Jointly integrating current context and social influence for improving recommendation

Meriam Bambia

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)

Cotutelle internationale avec "Université de Tunis"

Présentée et soutenue par :
Meriem BAMBIA

le mardi 13 juin 2017

Titre :

Jointly integrating current context and social influence for improving recommendation

---

École doctorale et discipline ou spécialité :

ED MITT : Image, Information, Hypermedia

Unité de recherche :

I.R.I.T.

Directeur/trice(s) de Thèse :

Mohand BOUGHANEM et Rim FAIZ

Jury :

Lynda TAMINE	Professeur, Université de Toulouse 3	Président du jury
Patrice BELLOT	Professeur, Université Aix-Marseille	Examinateur
Sylvie CALABRETTO	Professeur, Université de Lyon1	Rapporteur
Chiraz LATIRI	MCF/HDR, Université de La Manouba	Rapporteur
Mohand BOUGHANEM	Professeur, Université de Toulouse3	Directeur de thèse
Rim FAIZ	Professeur, Université de Carthage	Directrice de thèse

This thesis is dedicated to :

my parents

my sisters

my best friends
and everyone who loves me

Acknowledgments

I would like to express my heartfelt gratitude to my academic supervisors, Professors Mohand Boughanem and Rim Faiz, for the continuous academic and emotional support, their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I cant forget their hard times reviewing my thesis progress, giving me her valuable suggestions and made corrections. Their unflinching courage and conviction will always inspire me, and I hope to continue to work with her noble thoughts. My sincere gratitude goes to my professional supervisor, Nizar Menzli, for the guidance, encouragement, suggestions and valuable motivations he has provided me. His enthusiasm for research inspired me to continuously improve as a scientist. His warm welcome within Pinhole team encouraged me to realize my experiments under favorable conditions.
I would also like to thank the examiners of my thesis, Professor Lynda Tamine, University of Paul Sabatier, and Professor Patice Bellot, University of Aix-Marseille, who provided encouraging and constructive feedback. I am grateful for their thoughtful and detailed comments. I'm equally thankful to Professor Chiraz Latiri, University of Mannouba, and Professor Sylvie Calabretto, University of Lyon, for being part of the jury and for their brilliant comments and suggestions.
This thesis was funded by MobiDoc project organized by PASRI, and I would like to thank all contributors for their generous support. A heartfelt thanks to supportive colleagues at IRIT, all the members of Toulouse Institute of Computer Science Research IRIT, Mathematics, Informatics Telecommunications Toulouse Doctoral School EDMITT and University of Toulouse 3 Paul Sabatier for their instant help and kindness. Many thanks go to everyone who participated in this research study.
I would like thank my friends Imen, Ameni, and Ghada, for their support, encouragement and for those special moments that they have provided for me during vacances. A lot of thanks and gratitude to Thibaut and Rafik for their help during difficult times.
I would like to thank all my friends at IRIT lab including Mohamed, Thiziri, Hung, Diep, Lamjed, Bilel, Manel and Paul. All of you are great ! Lastly, and most importantly, I would like to thank my family for providing a loving environment for me. Father, I truly cannot thank you enough. I certainly would not be where I am today without your love and guidance since the day I was born. Thank you Mother for sitting on my shoulder and guiding me through my life, instilling me with a strong passion for learning and a great motivation to go forwards to further success. Celebrate ! Your kid has made it through the Ph.D ! Thank you sisters, I know how much you have sacrificed to help me. Wish you the best ! You believed in my dream and you helped me to realize it. Thank you infinitely, I love you so much !

Abstract

Due to the diversity of alternative contents to choose and the change of users' preferences, real-time prediction of users' preferences in certain users' circumstances becomes increasingly hard for recommender systems.
However, most existing context-aware approaches use only current time and location separately, and ignore other contextual information on which users' preferences may undoubtedly depend (e.g. weather, occasion). Furthermore, they fail to jointly consider these contextual information with social interactions between users. On the other hand, solving classic recommender problems (e.g. no seen items by a new user known as cold start problem, and no enough co-rated items with other users with similar preference as sparsity problem) is of significance importance since it is drawn by several works.
In our thesis work, we propose a context-based approach that leverages jointly current contextual information and social influence in order to improve items recommendation.
In particular, we propose a probabilistic model that aims to predict the relevance of items in respect with the user's current context. We considered several current context elements such as time, location, occasion, week day, location and weather. In order to avoid strong probabilities which leads to sparsity problem, we used Laplace smoothing technique.
On the other hand, we argue that information from social relationships has potential influence on users' preferences. Thus, we assume that social influence depends not only on friends' ratings but also on social similarity between users. We proposed a social-based model that estimates the relevance of an item in respect with the social influence around the user on the relevance of this item. The user-friend social similarity information may be established based on social interactions between users and their friends (e.g. recommendations, tags, comments). Therefore, we argue that social similarity could be integrated using a similarity measure. Social influence is then jointly integrated based on user-friend similarity measure in order to estimate users' preferences.
We conducted a comprehensive effectiveness evaluation on real dataset crawled from Pinhole social TV platform. This dataset includes viewer-video accessing history and viewers' friendship networks. In addition, we collected contextual information for each viewer-video accessing history captured by the plat form system. The platform system captures and records the last contextual information to which the viewer is faced while watching such a video.
In our evaluation, we adopt Time-aware Collaborative Filtering, Time-Dependent Profile and Social Network-aware Matrix Factorization as baseline models. The evaluation focused on two recommendation tasks. The first one is the video list recommendation task and the second one is video rating prediction task.
We evaluated the impact of each viewing context element in prediction performance. We tested the ability of our model to solve data sparsity and viewer cold start recommendation

problems. The experimental results highlighted the effectiveness of our model compared to the considered baselines. Experimental results demonstrate that our approach outperforms time-aware and social network-based approaches. In the sparsity and cold start tests, our approach returns consistently accurate predictions at different values of data sparsity.

Résumé

La diversité des contenus à recommandation et la variation des contextes des utilisateurs rendent la prédiction en temps réel des préférences des utilisateurs de plus en plus difficile à mettre en place.
Toutefois, la plupart des approches existantes n'utilisent que le temps et l'emplacement actuels séparément et ignorent d'autres informations contextuelles sur lesquelles dépendent incontestablement les préférences des utilisateurs (par exemple, la météo, l'occasion). En outre, ils ne parviennent pas à considérer conjointement ces informations contextuelles avec les interactions sociales entre les utilisateurs. D'autre part, la résolution de problèmes classiques de recommandation (par exemple, aucun programme de télévision vu par un nouvel utilisateur connu sous le nom du problème de démarrage à froid et pas assez ditems co-évalués par d'autres utilisateurs ayant des préférences similaires, connu sous le nom du problème de manque de données) est d'importance significative puisqu'ils sont attaqués par plusieurs travaux.
Dans notre travail de thèse, nous proposons un modèle probabiliste qui permet exploiter conjointement les informations contextuelles actuelles et l'influence sociale afin d'améliorer la recommandation des items.
En particulier, le modèle probabiliste vise à prédire la pertinence de contenu pour un utilisateur en fonction de son contexte actuel et de son influence sociale. Nous avons considéré plusieurs éléments du contexte actuel des utilisateurs tels que l'occasion, le jour de la semaine, la localisation et la météo. Nous avons utilisé la technique de lissage Laplace afin d'éviter les fortes probabilités.
D'autre part, nous supposons que l'information provenant des relations sociales a une influence potentielle sur les préférences des utilisateurs. Nous supposons ainsi que l'influence sociale dépend non seulement des évaluations des amis mais aussi de la similarité sociale entre les utilisateurs. Les similarités sociales utilisateur-ami peuvent tre établies en fonction des interactions sociales entre les utilisateurs et leurs amis (par exemple les recommandations, les tags, les commentaires). Nous proposons alors de prendre en compte l'influence sociale en fonction de la mesure de similarité utilisateur-ami afin d'estimer les préférences des utilisateurs.
Nous avons mené une série dexpérimentations en utilisant un ensemble de données réelles issues de la plateforme de TV sociale Pinhole. Cet ensemble de données inclut les historiques d'accès des utilisateurs-vidéos et les réseaux sociaux des téléspectateurs. En outre, nous collectons des informations contextuelles pour chaque historique d'accès utilisateur-vidéo saisi par le système de formulaire plat. Le système de la plateforme capture et enregistre les dernières informations contextuelles auxquelles le spectateur est confronté en regardant une telle vidéo.
Dans notre évaluation, nous adoptons le filtrage collaboratif axé sur le temps, le profil

dépendant du temps et la factorisation de la matrice axée sur le réseau social comme étant des modèles de référence. L'évaluation a porté sur deux tches de recommandation. La première consiste à sélectionner une liste triée de vidéos. La seconde est la tche de prédiction de la cote vidéo.
Nous avons évalué l'impact de chaque élément du contexte de visualisation dans la performance de prédiction. Nous testons ainsi la capacité de notre modèle à résoudre le problème de manque de données et le problème de recommandation de démarrage à froid du téléspectateur. Les résultats expérimentaux démontrent que notre modèle surpasse les approches de létat de lart fondées sur le facteur temps et sur les réseaux sociaux. Dans les tests des problèmes de manque de données et de démarrage à froid, notre modèle renvoie des prédictions cohérentes à pour différentes valeurs de manque de données.

List of Publications

Journal papers

1. **M. Bambia**, M. Boughanem, and R. Faiz, Jointly integrating current context and social influence for improving content recommendation. 2016 [under review].

International Conference papers

1. **M. Bambia**, M. Boughanem, and R. Faiz, Exploring current viewing context for TV contents recommendation. *IEEE/WIC/ACM International Conference on Web Intelligence*, 2016 : .
2. **M. Bambia**, R. Faiz, and M. Boughanem, Context-Awareness and Viewer Behavior Prediction in Social-TV Recommender Systems : Survey and Challenges. *In the* 19^{th} *East-European Conference on Advances in Databases and Information Systems* (ADBIS), 2015 : 52-59.
3. **M. Bambia** and R. Faiz, A Freshness Language Model for Optimizing Real-time Web Search. *In the 4th Computer Science On-line Conference*, Springer, 2015.

National Conference papers

1. **M. Bambia**, M. Boughanem, and R. Faiz, Un modèle de langue pour l'optimisation de la pertinence et la fracheur des documents Web. *INFORSID*, 2015 : 317-330.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

The explosive growth of the World Wide Web in the 1990s, and the rise of the amount of information available online outgrow the capacity of individual users to process all this information. This induces a keen interest in research fields and technology that could help manage this information overload. The most distinctive fields (Belkin & Croft 1992) are *Information Retrieval* and *Information Filtering*.

Information Retrieval (IR) (Manning et al. 2008) is a research field originated in the 1950s and is concerned with automatically fitting a user's information need against a collection of documents. IR is based on indexing data in order to respond to user queries. More specifically, the textual information retrieval stands on asking a collection of documents through queries or a set of keyword issued by a user. For instance, Google [1] is a well known Web search engine where the user formulates his needs through a query by submitting a set of keywords. These keywords are then compared to all the indexes of the documents existing in the search engine database. The 1990s realized a change from small document collections to the larger collections of pragmatic size needed to cope with the ever-growing amount of information on the Web.

From this main stream of researches and developments, a new research purpose started to be considered by the early 2000's : is it possible to predict how relevant a result returned by an IR system, before presenting it to the user, or even, before running the IR system at all ? This question has given rise to a fruitful strand of researches on performance prediction which finds additional motivation to a third type of technology.

Recommender Systems (RS) (Ricci et al. 2011), which have their derivation in the field of IR, and that were first studied as an independent research area in the 1990s, are the third type of technology designed to overcome information overload. RS are software tools and techniques providing suggestions and recommendations for items to be of use to a user.

1. https ://www.google.com/

These recommendations can help users make better decisions on choosing products or services, such as which movie to watch, which travel insurance to buy, or in which restaurant to have dinner.

The goal of a recommender system is to identify a set of items that are likely to fit the interest of a user based on a variety of information sources related to both the user and the items. RS actively predict which items the user might be interested in, and add them to information related to the user, whereas Information Filtering aims to removing items from the information stream (Hanani et al. 2001).

Over the past two decades many different recommendation algorithms have been proposed for many different domains. There are also many RS for commercial Web sites such as Amazon [2], and movie recommendation such as Netflix [3] and Movielens [4]. The value of recommendations is highlighted through their success in various areas. For instance, in 2/3 of recommended movies by Netflix are watched and, 38% more click-through are generated by Google News recommendations [5].

Traditional recommender systems, collaborative filtering (Sarwar et al. 2001, Konstan et al. 1997) and content-based (Pazzani & Billsus 2007) approaches, are considered to be the most popular and widely implemented techniques for predicting users' preferences. For a given user, collaborative methods recommend the items that users with similar preferences based on implicit data (e.g. ratings). However, content-based methods recommend the items that are similar to the ones the user preferred in the past. For example, if a user has positively rated a movie that belongs to the romance genre, the system may recommend other movies from this genre.

