graph partitioning. Following the posed hypothesis, we perform a high-order co-clustering as the fusion of pair-wise sub-problems over two bipartite graphs. Then, for predicting cluster-based multi-criteria ratings, we consider the relationship between contexts and users in a low dimensional space using a user splitting approach, and we also point to the correlations that may exist among criteria using a correlation-based rating prediction algorithm.

Future Works

The extensive work reported in this thesis has provided answers to some issues we have discussed, but it also has led to the emergence of new questions and perspectives. Therefore, this dissertation lays the foundation for several possible enhancements and extensions to enrich our proposals, some of which we mention in the following:

1. **Building a recommender system.**
   For each of our proposals, we have developed a predictive model that can predict how much a user likes a particular item according to different aspects. To receive items recommendation, it will be effective to build a complete recommender system. We mean by this, designing and implementing a human-recommender interaction layer on top of the core predictive model. Therefore, the user could be able to query the system for recommendations, specifying his preferences and contextual dimensions, and will receive useful suggestions tailored to his needs.

2. **Enlarging the experimental evaluation.**
   Even though we were able to build two new large datasets for contextual recommendations, where in one of them more than 9500 users involved in different contextual situations gave their judgements about more than 1000 items, we feel strongly that the experimental evaluation has to be undertaken on real users interacting with the system. As a result, we can measure the effect of the recommendation system on the users behaviors.

   Added to that, we aim to use more evaluation metrics in real scenarios. In fact, in the majority of previous recommendation approaches, the focus goes to evaluating the recommendation accuracy. Yet, user satisfaction can be considered as an important factor that we need to focus besides accuracy. For this purpose, we aim to conduct an extensive experimental evaluation using other performance measures such as novelty and diversity and provide a formal ground for the unification of the different ways of measuring these metrics in the recommendation area. Additionally, since a recommendation system is expected to generate rapid online recommendations, it is useful to measure how fast the system can provide these recommendations. Furthermore, alternative metrics such as serendipity which refers to the experience of discovering unexpected and relevant items, privacy and adaptivity have been less discovered in the literature. However, these metrics should be considered as they are closely related to the user's experience and satisfaction.

3. **Integrating social networks information.**
   We used different useful additional information to improve the recommendation quality including users surrounding contextual information and items multi-criteria ratings. To obtain better personalized recommendation results, we aim to provide an in-depth study about the user's surrounding to take advantage of using more useful information. Thus, we propose as a perspective, the enrichment of users profiles and extending the tripartite graph with more types of links between the graph entities. One example would be to additionally make use of user-to-user associations by taking into account social ties between users to consider the social impact.

4. **Incorporating technological opportunities such as the Internet Of Things (IoT) and blockchain.**
   As a future improvement, we intend to exploit the advantages of IoT by integrating contextualization with real-time data from sensors to provide accurate dynamic recommendations using deep learning techniques. We also aim to address the privacy issue by using secured solutions such as incorporating blockchain technologies that can be effective in protecting the recommendations.

Appendix A

Questionnaire

This online questionnaire is designed to investigate the users satisfaction about the recommended music according to users current contexts. More precisely, to understand the influence of context on the music preferences of a car driver, this latter is asked to evaluate the recommended music genres in different contextual conditions using a likert preference scale: 1 = "I do not like very much", 2 = "I like a little", 3 = "I like", 4 = "I often like", 5 = "I really like".

Question 1. Imagine that you are driving a car. Your radio station is broadcasting the following Pop music: "Paparazzi, Lady Gaga ". How likely is that you will listen to that music genre when you are feeling very active ?

- □ 1= "I do not like very much"
- □ 2 = "I like a little"
- □ 3 = "I like"
- □ 4 = "I often like"
- □ 5 = "I really like"

Question 2. Imagine that you are driving a car. Your radio station is broadcasting the following Hip Hop music: "Gangsta Paradise, Coolio". How likely is that you will listen to that music genre when your driving style is very sportive?

- □ 1= "I do not like very much"
- □ 2 = "I like a little"
- □ 3 = "I like"

- □ 4 = "I often like"
- □ 5 = "I really like"

Question 3. Imagine that you are driving a car. Your radio station is broadcasting the following Dance music: "This Is My Time, Calderone Inc". How likely is that you will listen to that music genre when you feel sad?

- □ 1= "I do not like very much"
- □ 2 = "I like a little"
- □ 3 = "I like"
- □ 4 = "I often like"
- □ 5 = "I really like"

Question 4. Imagine that you are driving a car. Your radio station is broadcasting the following Classical music: "Trout Quintet, Franz Schubert". How likely is that you will listen to that music genre when it is raining outside?

- □ 1= "I do not like very much"
- □ 2 = "I like a little"
- □ 3 = "I like"
- □ 4 = "I often like"
- □ 5 = "I really like"

Question 5. Imagine that you are driving a car. Your radio station is broadcasting the following Reggae music: "Satta Massagana, The Abyssinians". How likely is that you will listen to that music genre in the morning?

- □ 1= "I do not like very much"
- □ 2 = "I like a little"
- □ 3 = "I like"
- □ 4 = "I often like"
- □ 5 = "I really like"

Question 6. Imagine that you are driving a car. Your radio station is broadcasting the following Rock music: "Highway Star, Deep Purple". How likely is that you will listen to that music genre when you are lazy?

- ☐ 1= "I do not like very much"
- ☐ 2 = "I like a little"
- ☐ 3 = "I like"
- ☐ 4 = "I often like"
- ☐ 5 = "I really like"

Question 7. Imagine that you are driving a car. Your radio station is broadcasting the following House music: "One Love, David Guetta". How likely is that you will listen to that music genre at night?

- ☐ 1= "I do not like very much"
- ☐ 2 = "I like a little"
- ☐ 3 = "I like"
- ☐ 4 = "I often like"
- ☐ 5 = "I really like"

Question 8. Imagine that you are driving a car. Your radio station is broadcasting the following Country music: "I Walk the Line, Johnny Cash". How likely is that you will listen to that music genre when your are driving in a relaxed mood?

- ☐ 1= "I do not like very much"
- ☐ 2 = "I like a little"
- ☐ 3 = "I like"
- ☐ 4 = "I often like"
- ☐ 5 = "I really like"

Appendix B

Graph-based Recommender Systems

In the real world, the majority of the objects around us are explicitly or implicitly related to each other. That is to say, we belong to a world of graphs. This characteristic is even more evident in recommender systems where the objects considered here, such as users, items, contexts, criteria, are closely connected to each other and impact each other through various relations. In practice, different types of graphs arise from the recommender systems data, and they can improve the recommendations quality. Therefore, a number of researchers have proposed graph-based recommender systems algorithms exploiting graph properties in order to generate recommendations. The graphs can be of varied types ranging from normal graphs [162, 163] to bipartite [164–168] and even multipartite graphs [169–173].

B.1 Bipartite Graph-based Recommender Systems

The main entities in the recommendation domain are users and their rated items. Accordingly, the most common graph-based approaches are based on bipartite graphs where the relations are from one part of the network, users, to the other part, items. In [164], authors proposed an inductive graph-based matrix completion that trains a graph neural network on subgraphs around (user, item) pairs generated from the rating matrix and maps these subgraphs to their corresponding ratings. In [165], a recommendation model based on spectral collaborative filtering was proposed to take advantage from the rich information of connectivity existing in the spectral domain, where the relationship between users and items was formulated as a bipartite graph. Rashed et al. [166] put forward a non-linear co-embedding model for rating prediction that models the user-item relation as a bipartite graph by leveraging additional criteria and content features using their vector representations. Bedi et al. [167], modeled the sparse user-item rating data as a weighted bipartite graph and exploits the graph properties to generate recommendations. In the proposed work, user similarity is computed using a hybrid similarity metric combining cosine similarity measure and information entropy. Besides modeling users and items entities, an example of recommending people has been studied by Geil et al. [168], where a social bipartite graph of individuals on one side and the ones they follow on the other side was created to recommend people with the highest relevance scores.

B.2 Multipartite Graph-based Recommender Systems

Some recommendation methods have extended the bipartite graph by adding some entities to it. In this respect, Shams et al. [169] designed a graph-based approach for collaborative ranking domain that models the relations between users, items, and pairwise preferences in a tripartite graph structure, and analyze it to infer a recommendation list through a ranking algorithm. In [170], different types of nodes in a multi-layer structure were used to make context-aware recommendations through a random walk in the graph capturing users' preferences and current decision context. In [171], Bogers suggested a recommendation algorithm using Markov random walks over a contextual graph considering ratings information with contextual information to recommend movies. It models the browsing process of a user on a movie database website using the links between different objects such as users, items, tags, genres, and actors to construct the contextual graph. In [172], a tripartite graph based recommendation approach modeling the preference data of users and tag information of items was proposed to avoid the data sparsity and items cold start problems when creating users' recommendation lists. In [173], a user-item-tag tripartite graph was used to optimize personalized recommendation based on an integrated diffusion on user-item and item-tag relation.

Bibliography

[1] J. R. David Reinsel, John Gantz, "Data age 2025: The digitization of the world from edge to core.," *White Paper 25, International Data Corporation (IDC), USA*, 2018.

[2] L. Baltrunas and F. Ricci, "Context-dependent recommendations with items splitting.," *IIR*, vol. 560, pp. 71–75, 2010.

[3] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[4] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[6] J. L. Herlocker, J. A. Konstan, and J. Riedl, "Explaining collaborative filtering recommendations," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 241–250, 2000.