Later on, thanks to the popularity of social networks (e.g. Facebook [6], LinkedIn [7] and MySpace [8]), traditional RS take advantage of social information (e.g. user friendships) in order to improve recommendation effectiveness. Recommendation approaches that exploit social information, such as contacts and interactions between users are recognized as Social Filtering (SF) approaches Groh & Daubmeier (2010).

2. https ://www.amazon.com/
3. https ://www.netflix.com/
4. https ://movielens.org/
5. https ://news.google.com/
6. https ://www.facebook.com/
7. https ://www.linkedin.com/
8. https ://myspace.com/

In SF techniques, aspects and components of traditional recommenders are explicitly designed using social entities (e.g. friends). In the literature, several works (Roth et al. 2010, Mislove et al. 2007, Kumar et al. 2006) showed that social network analysis is an essential tool to obtain information of interest that allow supporting recommenders to its users. Using SF techniques has been performed in recommending tags to people (e.g. Feng & Wang (2012)), predicting social interaction (e.g. Steurer & Trattner (2013)) and recommending points-of-interest to people (e.g. Macedo et al. (2015)).
There are mainly three concepts that became central to the popularity of SF techniques : Social influence (e.g. Jamali & Ester (2009)), Trust-based (e.g. Ma et al. (2011)), and Group recommendation or groups of users around an interest (e.g. Birnkammerer & Wolfgang Woerndl (2009)).

There are certain limitations that are inherent to the recommendation problem and largely dependent on the source of information being used (e.g. **Cold start** problem where a new user has not provide enough ratings, and Grey sheep where it is more hard for the system to find good neighbors, and to recommend interesting items since there are many users with rare and unique tastes) (Cantador et al. 2008, Pazzani & Billsus 2007).

The performance evaluation of RS has been the purpose of active research in the field. The evaluation of RS must take into account the goal of the system itself (Herlocker et al. 2004). As different applications have different requirements, the system designer must decide on the imperative properties to measure for the concrete application at hand.
Since the appearance of the first recommender systems, recommendation performance has been usually equated to the accuracy of rating prediction, where estimated ratings are compared against real ratings, and differences between them are computed by means of the mean absolute error and root mean squared error metrics (Chai & Draxler 2014). In terms of the effective utility of recommendations for users, the precision or the quality of a recommended items ranking can be more important than the accuracy in predicting specific rating values (Herlocker et al. 2004).

1.1 Problem Description

The vast majority of traditional RS fail to adapt users' preferences to the changing of their situations or contexts (e.g. time and location). Considering these contextual information plays a significant role in improving recommendation, whence the notion of Context-Aware Recommender Systems (CARS).
CARS (Adomavicius & Tuzhilin 2008) is a growing research area, which deal with modeling

and predicting user tastes and preferences by incorporating available contextual information into the recommendation process. In this way, ≪ Context ≫ is a multifaceted concept that has been studied across different research disciplines, such as computer science, cognitive science, linguistics, philosophy and psychology (Adomavicius & Tuzhilin 2005).
In our work, we adopt the context definition introduced by Dey et al. (2001) : ≪ Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application ≫. The context is commonly associated with the application domain of the recommendation and each context type has a well-defined structure. For instance, if the integrated context in such a movie recommender system is the *time*, then the predicted rating assigned to a movie by a user depends on when the movie has been seen.

Under these considerations, the concept of context-awareness in RS has been studied for several years (Ricci et al. 2015, Macedo et al. 2015, Turrin et al. 2014, Hariri et al. 2014). Most of these approaches have been conceptual, where certain methods have been developed and tested on some and often limited data.
They approaches did not exploit all the current contexts to predict users' preferences. They consider only two- dimensional representation – in every case only the current time and location are considered.

However, there are several contextual information on which users' preferences undoubtedly depend (e.g. the actual weather and occasion). This is significantly important for RS in which the relevance of the items is sensitive to several contexts, and in which content-based recommendation is not accurately predicted since the content of the same item is changing daily. For example, though being interested with the whole program, a viewer might not prefer the actual content.

On the other hand, RS' users are no longer passive consumers. Thanks to Social Networks that were implemented in last few years (e.g. Facebook, Twitter [9]), users can now rate items, comment and suggest them to friends through social networks. However, due to circumstances change and the interactions that a user may experience, the user preferences depend not only on her contexts but also on the social influence around her. For instance, a user might prefer to watch world news (e.g. CNN [10] or BBC [11]) in the morning with colleagues, and movies recommended by friends on weekends.

9. https ://twitter.com/
10. www.cnn.com/
11. www.bbc.com/news

Unfortunately, most of the existing social filtering approaches (Aleksandrova et al. 2014, Liu, Cao, Zhao & Yang 2010, Porteous et al. 2010) incorporate social influence in heuristic way. They used commonly matrix factorization techniques, which fail to consider the structure in the data such as the nature of the interactions between users and the response of the user towards these interactions. They fail also to jointly integrate social influence and contextual information in one matrix factorization model (Porteous et al. 2010, Lazar & Doncescu 2009).

Another key issue is that if the recommender system is based on explicit data, each user has to rate a sufficient number of items before the system can learn the user's preferences. However, in reality, most users are reluctant to provide ratings and typically rate only a small proportion of the available items. Therefore, the dataset is sparse.
In the field of traditional RS, most of the proposed approaches (Turrin et al. 2014, Pyo et al. 2013, Chang et al. 2013) solved this problem by applying collaborative methods with latent factors, such as matrix factorization.
However, in CARS, these methods are not always effective since other recommender problems might occur (e.g. no items seen by a new user or no similar contexts exist known as "**cold start** problem", and not enough social networks related to the user known as "**social sparsity** problem").

Our work aimed to alleviate the mentioned shortcomings by proposed an approach within a recommender system in order to improve context-based recommendation. Three main problems are being addressed : - Existing context-aware approaches ignore additional contextual information on which users' preferences may depend.
- Existing CARS can not deal with **social sparsity** and **context-cold** start problems. They can not generate accurate recommendations on sparse data and new contexts.
- Existing CARS fail to achieve accurate context modeling and social influence modeling at the same time.

Consequently, the following research questions are raised :
- How to model a RS that is able to jointly integrate personalized contexts and social influence ?
- How to overcome the **data sparsity** and **user-cold start** problems in CARS ?
- What is the impact of the context and social influence in RS ?

1.2 Contributions of this Dissertation

Based on the discussion presented in the previous section, there is a need to develop more accurate and more efficient solutions for improving context-based recommendations. Several aspects need to be considered for developing these solutions. These aspects refer to the integration of current contextual information and the social influence between users into a predictive model in order to improve items recommendation.
In the following, we summarize the contributions of this dissertation, whereas the detailed contributions along with the experiments and evaluations necessary to prove them are discussed in the rest of the chapters.

1. *A context-based model for improving content recommendation :* We process contextual information extraction and through a new proposed probabilistic context-based approach (Bambia et al. 2016). This approach captures and models contextual information, and estimates the relevance of items in respect with the actual context of the user. The proposed probabilistic model integrates several context elements (i.e. occasion, time slots, location, week day and weather) in order to mine viewers' preferences in certain contextual situations and to recommend more personalized items. These additional contextual information are integrated in generic way and independently of their complex and different structures. We study the impact of the integration of each context element and evaluate its importance in the prediction performance.

2. *Social influence-based model using users interactions :* We argue that social influence can provide useful information to predict users' preferences. The aim is to model the potential effect of social relationships on user' ratings. Obviously, we assume that there is a correlation between items selected by a user and those selected by her friends (i.e. friends share some common interests) and propose to exploit these correlations for items recommendation. We present a probabilistic social-based approach that captures quantitatively social interactions between users and their friends and employs the social influence on the relevance of the items in order to mine personal users' preferences. We assume that the social influence depends not only on friends' ratings but also on social trust between users. We integrate the user-user trust measure not only social interactions between users and their friends but also the response of users towards these interactions. We study the role of social influence among viewers and their friends in improving prediction of items' relevance. We study also the effectiveness of our model with and without incorporating the trust measure.

3. *Jointly integrating the current context and social influence :* We introduce a probabilistic approach that unifies the proposed context-based model and the social-based model into

the recommendation process. The model aims to jointly integrate several contextual information and the social influence in order to improve personalized recommendation.

4. *Tackling the cold start and sparsity problems :* We are interested in considering user cold start and social sparsity problems. We propose using smoothing techniques in order to cope with strong probabilities which occurs with missing data and leads to cold start and sparsity problems.
On the one hand, we study the effectiveness of our models at various levels of data sparsity, where recommendations may get biased if there are few similar context elements (data sparsity) or if a user has a very small social network (social sparsity problem). On the other hand, we test the ability of the proposed approach to solve user cold start recommendation problem which occurs when there is no similar context with her current one (context-cold start problem) and when there is a new user with no friends or no interactions (social-cold start problem).

1.3 Thesis Organization

This thesis is organized into a set of chapter, each of which pursues a distinct research goal. Each of these goals strengthens our characterize and identify the effective contextual information for improving context-based recommendation, and enables us to build mechanisms to integrate social influence among users into recommendation prediction and to solve cold start and social sparsity problems.

In chapter 1, we present the important role of contextual information and the social realm into RS, which is the motivation behind this work. Research questions and main contributions are presented in this chapter.

In chapter 2, we present an overview of Recommender Systems. First, we introduce a brief history of the RS field. Second, we describe the most popular recommendation techniques, and discuss the most common shortcomings that the RS are suffering. Finally, we closely take a more detailed look at related work on evaluating the performance of RS.

In chapter 3, we present an overview on Context-Aware Recommender Systems. First, we discuss the general notion of context and how it can be defined and integrated in RS. Second, we define the context in different RS applications. Then, we present the classification of diverse context-aware approaches. Afterward, we introduce three different algorithmic paradigms for incorporating contextual information into the recommendation process that is contextual pre-filtering, post-filtering, and modeling. Finally, we present diverse capabilities for incorporating additional contextual information into recommendeation process and discuss its promising directions for future research.

In chapter 4, we present the proposed context-aware approach based on a probabilistic model for improving items recommendation. First, we formulate the problems and the limits related to context-aware approaches. Second, we define the basic concepts on probabilistic and language models. Afterward, we present the proposed context-based approach. Finally, we conduct a series of experiments based on real data set extracted from Pinhole platform in order to evaluate the effectiveness of our model.

In chapter 5, we present our proposed approach that unifies jointly current contextual information and social influence in order to improve items recommendation. First, we formulate the problems behind integrating the social aspect in the recommendation process. Afterward, we present the proposed probabilistic model and explain how contextual and social information are exploited. Finally, we conduct a series of experiments in order to evaluate the effectiveness of our model compared to time-based models and to test the ability of our model to cope with cold start and sparsity problem.

In chapter 6, we present the system architecture of the platform Pinhole. Then, we demonstrate how we contribute in database conception, and the transformation of data to graph-based data. Finally, we present and describe some user interfaces on Pinhole Platform.

In chapter 7, we conclude this dissertation by discussing our findings and outlining some possible directions for future work.

Chapitre 2

Overview on Recommender Systems

Contents

2.1 Introduction

RS have their derivation in the field of information filtering (Hanani et al. 2001), and are designed to overcome information overload. The goal of a recommender system is to identify sets of items that are likely to fit the interest of a certain user based on a variety of information sources related to both the user and the items. RS actively predict which items the user might be interested in and add them to the information flowing to the user to Information Filtering, whereas information filtering is aimed at removing items from the information stream (Hanani et al. 2001). Over the past two decades many different recommendation algorithms have been proposed for many different domains.

This chapter provides some basic concepts and describes some common techniques of RS. We start this chapter in Section 2.2 by introducing RS : first, a brief history of the field will be given, followed by the most popular algorithms and applications, as well as the most common shortcomings that the RS are suffering.
In Section 2.5.2, we closely we take a more detailed look at related work on evaluating the performance of RS.

2.2 Basic Concepts

In this section, we present the basic concepts of RS. We introduce the origins and the purpose of RS field. Then, we discuss the formulation of the recommendation problem in Section 2.2.2.

2.2.1 Introducing RS

A Recommender System is a computer program able to identify specific items for different user interests (see. (Resnick & Varian 1997, Adomavicius & Tuzhilin 2005, Ricci et al. 2011)).
In recent years, RS have become extremely utilized in a variety of application domains such as music, restaurants, movies, social tags and twitter pages.
RS emerged as an independent research field in the mid-1990s, when researchers and practitioners started focusing on recommendation issues that are explicitly based on ratings to predict user preferences for different items. Obviously, there are many RS for commercial Web sites such as Amazon [1] and movie recommendation such as Netflix [2] and Movielens. The value of recommendations is highlighted through their success in various areas. For instance, in 2/3 of recommended movies by Netflix are watched and 38% more click-through

1. https ://www.amazon.com/
2. https ://www.netflix.com/

are generated by Google News[3] recommendations.
A recommendation system aims to provide relevant resources to a user according to his/her preferences. It reduces user's search time by making suggestions that he/she would not have pay attention.
Particularly, the emergence and the popularity of the Web have contributed to the development of many RS in the field of e-commerce such as Amazon and CiteSeerX[4].