[7] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[8] F. Meyer, "Recommender systems in industrial contexts," *arXiv preprint arXiv:1203.4487*, 2012.

[9] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*, pp. 1–35, Springer, 2011.

[10] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "Mobile recommender systems in tourism," *Journal of network and computer applications*, vol. 39, pp. 319–333, 2014.

[11] A. J. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, and Q. Tang, "Privacy in recommender systems," in *Social media retrieval*, pp. 263–281, Springer, 2013.

[12] N. Polatidis and C. K. Georgiadis, "Mobile recommender systems: An overview of technologies and challenges," in *2013 Second International Conference on Informatics & Applications (ICIA)*, pp. 282–287, IEEE, 2013.

[13] L. Hong, L. Zou, C. Zeng, L. Zhang, J. Wang, and J. Tian, "Context-aware recommendation using role-based trust network," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 10, no. 2, pp. 1–25, 2015.

[14] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*, pp. 291–324, Springer, 2007.

[15] R. Dridi, S. Zammali, T. Alsulimani, and K. Arour, "Effective rating prediction based on selective contextual information," *Information Sciences*, vol. 510, pp. 218–242, 2020.

[16] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 1, pp. 103–145, 2005.

[17] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*, pp. 73–105, Springer, 2011.

[18] J. B. D. CarlKadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Microsoft Research Microsoft Corporation One Microsoft Way Redmond, WA*, vol. 98052, 1998.

[19] R. Bell, Y. Koren, and C. Volinsky, "Modeling relationships at multiple scales to improve accuracy of large recommender systems," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 95–104, 2007.

[20] T. Miranda, M. Claypool, A. Gokhale, T. Mir, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *In Proceedings of ACM SIGIR Workshop on Recommender Systems*, Citeseer, 1999.

[21] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial intelligence review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[22] I. Soboroff and C. Nicholas, "Combining content and collaboration in text filtering," in *Proceedings of the IJCAI*, vol. 99, pp. 86–91, sn, 1999.

[23] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 253–260, 2002.

[24] J. Bennett, S. Lanning, and N. Netflix, "The netflix prize," in *In KDD Cup and Workshop in conjunction with KDD*, 2007.

[25] S. Chou, Y. H. Yang, and Y. Lin, "Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics," in *Procceding IEEE International Conference on Multimedia and Expo*, pp. 1–6, Italy, 2015.

[26] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, (New York, NY, USA), pp. 22–32, ACM, 2005.

[27] E. Blanchard, M. Harzallah, and P. Kuntz, "A generic framework for comparing semantic similarities on a subsumption hierarchy," in *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, (Amsterdam, The Netherlands, The Netherlands), pp. 20–24, IOS Press, 2008.

[28] B. N. Schilit and M. M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, no. 5, pp. 22–32, 1994.

[29] G. Chen and D. Kotz, "A survey of context-aware mobile computing research," *Dartmouth Computer Science Technical Report TR2000-381*, 2000.

[30] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *IEEE Network*, vol. 8, no. 5, pp. 22–32, 1994.

[31] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human–Computer Interaction*, vol. 16, no. 2-4, pp. 97–166, 2001.

[32] J. Grudin, "Partitioning digital worlds: focal and peripheral awareness in multiple monitor use," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 458–465, 2001.

[33] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender systems handbook*, pp. 217–253, Springer, 2011.

[34] B. Lamche, Y. Rödl, C. Hauptmann, and W. Wörndl, "Context-aware recommendations for mobile shopping," in *Proceedings of the Workshop on Location-Aware Recommendations, LocalRec , co-located with the 9th ACM Conference on Recommender Systems (RecSys), Vienna, Austria, September 19*, pp. 21–27, 2015.

[35] T. Stepan, J. M. Morawski, S. Dick, and J. Miller, "Incorporating spatial, temporal, and social context in recommendations for location-based social networks," *IEEE Transactions on Computational Social Systems*, vol. 3, pp. 164–175, Dec 2016.

[36] A. M. Otebolaku and M. T. Andrade, "Context-aware personalization using neighborhood-based context similarity," *Wireless Personal Communications*, pp. 1–24, 2016.

[37] S. Zammali, K. Arour, and A. Bouzeghoub, "A context features selecting and weighting methods for context-aware recommendation," in *39th IEEE Annual Computer Software and Applications Conference, COMPSAC , Taichung, Taiwan, July 1-5, 2015. Volume 2*, pp. 575–584, 2015.

[38] Y. Hu, P. Lee, K. Chen, J. Tarn, and D. Dang, *Hotel recommendation system based on review and context information: A Collaborative Filtering approach*. Pacific Asia Conference on Information Systems, 2016.

[39] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the fifth ACM conference on Recommender systems*, pp. 301–304, 2011.

[40] Q. Zhao, Y. Kou, D. Shen, T. Nie, and G. Yu, "Combining influence and sensitivity to factorize matrix for multi-context recommendation," in *13th Web Information Systems and Applications Conference, WISA , Wuhan, China, September 23-25*, pp. 71–76, 2016.

[41] C.-L. Liu and X.-W. Wu, "Large-scale recommender system with compact latent factor model," *Expert Syst. Appl.*, vol. 64, pp. 467–475, Dec. 2016.

[42] H. Miao, B. Luo, and Z. Sun, *An Improved Context-Aware Recommender Algorithm, in: Intelligent Computing Theories and Application: 12th International Conference, ICIC, Lanzhou, China, August 2-5*, pp. 153–162. 2016.

[43] M. Braunhofer and F. Ricci, "Selective contextual information acquisition in travel recommender systems," *Information Technology & Tourism*, vol. 17, pp. 5–29, Mar 2017.

[44] Y. Zheng, B. Mobasher, and R. Burke, "Incorporating context correlation into context-aware matrix factorization," in *Proceedings of the International Conference*

on Constraints and Preferences for Configuration and Recommendation and Intelligent Techniques for Web Personalization, (Aachen, Germany), pp. 21–27, 2015.

[45] X. Zheng, Y. Luo, L. Sun, and F. Chen, "A new recommender system using context clustering based on matrix factorization techniques," *Chinese Journal of Electronics*, vol. 25, no. 2, pp. 334–340, 2016.

[46] Y. Jhamb, "Machine learning models for context-aware recommender systems," 2018.

[47] A. Bozanta and B. Kutlu, "Developing a contextually personalized hybrid recommender system," *Mobile information systems*, vol. 2018, 2018.

[48] Z. Meng, R. McCreadie, C. Macdonald, and I. Ounis, "Variational bayesian context-aware representation for grocery recommendation," *arXiv preprint arXiv:1909.07705*, 2019.

[49] I. H. Sarker, "A machine learning based robust prediction model for real-life mobile phone data," *Internet of Things*, vol. 5, pp. 180–193, 2019.

[50] M. Al-Ghossein, *Context-aware recommender systems for real-world applications*. PhD thesis, Université Paris-Saclay (ComUE), 2019.

[51] I. H. Sarker, Y. B. Abushark, and A. I. Khan, "Contextpca: Predicting context-aware smartphone apps usage based on machine learning techniques," *Symmetry*, vol. 12, no. 4, p. 499, 2020.

[52] R. R. Yager, "Fuzzy logic methods in recommender systems," *Fuzzy Sets and Systems*, vol. 136, no. 2, pp. 133–149, 2003.

[53] S.-I. Lee and S.-Y. Lee, "Collaborative filtering based context information for real-time recommendation service in ubiquitous computing.," *Int. J. Fuzzy Logic and Intelligent Systems*, vol. 6, no. 2, pp. 110–115, 2006.

[54] N. Karacapilidis and L. Hatzieleftheriou, "Exploiting similarity measures in multi-criteria based recommendations," in *International Conference on Electronic Commerce and Web Technologies*, pp. 424–434, Springer, 2003.

[55] A. Livne, M. Unger, B. Shapira, and L. Rokach, "Deep context-aware recommender system utilizing sequential latent context," *arXiv preprint arXiv:1909.03999*, 2019.

[56] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM conference on recommender systems*, pp. 233–240, 2016.

[57] M. Jin, X. Luo, H. Zhu, and H. H. Zhuo, "Combining deep learning and topic modeling for review understanding in context-aware recommendation," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1605–1614, 2018.

[58] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, (New York, USA), pp. 301–304, 2011.

[59] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 17–23, Canada, 2012.

[60] M. Unger, "Latent context-aware recommender systems," in *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pp. 383–386, Vienna, Austria, 2015.

[61] V. Codina, F. Ricci, and L. Ceccaroni, *Exploiting the Semantic Similarity of Contextual Situations for Pre-filtering Recommendation, in : User Modeling, Adaptation, and Personalization: 21th International Conference, UMAP 2013, Rome, Italy, June 10-14*, pp. 165–177. 2013.

[62] X. Fan, Y. Hu, R. Zhang, W. Chen, and P. Brézillon, "Modeling temporal effectiveness for context-aware web services recommendation," in *the Proceedings of the IEEE International Conference on Web Services (ICWS) , New York, USA, June 27 - July 2*, pp. 225–232, 2015.

[63] U. Panniello, A. Tuzhilin, and M. Gorgoglione, "Comparing context-aware recommender systems in terms of accuracy and diversity," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 35–65, 2014.

[64] D. Bouneffouf, "Towards user profile modelling in recommender system," *arXiv preprint arXiv:1305.1114*, 2013.

[65] Y. Zheng, "Utility-based multi-criteria recommender systems," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 2529–2531, 2019.