Initially, RS are a valuable alternative to information retrieval algorithms as they help users to discover items they might not have found by themselves. In other words, RS can be considered as a response to users who have difficulties reaching a decision when using a classic information retrieval system.

Information retrieval is based on indexing data in order to respond to user queries. More specifically, the textual information retrieval stands on asking a collection of documents through queries or a set of keyword issued by a user. For instance, in most Web search engines, the user formulates his needs through a query by submitting a set of keywords. These keywords are then compared to all the indexes of the documents existing in the search engine database.

Obviously, as reported in The Economist[5] in 2006, people read around 10 MB worth of material a day, hear 400 MB a day and see 1 MB of information every second. The consumption is raised to 74 GB a day in 2015.

In this context, the main purpose of RS is this mass of Information Filtering (Belkin & Croft 1992) transparently to the user. The recommendation process is characterized by the results list ordered according to their relevance to the user's profile that can be seen as dual to the queries issued by the user The preludes of RS arise from researches on models construction of users' preferences. These researches are issued from several areas such as information retrieval, management and marketing sciences and cognitive science.

2.2.2 Formulation of the Recommendation Problem

Several specific formulations and notations have been proposed, among which the most common formulation is the overview of Adomavicius and Tuzhilin (2005). In that work the recommendation problem is defined as follows :
Let $\mathcal{U}$ be a set of users and $\mathcal{I}$ be a set of items. Let $\mathcal{G} : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{R}$, where $\mathcal{R}$ is a totally ordered set and $\mathcal{G}(u, i)$ is the utility function that measures the gain of usefulness of item i for user u. Therefore, for each user u, we aim to choose items $i^{max,u} \in \mathcal{I}$, unknown to the user, which maximize the utility function $\mathcal{G}$, as revealed in Equation2.1 :

3. https ://news.google.com/
4. citeseerx.ist.psu.edu/
5. http ://www.economist.com/

$$\forall u \in \mathcal{U},\ i^{max,u} \ = \ arg\ max_{i \in \mathcal{I}}\ \mathcal{G}(u,i) \tag{2.1}$$

There are two main types of RS that are commonly distinguished depending on the exploited source of user preference information, and the way in which the utility function is estimated for different users :
1) Content-based RS, where suggested items are similar to those the user liked or preferred in the past ; 2) Collaborative filtering systems, in which suggested items are those liked in the past by people with similar preferences.
Recently, this classification was extended by considering social RS, i.e. systems in which suggested items are those that friends (e.g. in an online social network) liked in the past. Social RS are related but significantly different from collaborative filtering systems and will be described in details in Chapter 3.
Generally, RS are based on four main factors : the knowledge on the user (i.e. his profile according to his tastes), similarity between users (the concept of classes or user networks), knowledge on the items to recommend, knowledge of the different classes of items to recommend. The most used recommendations' types in the literature are content-based filtering and collaborative filtering that will be described in details in the following Section.

2.3 Recommendation Techniques

As mentioned above, the main goal of a recommender system is to provide users with the most relevant items according to their preferences. As shown in Figure 2.1, different strategies may be used and can be categorized based on the type of data exploited, namely content-based, collaborative filtering, and social recommendation strategies. In this section, we formalize these strategies.

2.3.1 Content-based Filtering

Content-based filtering approaches are based on the description of the items and the profiles of the users' preferences. In other words, the aim of content-based filtering approach is to recommend items that are similar to those that a user liked in the past. Particularly, candidate items are compared with items previously rated by the user and the items that match the user profile. An extensive survey of this technique can be found in (Lops et al. 2011, Pazzani & Billsus 2007).

Pandora Radio is a popular example of a content-based recommender system that plays music with similar characteristics to that of a song provided by the user as an initial seed.

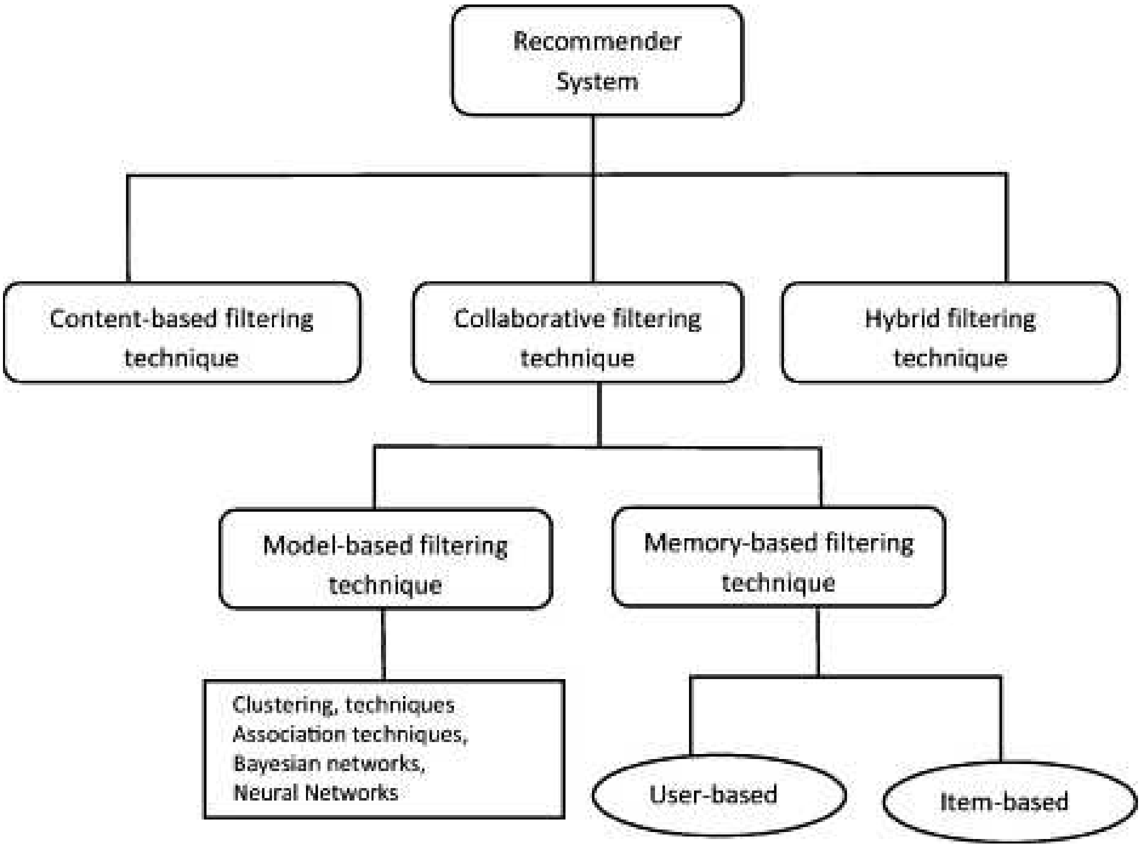

FIGURE 2.1: Recommendation techniques (Isinkaye et al. 2015)

Pandora Music Genome Project[6] uses the properties of a song or an artist (a subset of the 450 attributes to describe songs) in order to capture the essence of music with similar properties and to organize them. Users' feedbacks (likes and dislikes) are used to filter the station's results. This is an example of a content-based approach.

In content-based recommender, an item is represented by a vector of weighted terms extracted from its content. The system mostly focuses on the model of the user's preference or the history of the user's interaction with the recommender system in order to create a user profile.

Research works for content-based recommendation algorithms draw on perspectives and algorithms from various fields such as Information Retrieval, Semantic Web, and Machine Learning. For example, from Information Retrieval there are term-weighting models used for Web recommendations Balabanović & Shoham (1997), news recommendation (Lang 1995), and social tagging systems (Cantador et al. 2010). Approaches from Semantic Web technologies have also been introduced for content-based recommendations, as in the case of news recommendation (Cantador et al. 2008), or movie and music recommendations

6. https ://www.pandora.com

leveraging Linked Open Data (Ostuni et al. 2013). In Machine Learning, Mooney & Roy (2000) used Bayesian classifiers for book recommendations and Pazzani & Billsus (1997) used several techniques such as Bayesian classifiers, clustering, decision trees and artificial neural networks for Web site recommendation.

Probabilistic methods, particularly the Naïve Bayes approach generate a probabilistic model based on previously observed data (e.g, ratings). Based on Naïve Bayes model, the a posteriori probability $P(c|d)$ of document d belonging to class c, given a priori probability $P(c)$ for c, the probability of observing the document $P(d)$, and the probability of observing the document given the class $P(d|c)$ (Lops et al. 2011), is estimated as follows :

$$P(c|d) = \frac{P(c)\ P(d|c)}{P(d)} \tag{2.2}$$

In recommendation, the Naïve Bayes method is used to estimate the probability that an item is either relevant or irrelevant (class), based on the available information on each user. Therefore, items already rated are used to build the probabilities. Naïve Bayes model has been introduced in many works (Mooney & Roy 2000, Semeraro et al. 2007, De Gemmis et al. 2008, Lops et al. 2011).

Vector space models are also used in order both items and users by a set of weighted features and the similarity function used between them. Instead of using the frequency of each feature in a user/item profile, TF-IDF (Jones 1972) and BM25 (Robertson & Sparck Jones 1988) functions from the Information Retrieval field may be used. The most commonly used feature vector similarity measure is the cosine similarity (Cantador et al. 2010) :

$$sim_{dot}(d_i\ d_j) = /sumtw_{ti}\ w_{tj} \tag{2.3}$$

$$sim_{cos} = \frac{sim_{dot}(d_i\ d_j)}{\sum f \sqrt{w_{fi}^2} \sum f \sqrt{w_{fj}^2}} \tag{2.4}$$

where w_{ti} is the weight assigned to the feature t in item i.

The advantage of content-based recommendation is that the system does not require knowledge of the studied area, only the user knowledge is required. The dynamic nature of these systems is also an advantage because more users will use the system and more refined the relevance of the recommended items will be.

However, items that have not been judged similar to those appreciated by the user will not be recommended. This poses the overspecialization problem or the thematic redundancy of recommendations submitted to the user. Indeed, if a user is interested only in political news, news about sports' events will never be recommended. To counter this problem, Sheth & Maes (1993) discussed genetic algorithms based on classification algorithms allowing pseudorandom recommendations.

Similarly, a user who has never used the system will not have relevant recommendations because of the lack of information. To resolve this problem, Card et al. (1991) proposed a number of heuristics based on the assumption that recommendations are offered only if enough information were collected.

2.3.2 Collaborative-Filtering

Collaborative filtering approaches rely on collecting and analyzing a large amount of users' behaviors, activities or preferences and predicting users' future preferences based on their similarity to other users. The collaborative filtering approaches do not rely on items' contents and therefore they are able to recommend accurately complex items such as videos. Undoubtedly, the main advantage of the collaborative filtering RS is the active or the passive involvement of the system users. Indeed, recommending relevant items to a user based on their preferences appears naturally easier.

The algorithm popularized by *Amazon.com*'s recommender system represents one of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y). Other example includes *Last.fm* which is a music website that recommends songs by observing the bands and individual tracks to which the user has listened on a regular basis and comparing those against the listening behavior of other users. *Last.fm* will play tracks that are frequently played by other users with similar interests. The approach of *Last.fm* is an example of a collaborative filtering technique since it leverages the behavior of users. *Facebook*[7], *LinkedIn* and *MySpace*[8] use collaborative filtering approaches to recommend new friends, groups, and other social connections by examining the connections network between a user and his friends. There are also several innovative approaches on collaborative filtering applications such as (Terry 1993) and (Harman 1994), the *MovieLens*[9] recommendation system of movies, *Throw*[10] system which recommends practical jokes, and finally FlyCasting system recommending online radio (Hauver & French 23-24 Nov. 2001).

When building a predictive model based on a user's behavior, a distinction is often made between explicit and implicit forms of data collection. Explicit data collection includes user ratings and user rating of items collection. Implicit data collection is based on the items that

7. https ://facebook.com/
8. https ://myspace.com/
9. https ://movielens.org/
10. http ://www.jokes.monigo.com

a user use in an online store, by analyzing item/user times and discovering similar likes and dislikes. In other words, the recommender system calculates a list of recommended items for the user based on comparison of the collected data with similar and dissimilar data.

Based on the form of the inputs, we distinguish two types of collaborative filtering systems : systems that exploit explicit user ratings (rating-based systems), and systems that exploit implicit user preference information (log-based systems). The rating assigned to an item by a particular user is typically interpreted as the true utility of that item for the user. There are systems, however, where no explicit ratings are available, but where user interests can be inferred from implicit feedback information. In order to provide item recommendations in such systems, two plausible approaches do exist : 1) directly exploiting implicit preference data (Linden et al. 2003, Das et al. 2007, Wang, Robertson, de Vries & Reinders 2008), and 2) transforming implicit preference data into explicit ratings to be exploited by standard CF strategies (Celma & Herrera 2008).

In the literature, collaborative filtering algorithms can be themselves classified into two types : Memory-based and Model-based methods :

2.3.2.1 Memory-based

methods are characterized by their simplicity, easiness of implementation, immediate incorporation of new data and comprehensibility of results since minimal or no learning phase is involved. However, memory-based methods may suffer from scalability issues and lack of sensitivity to sparse data.
The the most popular memory-based approaches are the *Neighborhood* models. Data normalization, neighbor selection, and determination of interpolation weights represent the three major components that characterize neighborhood approaches.
The original form of neighborhood model is *user-based* model (Herlocker et al. 1999). User-based methods estimate unknown ratings based on recorded ratings of similar users. They generate recommendations for a user u by scoring the items in the profiles of the neighbors as a sum over the preference values assigned by the neighbors weighted by the similarity to the target user :

$$s_{UB}\ (u,i)\ =\ \sum v \in N(u) sim(u,v)\ r_{v,i} \tag{2.5}$$

where $sim(u,v)$ is the similarity value between users and $N(u)$ denotes the set of neighbors of user u.

Later, an analogous *item-based* approach was proposed by Linden et al. (2003), Sarwar et al. (2001), where a rating is estimated using known ratings of the same user on similar items. As highlighted by many works (Bell & Koren 2007, Sarwar et al. 2001, Takács et al. 2007), the improved accuracy and the significant scalability make the item-based approach more convenient. In the item-based methods, similarities between items with common users are exploited. The idea is that items that are similar to those that the user has already rated or consumed are good candidates for recommendation. In other words, items similar to those of the profile I_u of the user u is scored as the sum of their item-to-item similarities weighted by the preferences of u :

$$s_{IB}\ (u,i)\ =\ \sum j \in I(u) sim(i,j)\ r_{u,j} \tag{2.6}$$

where $sim(u,v)$ is the similarity value between users and $N(u)$ denotes the set of neighbors of user u.

The success of neighborhood methods relies on the choice of the interpolation weights, which are used to estimate unknown ratings based on neighboring known ones. However, most neighborhood methods require a rigorous way to derive interpolation weights. Latent factor models generally offer high expressive ability to describe various properties of the data. Therefore, they provide more accurate results than neighborhood models. However, most commercial systems (e.g. Amazon (Linden et al. 2003) and TiVo (Ali & Van Stam 2004)) are based on the neighborhood models due to their relative simplicity.

2.3.2.2 Model-based methods

take a different way to exploit collaborative filtering data. The algorithms of model-based methods depend on a learning phase, in which a predictive model of user preferences is built based on the observed data.
These methods are inspired in machine learning techniques such as Bayesian networks (Breese et al., 1998), clustering (Ungar and Foster, 1998), artificial neural networks (Salakhutdinov et al., 2007) and latent factor models (Blei et al., 2003 ; Hofmann, 2004 ; Koren et al., 2009). Latent factor models are the most studied and prevalent model-based techniques. These techniques perform a dimensionality reduction of the rating matrix R and use a set of latent variables in order to explain user preferences for recommendation purposes. Some other techniques include matrix factorization (Koren et al., 2009), Latent Dirichlet Allocation (Blei et al., 2003) and probabilistic Latent Semantic analysis (Hofmann, 2004).

2.3.2.3 Matrix factorization

Matrix factorization models have acquired popularity through their attractive accuracy and scalability. There are many different matrix decompositions techniques known as Singular Value Decomposition-based models. Each technique finds its use among a particular class of problems. For example, conventional SVD is defined when knowledge about the matrix

is complete.
The intuition behind matrix factorization techniques is to learn latent features that determine how user rates an item. The existing ratings can be represented in a matrix R of size $|U| \times |V|$, where U is the set of users and V is the set of items. The aim is to discover K latent features u_i^T and v_j which correspond to the i-th column and the u-th column of U and V, respectively. Then, the task is to find two matrices P of size $|U| \times K$ and Q of size $|V| \times K$ such that their product approximates R :

$$R \approx P \times Q^T = \widehat{R} \tag{2.7}$$

Based on this parametrization, the predicted rating is computed as follows :

$$\widehat{r}_{ui} = u_i^T \times v_j \tag{2.8}$$

The parameters u_i^T and v_j are learned based on a certain loss function in order to minimize iteratively the difference between their product and the matrix R.
In information retrieval, Singular Value Decomposition (SVD) is well established in order to identify latent semantic factors (Deerwester et al. 1990). Nevertheless, using SVD on explicit ratings in the Collaborative Filtering domain raises difficulties due to existing missing values.
Earlier works is built based on imputation (Kim & Yum 2005, Sarwar et al. 2000*b*), which replaces missing ratings and makes the rating matrix dense.
However, since it significantly increases the amount of data, imputation can be very expensive. Moreover, the data may be significantly imprecise due to inaccurate imputation. Hence, several recent works (Bell et al. 2007, Canny 2002, Koren 2008, Paterek 2007, Salakhutdinov et al. 2007, Takács et al. 2007) suggested to model only the observed ratings, while avoiding overfitting based on an adequate regularized model.
Unfortunately, despite the fact that matrix factorization techniques are commonly used by almost works, they are considered as the most complex techniques. This is due to their major drawback related to the non-convexity scheme. As a result, there is in general no algorithm that is guaranteed to compute the desired factorization. In addition, matrix factorization techniques fail to consider the structure in the data such as relationships between users.

2.3.3 Hybrid Approaches

Each of collaborative and content-based techniques has its own weaknesses, such as the well known cold-start problem where new users have few ratings. In this context, hybrid methods (Burke 2002, Adomavicius & Tuzhilin 2005) have been proposed to avoid the limitations of collaborative filtering and content-based algorithms instead of using them separately. As defined by Burke (2002) and Schein et al. (2002), a hybrid recommender system associates multiple techniques together in order to achieve some synergy between them.

Adomavicius et al. (2005) classified hybrid recommendation approaches as follows :

— Combining separate recommendations : the predictions of separate recommendation algorithms are combined to provide a single recommendation, using methods such as linear combinations (Claypool et al. 1999).

— Adding content-based characteristics to collaborative filtering : Pazzani & Billsus (1997) adapted the user-based method to calculate similarities based on content-based user profiles.

— Adding collaborative characteristics to content-based methods : latent factor models can be applied to content-based approaches for text recommendation.

— Developing a single unifying recommendation model : Popescul et al. (2001) and Schein et al. (2002) proposed a unified probabilistic method for combining collaborative and content-based recommendations.

Netflix is a good example of the use of hybrid RS. They make recommendations by comparing the watching and searching behavior of similar users (i.e. collaborative filtering) as well as by offering movies that share characteristics with movies that a user has rated highly (content-based filtering).

In the literature, several studies (Claypool et al. 1999, Basu et al. 2011, Popescul et al. 2001, Schein et al. 2002, Kim et al. 2006) compare the performance of hybrid RS with the pure collaborative and content-based methods and demonstrate that the hybrid methods can provide more accurate recommendations than pure methods. Generally, they are used to overcome some of the common problems in RS such as cold start and the sparsity problem.

2.3.4 Social-based Filtering

≪ Social ≫ is considered as the knowledge about the larger community of users other than the target user and the set users' profiles stored in a system. Social networks are characterized by their users that can actively expose their interests and personal data. From the social network sites they are encouraged to seek out other users and identify them as "friends", their own "family" or "friend group".

With the popularity of social networks, traditional RS take advantage of social information (e.g. user friendships and social influence) in order to improve recommendation effectiveness. Recently, exploiting social information is becoming one of the strongest areas where experts are currently working. In the literature, several works (Roth et al. 2010, Mislove et al.

2007, Kumar et al. 2006) showed that social network analysis is an essential tool to obtain information of interest that allow supporting recommenders to its users.

For instance, in the days following the announcement of the results of the presidential elections, on November 8 2016, the polemic has swollen : how could most of the pollsters and the journalists underestimate the number of voters for Donald Trump ?
Social networks, Facebook coming first, have been blamed and accused for having locked up many users in a "Personal Information Crowd". Each user has only seen content close to his/her ideas, leading him to ignore the existence of other people with opposing opinions. These "Crowd" are created by the filtering techniques and recommendation algorithms put in place in order to select the contents shared on the social network.

Recommendation approaches that exploit social information, such as contacts and interactions between users are recognized as Social Filtering approaches. In social filtering techniques, aspects and components of traditional recommenders are explicitly designed using social entities and social contexts. One important variant of social filtering (Groh & Daubmeier 2010) is based on substituting the user-neighborhood, whose ratings are considered to be similar to the current user's tastes. These techniques are also named community-based.

Simpler algorithms, referred as ≪ pure ≫ social recommenders, have been proposed in (Liu & Lee 2010). The authors proposed an adaptation of the user-based collaborative filtering technique, where the set of nearest neighbors is replaced by the set of (explicit) friends of the target user. That is :

$$N_k(u,i) = \{v \in U : v\ is\ friend\ of\ u\}(2.9)$$

Community-based systems recommend items based on the preferences of the users' friends, in which the search activities of communities of like-minded users are used to increase the results of a mainstream search engine and to provide a more focused community-oriented result list (Smyth et al. 2005, 2004). This technique is based on the epigram ≪ Tell me who your friends are, and I will tell you who you are ≫ (Budzik & Hammond 2000, Champin et al. 2010).

Generally, Social Filtering is associated with the integration of an underlying social network into recommender system prediction models. Relations between viewers or between viewers and items can be exploited together with context approaches for recommending TV programs. The former study of Groh et al. (2012) showed that social filtering approaches work very well in taste related domains by focusing on the significance of the social context.

Other studies showed also that in taste domains, users' preferences are influenced by their social environment. This is mainly due to the fact that users trust recommendations made by people they know such as their friends (Groh et al. 2012) and (Groh & Ehmig 2007). For instance, while watching TV with a group of friends, some proposed recommendation will be executed immediately. These social recommendations may be considered as a significant source to enrich the viewing experience and predict his preferences.

The basic assumption is that users tend to rely more on recommendations from their friends than on recommendations from similar but anonymous individuals (Sinha & Swearingen 2001). The recommendation relies commonly on ratings that were provided by the user's friends. It follows the rise of social-networks and enables a simple and comprehensive acquisition of data on social relations of the users.

Recommendations by social filtering approaches have the interesting property that they are generally easier to explain than user-based collaborative filtering approaches. Golbeck (2006), Massa & Avesani (2004) reported that in general, social-network based recommendations are no more accurate than those derived from traditional collaborative filtering approaches. However, they showed that using social information is accurate when user ratings of a specific item are highly varied or where the users did not provide enough ratings to compute similarity to other users. Groh & Ehmig (2007), Guy et al. (2009) have showed that in some cases social-network data yields better recommendations than profile similarity data and helps dealing with the cold start problem which improves recommendation results.

Nowadays, useful relationships between users can be found virtually everywhere such as in social networking sites (e.g. Facebook, LinkedIn and MySpace). Several studies (Ardissono et al. 2003, Bolger et al. 2003, Balabanović & Shoham 1997, Papadogiorgaki et al. 2007, Bernhaupt et al. 2008, Bonnefoy et al. 2007) qualified an important consideration on whether the use of the system is usually carried out alone or with other people.

There are mainly three concepts that became central to the popularity of social networking these social influence, trust and groups of users around an interest.

2.3.4.1 Group Recommendation

Another field of Social Recommenders is *Group recommendation* which not only has to take into account a single user's preferences but those of a whole group e.g. fairness. In this case, all members should be treated equally when making a recommendation.

Several works (Bar & Glinansky 2004, Birnkammerer & Wolfgang Woerndl 2009, Wörndl et al. 2009, Jameson 2004*a*, Jameson et al. 2004, Jameson & Smyth 2007, Masthoff 2004, O'Connor et al. 2001, Crossen et al. 2002, Pennock et al. 2000) have been proposed in the field of group recommendation. Arias et al. (2012) highlighted the performance of Social

Filtering compared to traditional CF approaches in a taste related domain (e.g. tastes in clothing, TV, cinema and music) which are strongly influenced by friends. They also proved that Social Filtering is a valuable source for recommendations in certain settings but may pose problems for others (e.g. domains where the social network data is too sparse are less well suited for social filtering). Ma (2013) performed an experimental study on implicit user and item social relationships. They developed a regularization matrix factorization method to employ the similar and dissimilar relationships between users and/or items. The similarity between items is measured using Pearson's correlation similarity. However, they focus on the role of implicit information and ignore the importance of the influences between users and between items.