[66] A. Odić, M. Tkalčič, J. F. Tasič, and A. Košir, "Predicting and detecting the relevant contextual information in a movie-recommender system," *Interacting with Computers*, vol. 25, no. 1, pp. 74–90, 2013.

[67] A. Odić, M. Tkalčič, A. Košir, and J. F. Tasič, "A.: Relevant context in a movie recommender system: Users' opinion vs. statistical detection," in *In: Proc. of the 4th Workshop on Context-Aware Recommender Systems, CARS, September 9, Dublin, Ireland*.

[68] H. Lu, J. Caverlee, and W. Niu, "Discovering what you're known for: A contextual poisson factorization approach," in *Proceedings of the ACM Recommender Systems conference (RecSys)*, 2016.

[69] Y. Zheng, "Situation-aware multi-criteria recommender system: Using criteria preferences as contexts," in *Proceedings of the Symposium on Applied Computing*, SAC '17, (New York, NY, USA), pp. 689–692, ACM, 2017.

[70] T. Strang, C. Linnhoff-Popien, and K. Frank, *CoOL: A Context Ontology Language to Enable Contextual Interoperability*, pp. 236–247. 2003.

[71] C. Anagnostopoulos, Y. Ntarladimas, and S. Hadjiefthymiades, "Situational computing: An innovative architecture with imprecise reasoning," *Journal of Systems and Software*, vol. 80, no. 12, pp. 1993 – 2014, 2007.

[72] H. Chaker, M. Chevalier, C. Soulé-Dupuy, and A. Tricot, *Business Context Information Manager: An Approach to Improve Information Systems*, pp. 67–70. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[73] X. Lin, B. Cheng, and J. Chen, "A situation-aware approach for dealing with uncertain context-aware paradigm," in *GLOBECOM 2009-2009 IEEE Global Telecommunications Conference*, pp. 1–6, IEEE, 2009.

[74] C. Richthammer and G. Pernul, "Situation awareness for recommender systems," *Electronic Commerce Research*, pp. 1–24, 2018.

[75] L. Niu, J. Lu, and G. Zhang, "Managerial cognition," in *Cognition-Driven Decision Support for Business Intelligence*, pp. 31–37, Springer, 2009.

[76] M. Stanners and H. T. French, "An empirical study of the relationship between situation awareness and decision making," 2005.

[77] H.-S. Park, J.-O. Yoo, and S.-B. Cho, *A Context-Aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory*, pp. 970–979. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[78] Y. Kim and S. B. Cho, "A recommendation agent for mobile phone users using bayesian behavior prediction," in *Proceedings of the Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 283–288, Oct Sliema, Malta, 2009.

[79] T. GS and U. P. Kulkarni, "Article: Design and implementation of user context aware recommendation engine for mobile using bayesian network, fuzzy logic and rule base," *International Journal of Computer Applications*, vol. 40, pp. 47–63, February,United Kingdom 2012.

[80] A. Sen and M. Larson, "From sensors to songs: A learning-free novel music recommendation system using contextual sensor data," in *Proceedings of the Workshop on Location-Aware Recommendations, LocalRec , Vienna, Austria, September 19*, pp. 40–43, 2015.

[81] R. Hermoso, J. Dunkel, and J. Krause, "Situation awareness for push-based recommendations in mobile devices," in *International Conference on Business Information Systems*, pp. 117–129, Springer, 2016.

[82] Y. Jung, C. Hur, and M. Kim, "Sustainable situation-aware recommendation services with collective intelligence," *Sustainability*, vol. 10, no. 5, 2018.

[83] J. Dötterl, R. Bruns, and J. Dunkel, "Incorporating situation awareness into recommender systems.," in *ICEIS (2)*, pp. 676–683, 2017.

[84] B. Roy, *Multicriteria methodology for decision aiding*, vol. 12. Springer Science & Business Media, 2013.

[85] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender systems handbook*, pp. 769–803, Springer, 2011.

[86] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intelligent Systems*, vol. 22, pp. 48–55, May 2007.

[87] R. S. Sreepada, B. K. Patra, and A. Hernando, "Multi-criteria recommendations through preference learning," CODS, (NY, USA), pp. 1:1–1:11, 2017.

[88] M. Wasid and R. Ali, "An improved recommender system based on multi-criteria clustering approach," *Procedia Computer Science*, vol. 131, pp. 93–101, 2018.

[89] A. Kouadria, O. Nouali, and M. Al-Shamri, "A multi-criteria collaborative filtering recommender system using learning-to-rank and rank aggregation," *Arabian Journal for Science and Engineering*, 09 2019.

[90] S. Gupta and V. Kant, "An aggregation approach to multi-criteria recommender system using genetic programming," *Evolving Systems*, vol. 11, no. 1, pp. 29–44, 2020.

[91] V. Kant, T. Jhalani, and P. Dwivedi, "Enhanced multi-criteria recommender system based on fuzzy bayesian approach," *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12935–12953, 2018.

[92] M. Nilashi, E. Yadegaridehkordi, O. Ibrahim, S. Samad, A. Ahani, and L. Sanzogni, "Analysis of travellers' online reviews in social networking sites using fuzzy logic approach," *International Journal of Fuzzy Systems*, vol. 21, no. 5, pp. 1367–1378, 2019.

[93] Z. Batmaz and C. Kaleli, "Ae-mccf: An autoencoder-based multi-criteria recommendation algorithm," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9235–9247, 2019.

[94] M. Hamada and M. Hassan, "Artificial neural networks and particle swarm optimization algorithms for preference prediction in multi-criteria recommender systems," in *Informatics*, vol. 5, p. 25, Multidisciplinary Digital Publishing Institute, 2018.

[95] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[96] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[97] Q. Li, C. Wang, and G. Geng, "Improving personalized services in mobile commerce by a novel multicriteria rating approach," WWW, (NY, USA), pp. 1235–1236, 2008.

[98] Y. Zheng, S. Shekhar, A. Anna Jose, and S. Kumar, "Integrating context-awareness and multi-criteria decision making in educational learning," SAC, ACM, 2019.

[99] Y. Zheng, "Criteria chains: A novel multi-criteria recommendation approach," in *IUI*, 2017.

[100] Y. Zheng, S. Shekhar, A. Anna Jose, and S. Kumar, "Integrating context-awareness and multi-criteria decision making in educational learning," in *the Proceedings of the 34th ACM SIGAPP Symposium on Applied Computing (ACM SAC)*, (Limassol, Cyprus), April 2019.

[101] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci, "Context relevance assessment and exploitation in mobile recommender systems," *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 507–526, 2012.

[102] M. Bazire and P. Brézillon, "Understanding context before using it," in *Modeling and Using Context* (A. Dey, B. Kokinov, D. Leake, and R. Turner, eds.), (Berlin, Heidelberg), pp. 29–40, Springer Berlin Heidelberg, 2005.

[103] M. Sugeno, *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.

[104] M. Grabisch, I. Kojadinovic, and P. Meyer, "A review of methods for capacity identification in choquet integral based multi-attribute utility theory: Applications of the kappalab R package," *European Journal of Operational Research*, vol. 186, no. 2, pp. 766 – 785, 2008.

[105] M. Grabisch, "Set function over finite sets: transformations and integrals," in *Handbook of Measure Theory*, vol. 2, pp. 1381–1401, Elsevier, 2002.

[106] D. Dubois and H. Prade, "An introduction to fuzzy systems," *Clinica Chimica Acta*, vol. 270, no. 1, pp. 3 – 29, 1998.

[107] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338 – 353, 1965.

[108] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[109] M. Grabisch and C. Labreuche, "A decade of application of the choquet and sugeno integrals in multi-criteria decision aid," *Annals of Operations Research*, vol. 175, no. 1, pp. 247–286, 2009.

[110] P. Cingolani and J. Alcala-Fdez, "jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation," in *2012 IEEE International Conference on Fuzzy Systems*, pp. 1–8, IEEE, 2012.

[111] W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," *Fuzzy sets and systems*, vol. 108, no. 2, pp. 159–178, 1999.

[112] M. G. Voskoglou, "Comparison of the cog defuzzification technique and its variations to the gpa index," *arXiv preprint arXiv:1612.00742*, 2016.

[113] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186, 1994.

[114] Y. Zheng, R. Burke, and B. Mobasher, *Recommendation with Differential Context Weighting in : 21th International Conference, Rome, Italy, June 10-14*. 2013.

[115] L. Yang and M. Ha, "A new similarity measure between intuitionistic fuzzy sets based on a choquet integral model," in *Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, (USA), pp. 116–121, 2008.

[116] X. C. Chunqiao Tan, "Dynamic similarity measures between intuitionistic fuzzy sets and its application, International journal of fuzzy systems," vol. 16, pp. 511–519, 2014.

[117] K. Chapphannarungsri and S. Maneeroj, "Combining multiple criteria and multidimension for movie recommender system," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2009.

[118] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, (NY, USA), pp. 269–274, 2001.

[119] S. Huang, Z. Xu, I. W. Tsang, and Z. Kang, "Auto-weighted multi-view co-clustering with bipartite graphs," *Information Sciences*, vol. 512, pp. 18 – 30, 2020.

[120] Z. Wang, M. Zhou, and C. Arnold, "Toward heterogeneous information fusion: bipartite graph convolutional networks for in silico drug repurposing," *Bioinformatics*, vol. 36, pp. i525–i533, 07 2020.

[121] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, pp. 24–45, Jan 2004.

[122] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, pp. 30–37, Aug. 2009.