Another aspect that occurs when making recommendations for groups is that a user's preferences change according to the presence of other group members. For instance, a user might like horror movies with his friends whereas he might prefer comedy when watching TV with his family. An example for a group recommendation is a FIT (Family Interactive TV) program recommender outlined by (Bar & Glinansky 2004) which takes into account the change of preferences. As shown in Figure 2.2, the recommendation process of FIT consists of three main components : User profile construction ; Prediction ; and Adaptation. Birnkammerer & Wolfgang Woerndl (2009) treated the group recommendations as sequences of recommendation listed two important dimensions of group recommendation : the Number of recommendations per group and type of group. O'Connor et al. (2001) differentiates if a group is ephemeral or persistent and if it is public or private (with respect to privacy concerns).

MOVIELENS which recommends movies based on an individual's taste as inferred from ratings and social filtering. POLYLENS (Felfernig 2005), a group recommender extension of MOVIELENS, allows users to create groups and ask for group recommendations.

2.3.4.2 Social Influence

The basic idea behind social influence is that a user's friends may share common interests with the user, and have influence on the user's decisions.
In the literature, social influence is incorporated in the recommendation process in various ways. Jamali & Ester (2009), Konstas et al. (2009) employed the random walk approach proposed by Tong et al. (2006) in order to incorporate user's social network for item recommendation. On the other hand, Ma, King & Lyu (2009), Ma, Lyu & King (2009), Ma et al. (2011) extended model-based systems to include social influence. They proposed to integrate users' social trust network into their models through a linear combination or as a regularization term.

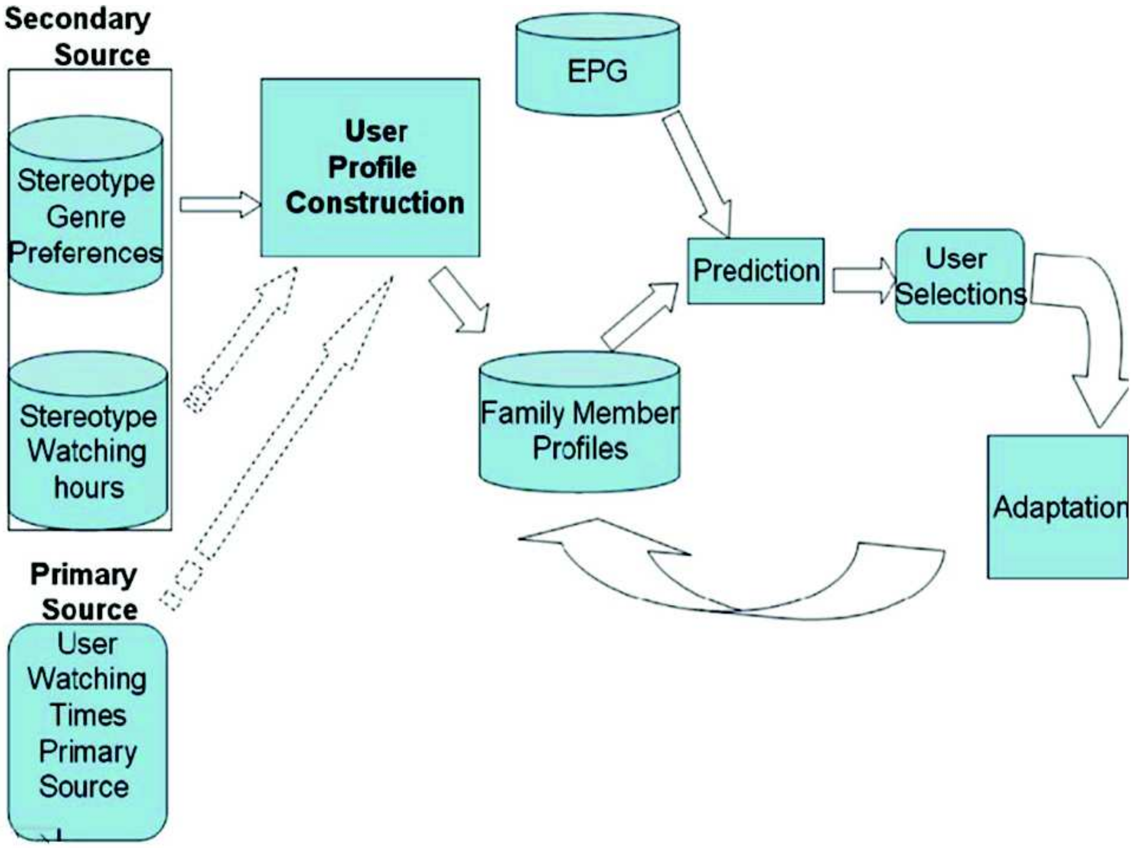

Figure 2.2: FIT architecture (Bar & Glinansky 2004)

One has to consider maximizing synergy effects among users (Zhu et al. 2011, Brocco, Groh & Forster 2010, Brocco & Groh 2009) and therefore will have to consider aspects like social influences (Brocco, Groh & Kern 2010).
Crandall et al. (2008) addressed the problem to determine a neighborhood that properly weighs both the profile's likeness and the trust between users and established the right balance between similarity and social influence.
Groh (2007) outlined several conclusions may be drawn in social influence among users :

- Virtual friend-relationships may be capable of providing similar ratings,

- Binary friend-relations on average show more rating similarity than disconnected pairs,

- Cliques and friend-pairs might are considered as important recommendation source since they share a common taste regarding the investigated domain.

Liu, Cao, Zhao & Yang (2010) proposed a social network-based movie recommendation technique. They used both collective matrix factorization and regularized matrix factorization. As described in Section2.3.2.3, the matrix factorization technique aims to factorize the rating matrix R ($R = U^T.V$), where U is the set of users and V is the set of items and

the latent features u_i^T and v_j correspond to the i-th column and the u-th column of U and V, respectively.
In Liu, Cao, Zhao & Yang (2010), the collective matrix factorization is used to jointly factorize the rating matrix R ($R = U^T.V$) and the binary matrix $n \times n$ G ($R = Z^T.V$), where the factor matrix V is being shared by the two matrices. Based on the following training objective function :

$$\begin{aligned} \min_{U,V,Z} \sum_{i,u}(r_{ui} - u_i^T.v_u)^2 + \alpha \sum_{j,i}(g_{uj} - z_j^T.v_u)^2 \\ +\lambda(||U||_F^2 + ||V||_F^2 + ||Z||_F^2) \end{aligned} \tag{2.10}$$

α is a parameter used to control the importance of the matrix G and $g_{uj} = 1$ if u and j are friends,.

On the other hand, the regularized matrix factorization is used to incorporate similarity between users and their friends based on the following network regularized matrix factorization function :

$$\begin{aligned} \min_{U,V} \sum_{i=1}^{m}\sum_{u=1}^{n}(r_{ui} - u_i^T.v_u)^2 \\ +\lambda_1(||U||_F^2 + ||V||_F^2) + \sum_{i,j} S_{i,j}||v_i - v_j||_F^2 \end{aligned} \tag{2.11}$$

$S_{i,j}$ denotes the similarity between user i and j, and equals to 1 if users i and j are friends, else it equals to 0.

Even though social influence and user similarity come with totally different mechanism, some might confuse them. Obviously, user similarity is content-dependent. Two users with similar preferences would buy same books on their own choice independently. However, influence is content-free. The influence among users is based on social relation rather than item content. Given that a user can affect his friend's decision, they are not necessarily similar in interests.

The impressive expansion of social media and social networking systems, social influence from friends presents new opportunities for RS but also brings many great challenges.

Pálovics et al. (2014) proposed a matrix factorization method to model social influence between users and their friends for music recommendation. Social influence is modeled using common preferences observed close together in time by a user and his friends. Chaney et al. (2015) developed a Bayesian method based on Poisson factorization model that captures latent user preferences and latent item attributes and estimates the influence of the observed friends' clicks on user preferences.

Other approaches have been proposed to incorporate the social relationships into predictive models : Huang et al. (2010) presented a method to calculate the utility of a social recommendation based on three factors, (i.e. receiver interest, item quality and interpersonal influence between the sender and receiver user). In this case, interpersonal influence is not aimed at measuring similarity between users, but is considered as the power of influence of one user over another, as presented in Fasli (2006). Ye et al. (2012) proposed a probabilistic generative model by integrated social influence, user behavior and item content for item recommendation and group recommendation. They assume that users rarely followed their friends' uncommon opinions. Additionally, a group does not always consist of friends, the strength of influence between a user and one friend was not correlated with their similarity. However, when the group is large, the strong assumption of pairwise influence in a group may not be true.

2.3.4.3 Trust-based Recommendation

Even though social relationships and trust relationships do not model exactly the same concept, trust-based recommendation approaches are considered as a different way for integrating social information into a recommendation process (Ma et al. 2011). In contrast to other approaches, trust-aware recommenders make use of trust networks where users express a level of trust on other users (Massa & Avesani 2007).

These recommenders acquire a trust network and a trust metric, so that trustworthiness of every user can be estimated. A plausible trust network must be inferred, depending on the available data, based on the information we already know about users (e.g. social interactions among users or explicit trust relations).

Recently, there are a few works focusing on incorporating social trust among users into RS. However, most of them considered a single type of trust between users or uses observed boundary, such as categories of items, to identify multi-faceted trust. Malinowski et al. (2005) proposed a trust approach and extended the model of Keim et al. (2003), by incorporating trust into the recommender-based approach in order to integrate relational information. Richardson et al. (2006) assumed that a single value can be used to express trust. Based on the assumption that if user u has trust in user f, it is not necessary that user f has also trust in u, they represented relations between user in a weighted directed graph where the weights of the edges express trust.

Trust-based RS (Golbeck 2006, O'Donovan & Smyth 2005) still operate on the core rating prediction problem but use trust relationships, since they exploit the trust relationships found in these social networking sites to build new recommendation algorithms. The main claimed advantage is that the mutual trust of users can be exploited also for increasing the trust in the system. Zhao et al. (2013) proposed an algorithm based on probabilistic matrix factorization to mine topics from tags on the items and to estimate the trust between users and their friends on specific topics. Forsati et al. (2015) introduced a collaborative social ranking model based on matrix regularization method to keep the latent vector of each user similar to his trusted neighbors in the social network. They proposed also an algorithm named PushTrust to simultaneously leverage trust, distrust and neutral relations between users.

2.4 General Limitations of RS

In Section 2.3, we noted the main characteristics of each recommendation technique, which are basically dependent on the source of information being used. However, each recommendation technique has strengths and weaknesses.

In this section, we analyze the main limitations of each recommendation technique. Even though hybrid recommendation techniques would overcome the problems of the combined techniques, there are certain limitations that are inherent to the recommendation problem. Thus, each problem has to be addressed independently. In addition, additional problems, along with more limitations, arise when combining different methods. As highlighted by (Adomavicius & Tuzhilin 2005, Pazzani & Billsus 2007, Cantador et al. 2008), the main limitations of content-based filtering approaches are the following :

- Restricted content analysis : Content-based recommendations depend on the available features explicitly associated with the items. These features should be in a form that can be automatically parsed by a computer or manually extracted. Their extraction depends on the domain and could be unfeasible or very difficult to maintain.

- User cold start problem (New user) : A user must emit some preferences (or ratings) for a sufficient number of items before a recommender can build a reliable user profile.

- Overspecialization : Given that content-based recommenders only retrieve items similar to those the user has already rated, recommended items are very similar, are most likely to be known by the user and provide little (or none) novelty from the user perspective.

- Portfolio effect : As a consequence of the previous limitation, the recommended items are often very similar, which leads to a set of insufficiently diverse or too redundant item suggestions.

Collaborative filtering techniques often suffer from three main problems :

- Sparsity problem : The number of items that might be recommended is extremely large. The most active users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings.

- Gray sheep : As collaborative recommendations are based on the tastes of similar users to suggest new items, when a user has very specific preferences, it will be more difficult for the system to find good neighbors, and thus, to recommend interesting items.

- Item cold start problem (New item) : As a new item has not been rated by a considerable number of users, a recommender system may not be able to recommend it. Thus, popular items tend to have advantage in this kind of systems.

- User cold start problem (New user) : As a new user has not provide enough ratings, the system is unable to recommend her relevant un-known items.

A common characteristic of the data sparsity and cold-start problems is that the small number of commonly rated items between users makes it difficult to accurately predict user similarity. Because of data sparsity, there is even no commonly rated items between two users, causing their similarity not computable.

Social filtering approaches have also their own limitations :

- Social sparsity : In order to produce recommendations, social filtering techniques require at least one contact in the social network connected to every user. This is not a typical situation for most of the users in a system.

- New social connection : Recommendations may get biased if a user has a very small social network or if she has only one connection. Therefore, every social recommendation would be generated based on the activity of just one user.

- Social similarity : As shown by (Ziegler & Lausen 2004), the fact that two users share such a connection in a social network probably means that these users have similar interests. However, the misuse of this similarity may lead to bad recommendations, even though the user's experience may be improved in terms of diversity and serendipity.