[123] Z. Khan, N. Iltaf, H. Afzal, and H. Abbas, "Enriching non-negative matrix factorization with contextual embeddings for recommender systems," *Neurocomputing*, vol. 380, pp. 246–258, 2020.

[124] L. Liu, N. Mehandjiev, and D.-L. Xu, "Multi-criteria service recommendation based on user criteria preferences," RecSys '11, (New York, NY, USA), pp. 77–84, ACM, 2011.

[125] C. da Costa Pereira, M. Dragoni, and G. Pasi, "Multidimensional relevance: A new aggregation criterion," in *European Conference on Information Retrieval*, pp. 264–275, Springer, 2009.

[126] C. da Costa Pereira, M. Dragoni, and G. Pasi, "Multidimensional relevance: A new aggregation criterion," in *Advances in Information Retrieval*, (Berlin, Heidelberg), pp. 264–275, Springer Berlin Heidelberg, 2009.

[127] C. da Costa Pereira, M. Dragoni, and G. Pasi, "A prioritized "and" aggregation operator for multidimensional relevance assessment," in *AI*IA*, (Berlin), pp. 72–81, 2009.

[128] P. Liu and W. Liu, "Multiple-criteria decision making method based on the scaled prioritized operators with unbalanced linguistic information," *Artificial Intelligence Review*, pp. 1–25, 2020.

[129] S. B. Aydemir and S. Y. Gündüz, "Extension of multi-moora method with some q-rung orthopair fuzzy dombi prioritized weighted aggregation operators for multi-attribute decision making," *Soft Computing*, vol. 24, no. 24, pp. 18545–18563, 2020.

[130] S. Marrara, G. Pasi, and M. Viviani, "Aggregation operators in information retrieval," *Fuzzy Sets and Systems*, vol. 324, pp. 3–19, 2017.

[131] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, p. 513–523, Aug. 1988.

[132] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, (NY, USA), pp. 41–50, 2005.

[133] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, (Ny, USA), pp. 650–658, 2008.

[134] J. He, H. Tong, S. Papadimitriou, T. Eliassi-Rad, C. Faloutsos, and J. Carbonell, "Pack: Scalable parameter-free clustering on k-partite graphs," in *Society for Industrial and Applied Mathematics - 9th SIAM International Conference on Data Mining 2009, Proceedings in Applied Mathematics 133*, vol. 110, pp. 1278–1287, dec 2009.

[135] Y. Zheng, R. Burke, and B. Mobasher, "Splitting approaches for context-aware recommendation: An empirical study," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, (New York, NY, USA), pp. 274–279, ACM, 2014.

[136] Y. Zheng, "Personality-aware decision making in educational learning," in *Proceedings of the 23rd international conference on intelligent user interfaces companion*, IUI '18 Companion, (New York, NY, USA), pp. 1–2, Association for Computing Machinery, 03 2018.

[137] L. Baltrunas and F. Ricci, "Context-based splitting of item ratings in collaborative filtering," in *Proceedings of the 3th ACM Conference on Recommender Systems*, RecSys '09, (NY, USA), pp. 245–248, 01 2009.

[138] R. B. Statnikov, J. Matusov, and A. R. Statnikov, *Multicriteria Engineering Optimization Problems: Statement, Solution and Applications*, vol. 155. Springer, 11 2012.

[139] D. Jannach, M. Zanker, and M. Fuchs, "Leveraging multi-criteria customer feedback for satisfaction analysis and improved recommendations," *Information Technology Tourism*, vol. 14, pp. 119–149, 07 2014.

[140] C. Ono, Y. Takishima, Y. Motomura, and H. Asoh, "Context-aware preference model based on a study of difference between real and supposed situation data," pp. 102–113, 2009.

[141] Y. Zheng, B. Mobasher, and R. D. Burke, "Carskit: A java-based context-aware recommendation engine," in *IEEE International Conference on Data Mining Workshop, ICDMW, Atlantic City, NJ, USA, November 14-17*, pp. 1668–1671, 2015.

[142] A. Košir, A. Odic, M. Kunaver, M. Tkalcic, and J. F. Tasic, "Database for contextual personalization," *ELEKTROTEHNISKI VESTNIK*, vol. 78, no. 5, pp. 270–274, 2011.

[143] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl, "Getting to know you: Learning new user preferences in recommender systems," in *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, (New York, NY, USA), pp. 127–134, ACM, 2002.

[144] T. N. Lillegraven and A. C. Wolden, "Design of a bayesian recommender system for tourists presenting a solution to the cold-start user problem," 2010.

[145] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowledge-Based Systems*, vol. 26, pp. 225 – 238, 2012.

[146] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, "Application of dimensionality reduction in recommender system– a case study," 2000.

[147] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 4:2–4:2, Jan. 2009.

[148] J. Yuan, W. Shalaby, M. Korayem, D. Lin, K. AlJadda, and J. Luo, "Solving cold-start problem in large-scale recommendation engines: A deep learning

approach," in *Proceedings of the IEEE International Conference on Big Data*, pp. 1901–1910, 2016.

[149] D. Lin *et al.*, "An information-theoretic definition of similarity.," in *Icml*, vol. 98, pp. 296–304, 1998.

[150] A. Sebti and A. A. Barfroush, "A new word sense similarity measure in wordnet," in *2008 International Multiconference on Computer Science and Information Technology*, pp. 369–373, Oct 2008.

[151] H.-M. Wang and G. Yu, "Personalized recommendation system k-neighbor algorithm optimization," 2015.

[152] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowl. Discov. Data*, vol. 4, pp. 1:1–1:24, Jan. 2010.

[153] Y. Shi, M. A. Larson, and A. Hanjalic, "List-wise learning to rank with matrix factorization for collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*, (Spain), pp. 269–272, 2010.

[154] F. Isinkaye, Y. Folajimi, and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261 – 273, 2015.

[155] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39–46, 2010.

[156] P. Cingolani and J. Alcalá-Fdez, "jfuzzylogic: a robust and flexible fuzzy-logic inference system language implementation," in *FUZZ-IEEE, IEEE International Conference on Fuzzy Systems*, pp. 1–8, Brisbane, Australia, 2012.

[157] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: How accuracy metrics have hurt recommender systems," in *Proceedings of the Extended Abstracts on Human Factors in Computing Systems*, (New York, USA), pp. 1097–1101, 2006.

[158] C. Porcel, A. Ching-López, G. Lefranc, V. Loia, and E. Herrera-Viedma, "Sharing notes: An academic social network based on a personalized fuzzy linguistic recommender system," *Engineering Applications of Artificial Intelligence*, vol. 75, pp. 1–10, 2018.

[159] L.-L. Wu, Y.-J. Joung, and J. Lee, "Recommendation systems and consumer satisfaction online: moderating effects of consumer product awareness," in

Proceedings of the 46 th Hawaii International Conference on System Sciences (HICSS), (HI, USA), pp. 2753–2762, 2013.

[160] C. Hayes, C. Hayes, P. Massa, P. Cunningham, P. Avesani, and P. Cunningham, "An on-line evaluation framework for recommender systems," in *In Workshop on Personalization and Recommendation in E-Commerce (Malaga*, Springer Verlag, 2002.

[161] P. V. Kannan, M. Jain, and R. Vijayaraghavan, "Apparatus and method for predicting customer behavior," Sept. 8 2015. US Patent 9,129,290.

[162] K. Lee and K. Lee, "Escaping your comfort zone: A graph-based recommender system for finding novel recommendations among relevant items," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4851–4858, 2015.

[163] Z. Huang, W. Chung, T.-H. Ong, and H. Chen, "A graph-based recommender system for digital library," in *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pp. 65–73, 2002.

[164] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," *arXiv preprint arXiv:1904.12058*, 2019.

[165] L. Zheng, C.-T. Lu, F. Jiang, J. Zhang, and P. S. Yu, "Spectral collaborative filtering," in *Proceedings of the 12th ACM conference on recommender systems*, pp. 311–319, 2018.

[166] A. Rashed, J. Grabocka, and L. Schmidt-Thieme, "Attribute-aware non-linear co-embeddings of graph features," in *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019* (T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, eds.), pp. 314–321, ACM, 2019.

[167] P. Bedi, A. Gautam, S. Bansal, and D. Bhatia, "Weighted bipartite graph model for recommender system using entropy based similarity measure," in *Intelligent Systems Technologies and Applications* (S. M. Thampi, S. Mitra, J. Mukhopadhyay, K.-C. Li, A. P. James, and S. Berretti, eds.), (Cham), pp. 163–173, Springer International Publishing, 2018.

[168] A. Geil, Y. Wang, and J. D. Owens, "Wtf, gpu! computing twitter's who-to-follow on the gpu," in *Proceedings of the Second ACM Conference on Online Social Networks*, COSN '14, (New York, NY, USA), p. 63–68, Association for Computing Machinery, 2014.

[169] B. Shams and S. Haratizadeh, "Graph-based collaborative ranking," *Expert Systems with Applications*, vol. 67, pp. 59–70, 2017.

[170] W. Yao, J. He, G. Huang, J. Cao, and Y. Zhang, "Personalized recommendation on multi-layer context graph," in *International Conference on Web Information Systems Engineering*, pp. 135–148, Springer, 2013.

[171] T. Bogers, "Movie recommendation using random walks over the contextual graph," 2010.

[172] Z. Yao, J. Wang, and Y. Han, "A tripartite graph recommendation algorithm based on item information and user preference," in *2019 6th International Conference on Systems and Informatics (ICSAI)*, pp. 1460–1465, IEEE, 2019.

[173] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 179–186, 2010.