As a summary, Table 2.1 draws a comparison of the limitations of the three types of recommendation techniques described above.

2.5 Performance Evaluation of RS

The performance evaluation of RS has been the purpose of active research in the field. Since the advent of the first RS, recommendation performance has been usually assimilated to the accuracy of rating prediction and the effective utility of recommendations for users.

Problem	**CBF**	**CF**	**SF**
Restricted content analysis	Yes	No	No
Overspecialization	Yes	No	No
Portfolio effect	Yes	No	No
New user	Yes	Yes	No
New item	No	Yes	No
Grey sheep	No	Yes	No
Rating data sparsity	No	Yes	No
Social sparsity	No	No	Yes
New social connection	No	No	Yes
Social similarity	No	No	Yes

Table 2.1: List of limitations in Content-based filtering (CBF), Collaborative filtering (CF), and Social filtering (SF) systems

In this section, we review the process of evaluating a recommendation system. We discuss three different types of experiments (offline, user studies and online experiments). We represent several measures of accuracy evaluation in RS.

2.5.1 Experimental Settings

In this subsection, we describe three levels (offline, user studies and online experiments) of experiments that can be used in order to compare several recommender approaches. Generally, it is important to follow a few basic guidelines in all experimental studies :

- Hypothesis : a hypothesis must be formed before running the experiments. This hypothesis must be concise and restrictive. An experiment must be designed to test this hypothesis.

- Controlling variables : It is important that all not tested variables will stay fixed when comparing a few candidate algorithms on a certain hypothesis.

- Generalization : We must hold conclusions on the deployed system, and generalize beyond the experimental data set when choosing an algorithm for a real application. We must typically experiment with several data sets or applications in order to increase the probability of results generalization.

2.5.1.1 Offline Evaluation

An offline experiment is realized by using a pre-collected data set of items chosen or rated by users. The behaviors of users that interact with a recommendation system are simulated by using the pre-collected data. Offline experiments are attractive because they do not require interactions with real users, and thus allow comparing a wide range of candidate algorithms at a low cost. The shortcoming of offline experiments is that they can answer a very small set of questions, typically about the prediction power of an algorithm.

Consequently, the aim of the offline experiments is to filter out inappropriate approaches, leaving a relatively small set of candidate algorithms to be tested by the more costly user studies or online experiments. Therefore, the used data should match as closely as possible the data the designer expects the recommender system to face when deployed online.

It is necessary to simulate the online process where the system makes predictions, and the user corrects the predictions in order to evaluate algorithms offline. This is usually realized by recording historical user data, and then hiding some of her interactions in order to simulate the knowledge of how a user will rate an item.

This makes some assumptions concerning the behavior of users, which could be considered as a user-modeling for the specific application. User-modeling is a difficult task. There is a vast amount of research on the subject (Fischer 2001). In addition, we may optimize a system whose performance in simulation has no correlation with its performance in practice, when the user model is inaccurate.

2.5.1.2 Online Experiments

In many recommendation applications, the designer of the system aims to influence the behavior of users. Therefore, we are interested in measuring the change in user behavior when interacting with different RS. For instance, if some utility gathered from users of one system exceeds utility gathered from users of other systems, then we can conclude that one system is superior to the others.
The real outcome of a recommendation system depends on several factors such as the user's intent (e.g. how specific their information needs are), the user's context (e.g. what items they are already familiar with ? How much they trust the system ?), and the interface through which the recommendations are exposed. Obviously, Kohavi et al. (2009) employed an online testing system. In general, online evaluations are distinctive in that they allow direct measurement of system goals, such as users' retentions.
However, it can be difficult to gain a complete understanding of system properties given that varying such properties independently is difficult, and comparing many algorithms through online trials is expensive.

2.5.2 Evaluation Metrics of RS

The evaluation of RS must take into account the goal of the system itself (Herlocker et al. 2004). As different applications have different requirements, the system designer must decide

on the imperative properties to measure for the concrete application at hand. We must understand and evaluate the trade-offs of some properties and their effect on the overall performance of the system.
In this section, we survey some of the properties that are commonly regarded when deciding which recommendation approach to select.

Prediction accuracy Prediction accuracy is the most discussed property in the RS literature. The majority of RS are based on a prediction module that may predict user opinions over items (e.g. ratings on items) or the usage probability (e.g. purchase).
The basic assumption in RS is that the more a system will provide accurate predictions the more it will be preferred by the user. Therefore, several researchers set out to find algorithms that improve predictions.
In some applications, such as the popular Netflix DVD rental service, the goal is to predict the rating a user would give to an item. In other cases, the aim is to measure the accuracy of the system's predicted ratings.
The mean absolute error (MAE) is a quantity commonly used to measure how close predictions are to the eventual outcomes. It measures the average magnitude of the errors in a set of forecasts without considering their direction. It measures accuracy for continuous variables. The MAE is the average over the verification sample of the absolute values of the differences between forecast and the corresponding observation. It is a linear score which means that all the individual differences are weighted equally in the average. The mean absolute error is given by the following equation :

$$MAE = \frac{1}{N}\sum_{v\in N} |q_{uv} - p_{uv}| \tag{2.12}$$

Where q_{uv} is the real rating of user u for item v, p_{uv} is the predicted rating of viewer u for item v and N is the number of recommended items.

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values (e.g. ratings) predicted by a model or an estimator and the values actually observed (e.g. real ratings). The RMSE represents the sample standard deviation of the differences between predicted values and observed values. It represents a quadratic scoring rule which measures the average magnitude of the error. The difference between forecast and corresponding observed values are each squared and then averaged over the sample. The RMSE gives a relatively high weight to large errors, since the errors are squared before they are averaged. This means that the RMSE is most useful when large errors are particularly undesirable. The RMSE is defined as the square root of the mean square error :

$$RMSE = \sqrt{\frac{1}{N}\sum_{v \in N}(q_{uv} - p_{uv})^2} \tag{2.13}$$

Where q_{uv} is the real rating of user u for item v, p_{uv} is the predicted rating of viewer u for item v and N is the number of recommended items.
The MAE and the $RMSE$ can be used together to diagnose the variation in the errors in a set of forecasts (Chai & Draxler 2014).

Normalized MAE (NMAE) and Normalized RMSE (NMRSE) are versions of MAE and RMSE that have been normalized by the range of the ratings (i.e. the maximum rating mines the minimum rating). The resulting ranking of algorithms is the same as the ranking given by the unnormalized measures, as they are simply scaled versions of MAE and RMSE.

In many applications, a recommendation system does not predict the rating a user would give to an item, but attempts to recommend to users items that they may use.
In an offline evaluation of usage prediction, the data consisting of items each user has used is typically collected. Then, we select a test user, hide some of her selections, and ask the recommender to predict a set of items that may be used by the targeted user.
The usage prediction can be measured based on the following quantities :

$$Precision = \frac{\sum_{i=1} Nprecision_i}{N} \tag{2.14}$$

where N is the number of recommended items and :

$$precision_i = \begin{cases} 1, & \text{if } Used\ Item\ I\ \in\ Recommended\ Item\ set \\ 0, & \text{if } Used\ Item\ I\ \notin\ Recommended\ Item\ set \end{cases} \tag{2.15}$$

$$Recall = \frac{|relevant\ and\ recommended\ items|}{|relevant\ items|} \tag{2.16}$$

The most useful measure of interest, when the number of recommendations that can be presented to the user is preordained, is Precision at N.

Other metrics Since applications have different needs, additional characteristics of recommendations could be taken into consideration. Thus, alternative metrics further than accuracy and precision may be measured (Shani & Gunawardana 2011). For example, probably due to data sparsity, some algorithms may provide recommendations with high accuracy, but only for a small amount of users or items.

This effect can be quantified by measuring Coverage, novelty and diversity.

- Coverage :
Two types of coverage can be defined :
- user coverage (proportion of users to whom the system can recommend items) ; and
- item or catalog coverage (proportion of items the system can recommend).

Shani & Gunawardana (2011) proposed two metrics for measuring item coverage. The first one is based on the Gini's index, and the other one is based on Shannon's entropy.
Ge et al. (2010) proposed simple ratio quantities in order to measure such metrics, and to discriminate between the percentage of the items for which the system is able to generate a recommendation (prediction coverage), and the percentage of the available items that are effectively ever recommended (catalog coverage). A similar distinction is highlighted by Herlocker et al. (2004) and Salter & Antonopoulos (2006). Herlocker et al. (2004) considered that item coverage is mainly important for the tasks of find all good items and annotation in context. Furthermore, a system with low coverage is expected to be less valuable to users. Therefore, the authors proposed the combination of coverage with accuracy measures to yield an overall "practical accuracy" measure for the system. In such a way, the coverage is raised only because recommenders produce counterfeit predictions.

Another measure of catalog coverage is the sales diversity (Fleder & Hosanagar 2007), which measures how unequally different items are chosen by users when a particular recommender system is used.
Recently, two recommendation metrics have become very popular : novelty and diversity. Several works have focused on defining metrics for measuring such characteristics (Vargas & Castells 2011, Zhang & Hurley 2009, Lathia et al. 2010, Shani & Gunawardana 2011), and designing algorithms to provide novel and/or diverse recommendations (Weng et al. 2007, Onuma et al. 2009, Zhou et al. 2010, Jambor & Wang 2010).

- Novelty :
Novelty is based on suggesting to the user items she did not know before the recommendation (Shani & Gunawardana 2011), referred to as non-obvious items in (Herlocker et al. 2004, Zhang et al. 2002). Novelty can be explicitly measured in online experiments by asking users whether they are familiar with the recommended item (Celma & Herrera 2008). However, it is also interesting to measure novelty in an offline experiment, so as not to restrict its evaluation to costly and hardly reproducible online experiments.

In (Weng et al. 2007), novelty can be introduced into recommendations by using topic taxonomy, where items containing new topics are appreciated. In general, new topics are obtained by clustering the previously observed topics for each user. Onuma et al. (2009) introduced a graph-based technique in order to suggest nodes (items) well connected to older choices, but at the same time well connected to unrelated choices. Lathia et al. (2010) considered novelty as the amount of new items appearing in the recommended lists over time.

- Diversity : In Information Retrieval, diversity is defined as an issue of finding results that cover different aspects of an information need and avoiding redundancy (Radlinski et al. 2009). Therefore, most of the proposed methods used (explicit or inferred) query aspects (or topics) in order to diversify a prior result set (Clarke et al. 2008, Agrawal et al. 2009, Chandar & Carterette 2010, Radlinski et al. 2008, Rafiei et al. 2010).

In the literature, more formal definitions for diversity have also been introduced. Lathia et al. (2010) analyzed diversity of top-N lists over time by comparing the intersection of sequential top-N lists. Zhang & Hurley (2009) proposed a statistical measure of diversity, where a recommendation algorithm is considered as fully diverse if it is equally likely to recommend any item that the user likes. Bradley & Smyth (2001) propose a quality metric which considers both the diversity and similarity obtained in the recommendation list based on item similarities and focused on content-based algorithms.

- Scalability :
As RS are designed to offer to users a large collections of items, one of the goals of the designers of such RS is to scale up to real data sets. As presented by Das et al. (2007), it is often the case that algorithms trade other properties, such as coverage, in order to provide rapid results even for huge data sets. Sarwar et al. (2000*a*) evaluated the computational complexity of an algorithm in terms of time or space requirements.

Scalability is typically measured by experimenting with growing data sets in order to show how the speed and resource consumption behave when the task scales up (George & Merugu 2005). For example, if the accuracy of the algorithm is lower than other algorithms that only operate on small data sets, the difference in accuracy over small data sets must be showed.

Additionally, as RS are expected to provide rapid online recommendations, it is important to measure how fast the system provides recommendations (Herlocker et al. 2000, Sarwar et al. 2001).
Other metrics such as serendipity, privacy, adaptivity, and confidence have been less discovered in the literature. However, their importance and application to RS have already been discussed, making clear their relation with the user's experience and satisfaction (Herlocker et al. 2004, McNee et al. 2006, Shani & Gunawardana 2011).

2.6 Conclusion

We introduced in this Chapter basic concepts of RS and main state-of-the-art recommendation techniques proposed for this aim.
Moreover, we gave a brief introduction of the recommendation problem analysis and we

discussed principal limitations of the recommendation techniques.
Finally, we gave an overview on performance evaluation of RS.
After this general introduction to RS, we will move on to the problem of context awareness. In the next chapter, we will concentrate on context awareness in RS and we will dis- cuss main context-aware approaches proposed for these application domains.

Chapitre 3

Overview on Context-aware Recommender Systems

Contents

3.1 Introduction

The importance of contextual information has been recognized by researchers and practitioners in many disciplines such as e-commerce personalization, information retrieval, and ubiquitous and mobile computing (Schilit & Theimer 1994, Chen & Kotz 2000).
While a substantial amount of research has previously been performed, most existing approaches focus on recommending the most relevant items to users without considering any additional contextual information (e.g., the company of other people : watching TV with friends) other than time and location.
In this chapter, we discuss the general notion of context and how it can be defined and integrated in RS. In Section 3.2, we define the context in IR and RS. In Section 3.3, we introduce the ways of obtaining contextual information. In section 3.4, we present the classification of the dimensions of the context and we present how contextual information are integrated into the recommendation process. Finally, we present diverse capabilities for incorporating contextual information in TV recommender systems.

3.2 General Notion of Context

In this section, we define the multifaceted concept of the context and describe its use in several fields that are directly related to RS, such as information retrieval. Applications that take into account contextual information are called 'context-aware systems (Schilit & Theimer 1994).

3.2.1 Defining context

Context is a multifaceted concept that has been studied across different research fields, such as computer science, cognitive science, linguistics, philosophy and psychology (Adomavicius & Tuzhilin 2005).
An entire conference, CONTEXT [1], is dedicated exclusively to studying this topic and incorporating it into various other branches of science, including medicine, law, and business.
The standard generic dictionary definition of context as *"conditions or circumstances which affect something"* (McKechnie 1983). Bazire & Brézillon (2005*a*) presented and discussed 150 definitions of context from various fields. For instance, Schilit & Theimer (1994) defined context as *"location and the identity of nearby people and objects"*.
However, according to Schilit et al. (1994), *"Context encompasses more than just user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, communication bandwidth, network connectivity and even the social*

1. http ://context-07.ruc.dk

situation (e.g. whether you are with your manager or with a co-worker)".
Later, Dey et al. (2001) moved to a more abstract definition : *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application"*. This is probably the first definition that was broadly adopted in the computational sciences.
Prahalad (2004) affirmed also that "the ability to reach out and touch customers anywhere and at anytime means that companies must deliver not just competitive products but also unique, real-time customer experiences shaped by customer context".

3.2.2 Context in Information Retrieval

Although Bazire & Brézillon (2005*b*) did not settle on a specific definition, the questions raised by the authors take in consideration all domains in which context awareness is necessary or desired, including information retrieval and recommender systems. Particularly, in web search, context is defined as the set of topics potentially related to the search term. However, with the evolution of information retrieval, the evolution of the Web and devices, many researchers focused on recommendation systems field to improve results to the users queries on the web.

As proved by Jones & Brown (2002), contextual information are widely exploited in Information Retrieval. While most existing systems set up their retrieval decisions only on queries and document collections, information about search context is often ignored (Akrivas et al. 2002).
In Web search, context is defined as the set of topics potentially related to the search term. For example, Lawrence (2000) described how contextual information can be used and proposed several domain-specific context-based search engines.

The effectiveness of a proactive retrieval system relies on the ability to perform context-based retrieval by generating queries which return context-relevant results (Sieg et al. 2007).
The integration of context into the Web services composition is initially suggested by Maamar et al. (2006).
Many works have exploited combinations of the users context. Preferences and content are proposed in Missaoui & Faiz (2014) to give accurate information that meet the individual user needs without waiting for the user to initiate any interaction or activity with his device.

Boughareb & Farah (2014) proposed a taxonomy which gathers all user contextual dimensions studied in the search contexts field. Others researchers have focused on contexts such as the user's intention behind the query (Kathuria et al. 2010, Missaoui & Faiz 2014), the user's location (Magara et al. 2016, Yuan et al. 2013, Noguera et al. 2012) and temporal information Missaoui & Faiz (2014), Ji et al. (2015), Panayiotou et al. (2005), Liu, Zhao, Xiang & Yang (2010) to enhance the research engines' results.

3.2.3 Context in Recommender Systems

In RS, entity is usually a user, an item and the experience that user is evaluating. (Dourish 2004) distinguished two main definitions of the context in computational environments : representational and interactional. According to the author, the representational view separates the context from the action. In such a way, the context defines an action and provides some form of information about it. On the other hand, the interactional view defines the context as a relational property and the scope of the context is defined dynamically. Thus, no enumeration of contextual conditions is possible beforehand. The author show that Context arises from the activity. Context isn't just there, but is actively produced, maintained and enacted in the course of the activity at hand.

Obviously, Dey et al. (2001) introduced a large definition that should be refined in a concrete Recommendation Systems scenario. In RS, context is usually considered as an additional and relevant information (excepting users and items) at the current time of recommendation.

As explained in Chapter 2, the recommendation process starts with the specification of the initial set of ratings explicitly provided by the users or implicitly inferred by the system. Then, a recommender system tries to estimate the rating function R for the (*user*, *item*) pairs that have not been rated. These systems are called traditional or two-dimensional as they consider only the User and Item dimensions in the recommendation process.
Therefore, with the incorporation of the context, the rating function R is extended into three dimensions :

$$R : User \times Item \times Context \times Rating \tag{3.1}$$

where *User* and *Item* are the domains of Users and Items respectively, *Rating* is the domain of ratings, and *Context* represents the contextual information.

The context is commonly associated with the application domain of the recommendation and each context type has a well-defined structure. For instance, if the integrated context is the *time*, then the rating assigned to an item by a user depends on when the item has been seen. Adomavicius & Tuzhilin (2005) presented a methodology to decide which contextual attributes should be used in a recommendation application (and which should not). Their methodology is based on the assumption that a wide range of contextual attributes must be

initially selected by the domain experts as possible candidates for the contextual attributes for the application.
Contextual information can be of different types, each type defining a certain aspect of context (e.g., time and location). Furthermore, each contextual information can have a complicated structure and complex nature.

There are several works that focused on context representation. For instance, Yu et al. (2006), Kim & Kwon (2007) proposed an ontology-based context-aware recommendation system. They used ontologies to represent semantics of the recommender knowledge. The advantage of representing concepts through ontologies is enriching information when it is imprecise or incomplete and supporting the interoperability and the exchange of information between systems (Buriano et al. 2006). Kim & Kwon (2007) used also four types of ontologies : product, location, record and customer. Based on his shopping history, the authors proposed a method that extracts a consumer's preferred items and recommends similar items according to the ontology.

Adomavicius & Tuzhilin (2005) used a hierarchical representation of contextual dimensions. In the proposed method, each dimension of the context has an associated hierarchy, which could be used to aggregate underlying data. The enumeration of variables (one for each contextual dimension) Domingues et al. (2011), Baltrunas & Ricci (2010), Oku et al. (2006) are considered the simplest and the most extensively used approach.

Early works on context-aware computing concentrated on motivating and explaining how contextual information are modeled and the technical aspects related to how to collect and store contextual information. In this subsection, we present theoretical analysis on how literature works have incorporated contextual information into recommendation process.

3.3 Obtaining Contextual Information in RS

In recent years, companies like Apple [2] can easily track the user's location, which can be important to the study of many Context-Aware RS. Unfortunately, Apple has been very protective of its CARS related data and relevant research.
As far as the online music listening platform Spotify [3] is making context-aware recommendation to its users (more than half a million users) based on time and location, but has never released also any user data and any research work.

Consequently, how to obtain contextual data and effectively learn user's preferences from the data are considered very challenging for researchers of Context-Aware RS (Adomavicius

2. http ://www.apple.com
3. https ://www.spotify.com/

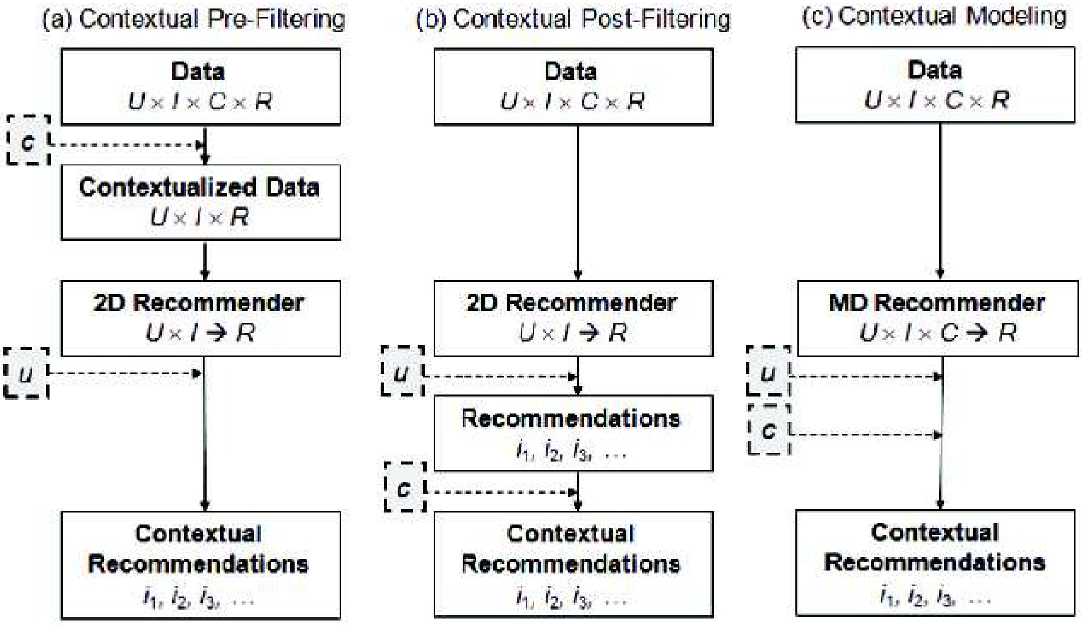

FIGURE 3.1: The difference between the three forms of context uses (Ricci et al. 2010)

& Tuzhilin 2005, Ansari et al. 2000, Umyarov & Tuzhilin 2011, Oku et al. 2006, Yu et al. 2006, Aizenberg et al. 2012, Hariri et al. 2012*a*) . The lack of contextual data must be previously regarded to designing any Context-Aware Recommender System.
Accordingly, it is very difficult for researchers (Adomavicius & Tuzhilin 2008, Verbert et al. 2012) to conduct research on Context-Aware Recommendation Systems without necessary and enough data. The lack of real contextual data was and still the fundamental issue of Context-Aware Recommendation Systems.

Several researchers and practitioners presented different ways to obtain contextual information and can be broadly categorized into two types of approaches : *Transaction-based methods* and *Session-based methods*. In the following subsections, we introduce these two types of approach and discuss their advantages and their limits.

3.3.1 Transaction-based Methods

In this method, implicit transaction data are used. For instance, we can obtain geographical information from transaction location, and infer user's preference from transactions (Adomavicius & Tuzhilin 2005).

As described in Figure 3.1, the user preference estimation can take one of the three forms, based on which the context is used :

— Contextual pre-filtering (or contextualization of recommendation input) : As drawn in (Adomavicius & Tuzhilin 2005), this recommendation form is based on the assumption that information about to the current context c is used to construct or select the relevant set of data records (i.e., ratings). Therefore, ratings can be predicted using any traditional two-dimensional recommender system on the selected data.

— Contextual post-filtering (or contextualization of recommendation output) : In this recommendation paradigm (presented in Figure 3.1b), contextual information is initially ignored, and the ratings are predicted using any traditional two-dimensional recommender system on the entire data. Then, the resulting set of recommendations is adjusted (contextualized) for each user using the contextual information.

— Contextual modeling : In this recommendation paradigm (presented in Figure 3.1c), contextual information is used directly in the modeling technique as part of rating estimation.

In this context, several recommendation algorithms (Ansari et al. 2000, Adomavicius & Tuzhilin 2005, Umyarov & Tuzhilin 2011, Oku et al. 2006, Yu et al. 2006) have been proposed based on a variety of heuristics and predictive modeling approaches. In heuristic-based approaches, the traditional two-dimensional neighborhood-based approach (Breese et al. 1998, Sarwar et al. 2001) can be extended to the multidimensional case where contextual information is included in a straightforward manner by using an n-dimensional distance metric instead of the user-user or item-item similarity metrics.

On the other hand, some of model-based methods were directly extended to the multidimensional case. For instance, Ansari et al. (2000) combined the information about users and items into a single hierarchical regression-based Bayesian preference model that uses Markov Chain techniques in order to estimate its parameters. They showed that their proposed two-dimensional technique outperforms some collaborative filtering methods.

Oku et al. (2006) incorporated additional contextual dimensions such as time directly into recommendation space and used machine learning technique in order to provide restaurant recommendations. Particularly, they used support vector machine classification method, which considers the set of liked items and the set of disliked items of a user in various contexts as two sets of vectors in an n-dimensional space. Then, they constructed a separating hyperplane which maximizes the separation between the two data sets.

However, after data selection in contextual pre-filtering, the data sparsity increases. Therefore, the system may not have sufficient data to make accurate recommendations (Adomavicius & Tuzhilin 2005). Contextual post-filtering is faced with the same problem. In these methods, the contextualization process requires defining a set of contexts which are not personalized. Exact and detailed context can also lead to data sparsity, while generalized context leads to inaccurate recommendations.

Consequently, the following research questions are raised :
- Whether a recommendation method, that is able to overcome the data sparsity in context-aware RS, could be developed ?
- Whether a recommendation method, that is able to define personalized contexts, could be implemented ?

3.3.2 Session-based Models

In this method, active sessions are used. We can analyze the characteristics of the session and make context-aware recommendations, if we know what items the user has chosen in an ongoing session (e.g. browsing session, movies watching session).
Frequently, it is not possible for a recommendation system to obtain additional contextual information. However, items chosen by the user in the current context are known (seed items). Thus, if we assume that the context remains the same, the system can make context-aware recommendations based on these seed items.
In this context, several methods (Aizenberg et al. 2012, Hariri et al. 2012*a*) are proposed to make context-aware recommendations where a context is usually defined as an active and continuous session with the system (e.g., a music listening session, a web browsing session).

- Non-personalized session-based methods :
Recommendations are based on the items chosen by the user in the current context (seed items). The user's general preference has little influence on the recommendations, thereby, recommendations are not personalized. The same seed items would lead to the same recommendations (Hariri et al. 2012*a*).

- Personalized session-based methods :
Typically, each session is viewed as a mixture of topics. Each topic has unique item distributions. Thus, the principal task of session based methods is to identify the underlying association between contexts, and to find similar contexts (sessions) based on the topic mixture.
Most session-based methods are based on latent factors, which determine the characteristics of topics (i.e., characteristics of a context) (Jin et al. 2006, Hariri et al. 2013, Blei et al. 2003, Zheleva et al. 2010).

3.4 Context Dimensions

The context could be considered as the trigger of the user preference change. As we already mentioned, classic recommender systems ignore the contextual reasons behind the user preference. Such missing link would lead to inaccurate recommendation, and exploiting it

might results better personalized outputs. Context can be defined as set of dimensions that can be used to characterize the situation of the user. It could be a location, time, activity or companion that is considered relevant for improving recommendation.
In this Section, we presented the different dimensions of the context used in the literature.

3.4.1 Spatio-Temporal Dimension

The user's behaviors are dynamic and continually changing since they are not only influenced by their personal interests, but also by external factors such as the location and the temporal context. For example, during normal working day, pizza restaurants and rent-a-movie places might be a good recommendation as places to go. In contrast, open bars and happy hours are of great interest to users during vacations period. Thus, the construction of user profiles is a challenging task since it might be different during the various periods.

In Ji et al. (2015), authors proposed to track the changes of user interests' over time taking into account the long-term (user's global interest) and short-term effects (distance between items) as well as session term effect (user's local interest in each session) in order to recommend next-song music. Then the three time changing effects are joined up to identify user's present interest. Panayiotou et al. (2005) suggested dividing the day into different time-zones according to the users daily routine and activities for each period. Then, associate each users interest in a particular time zone with a set of weights. Doing so allows the dynamic creation of the user profile based on the current time and activity by applying the relevant weight set on his preferences. Similarly, Missaoui & Faiz (2014) split the day into time slots that help to determine the information type to recommend according two levels time of the day (morning, midday, afternoon, evening and night) and Week day (workdays, vacations and public holidays).

In Liu, Zhao, Xiang & Yang (2010), authors improved the classic neighborhood based recommender systems by incorporating temporal information to adapt it to the changes in both user and item characteristics over time. Also, they proposed a new algorithm for updating neighborhood similarities as new data are generated at each time step. Progression in position localization techniques and the evolution of devices such as smart phones and other mobile gadgets have enhanced the recommendations in many services and have launched a new wave of research in the area of recommender systems.

Offering many opportunities, location-based recommendation system have vast applications such as transportation, and tourism. Many online services like Foursquare and Facebook have succeeded in improving their recommendations by using the users location-related information. Obviously, users location histories contain a rich set of information reflecting their preferences. In this context, Hariri et al. (2012*b*) introduced a new recommender system of music MPlist that matches users context and the next song played. MPlist utilizes input sensors available in smart mobile devices (GPS and Wi-Fi) to collect raw data. Location of the user is then deduced as either being indoor (e.g. office, home etc.) or outdoor, to play his favorite kind of music in that context based on the users previous listening history in reference to the current context, other users listening preferences when in similar context, and listening profiles of nearby users.

Mobile tourism applications which recommend attractions or tourist services is another example that reflects the importance of location context. This kind of applications are more focused on contextual information to determine the appropriateness of items because of the extreme sparsity of user-item interactions since a user can only visit a limited number of places. In Yuan et al. (2013), the authors introduced a place-to-go recommender. They studied the impact of distance on users check-in behaviors. Assuming that human tend to visit nearby Point-of-interest (POI) such as restaurants to their previous locations, and their willingness to visit a POI decreases as the distance increases, they proposed a new recommendation method. Indeed, the application proposed in Noguera et al. (2012) streams progressively an interactive 3D map and provides the client with different categories of POIs (cities, monuments, geographic features, etc.) according to the users geographical position obtained via GPS. First, the introduced algorithm reduces the number of items considered for the recommendation according to the users location. Then a distance based re-ranking is applied to re-rank the previous top-N list according to the physical distance from the user to each item.

3.4.2 Social Dimension

According to Adomavicius & Tuzhilin (2008), the *social context* in recommender systems represents the presence and role of other people (either using or not using the application) around the user, and whether the user is alone or in a group when using the application.

As the web 2.0 has developed and the number of users of social media has increased significantly. Recommender Systems have increasingly incorporated social information (e.g., trusted and untrusted users, followed and followers, friends lists) to improve their precision since people tend to seek advice from their entourage before purchasing a product or consuming a service. Thus, recent works such as (Ebrahimi & Golpayegani 2016, Yang et al. 2012, Sun et al. 2015, Seo et al. 2017) proposed new models and techniques to integrate the social context into the recommendation process improving both the recommendation quality and scalability and alleviate the data sparsity problem.

Ebrahimi & Golpayegani (2016) proposed a novel framework based on Collaborative Filtering recommender system by making use of social network data for computing similarity and neighbors set simultaneously without consideration of any rating history. Yang et al. (2012) refined the Friends concept to Friends Circles and introduce a recommender system using the trust circles. In Sun et al. (2015) authors used the same approach and cluster the users friends to obtain smaller groups with the similar tastes for generating good recommendations. Indeed, Seo et al. (2017) proposed a personalized recommender algorithm based on friendship strength that recommends items (i.e., interests) to users by considering their tendency. The approach calculates the friendship strength by applying various characteristics of big social data on Social Networks Systems and use it as the similarity measure between users.

The work Guo et al. (2017) focused on social community discovery and friends recommendation systems in social media. Using an unsupervised algorithm, the method models the user relationship with other users through the multi-activities such as (tag, cofollow, comment, colike, like and follow) and suggest new friends based on the strength of relationship of the active user in his community.

3.4.3 Sentiments and Behaviors Dimensions

Emotions and feeling play an important role in our daily life decisions and activities. For example, psychology researchers have proved that happy users used to make positive choices such as shopping easier than sad users. But despite the sharp relation between emotions and feelings, and decision making, the combination of the two research fields have been limited. Therefore, emotions can be considered as a contextual factor. Few papers have introduced the concept of sentiment analysis into recommendation systems. However, many works (e.g. Narducci et al. (2015), Turrin et al. (2014)) highlighted the importance of considering the users feelings.

Narducci et al. (2015) proposed a general architecture EA-CBRS for developing emotion based recommender systems. The model includes an emotion analyzer for the content and the users profile, what make it able to assign an emotional label to a natural language text which can be extracted from social media or feedback. The model proposed in Turrin et al. (2014) demonstrates that a lexicon based sentiment can improve the performance of a music recommendation system. It extracts the users' sentiments from sentences posted on online social networks and then the music recommendation system suggests songs based on the current users sentiment intensity.

In the literature, there are several works that exploit users ratings history in order to estimate their preferences. Almost of these approaches involved collaborative filtering methods (Wang, de Vries & Reinders 2008, Umyarov & Tuzhilin 2011).
Furthermore, due to the emergence and the prevalence of social networks, users are no longer passive elements. They can now rate items, comment and suggest them to friends through social networks. For instance, Jessica has listened to a musical extract, then she makes several behaviors (e.g. she share and comment this extract in a social network). An

effective prediction approach must estimate the degree of interestingness or the relevance of this extract based on these interactions.
Obviously, user behavior is one of the most significant aspect to infer the users interests and opinion about the item in RS and recently, supporting information-gathering users behavior has been greatly studied (Bambia et al. 2015, Iwata et al. 2007).
Recommender systems (Lieberman 1995) are based on inferences made about user interests collected from their task environment for instance recently-viewed Web pages or the contents of active desktop applications.
User interests modeling systems have typically process previous user search-related interactions, explicit ratings history or reviews to predict user interests (Umyarov & Tuzhilin 2011).

3.5 Context-aware recommendation systems for TV contents and movies recommendation

With the evolution of smart TVs and the growth of programs and contents number, hundreds of channels from cable or satellite provider, along with great Internet-based content providers like Netflix, Hulu, YouTube, are available to users. While there are hundreds of channels with an abundance of programs, TV viewers usually switches the TV on and surfs over channels to select the program to watch. They waste a lot of time browsing the available options or end up watching a very limited number of channels.

Therefore, many famous television makers and content providers such as Google TV, Apple TV, Sony TV and YouTube have increasingly adopted recommender systems. As ambiance and situation might have an impact on the relevance of a TV content for the viewers, an effective context-aware system must take into account several types of contextual information such as time, companions and users' preferences.

Different architectures of personalized videos recommendation systems proposed in the literature were outlined by the survey presented by Asabere Asabere (2012). Likewise, several Social TV offerings and platforms were implemented in last few years (e.g. Netflix, GetGlue, GoogleTV, etc.), which allows the TV experience to move beyond the traditional confines of entertainment into a more holistic media. Obviously, according to "Netflix Challenges" [4], Netflix algorithms draw on the item-based collaborative filtering method and Matrix Factorization method to predict users′ preferences.

4. http ://www.netflixprize.com/

3.5.1 Spatio-Temporal in TV content Recommendation

Obtaining location and time from users devices such as laptop, smartphone or even smart TV has already become a global trend, since using the information that relates directly to the users specific might improve the recommendation proposed. Examples include store promotions, exhibition activity information, and TV programs.

In this context, Zong et al. (2017) obtained the geographic locations of the users and matched them with their weather. Since weather can influence people behaviors and affect their lives, the research analyzes whether people watch different genres of programs in different weather conditions. This study presents the first analysis that looks at the interplay between weather and watching TV. The correlations proved leave incorporating weather into a context-aware recommender for future work.

Indeed, one of the most important factors that affect the users preferences : time. For instance, watching TV depends on the temporal context (i.e., day of week and time of day). For example, during weekdays the user prefer watching weather forecasts and news while in the weekend he usually chooses to watch his favorite TV reality and talk shows. Indeed, if a user like watching horror movies in the daytime. It would be irrelevant to propose the film the conjuring at midnight.

Exploiting the advantage of using the temporal factor, Liu et al. (2016) designed a new recommender system for smart TV. The system proposed solve a major problem for recommendations systems which is data sparsity. In fact, it deduces the users interest distribution based on the video co-occurrence in his watching lists and applies then a weight post-filtering to temporal contextualize the top-N recommendation results.

Assuming that only a minor role is played by the characteristics (e.g., genre and subgenre) of the broadcast TV program, Turrin et al. (2014) limited the contextual parameters integrated in the algorithm proposed to the users preferred time slots and channels. In the same orientation, Oh et al. (2012) proposed a Time-Dependent method, grouping watch log dataset, to create an efficient user profile for TV Recommendation based on time and not based on user.

(Liu, Cao, Zhao & Yang 2010) implemented a time-aware collaborative model for movie recommendation. Based on the assumption that recent ratings are more important than historical ones, they incorporated temporal relevance using matrix factorization technique. The temporal relevance $f_{ui}(t)$ measures the relevance of each observed rating r_{ui} in order to make recommendation to viewer u at time t, as defined in Equation 3.2.

$$fui^{\beta}(t) = e^{-\beta(t-T_{ui})} \tag{3.2}$$

Where β is the parameter controlling the decaying rate.

They used Singular Value Decomposition (SVD) technique based on the loss function defined in Equation 3.3.

$$\min_{U,V} \sum_{i=1}^{m} \sum_{u=1}^{n} w_{ui}.(r_{ui} - u_i^T.v_u)^2 + \lambda(||U||_F^2 + ||V||_F^2) \tag{3.3}$$

Where w_{ui} is calculated as follows :

$$w_{ui} = 1 + fui^{\beta}(t) * (w_{max} - 1) \tag{3.4}$$

Oh et al. (2012) proposed a time-dependent recommendation technique. The construction of the user profile is based on splitting each watch log into time slots and generating a time-dependent profile for each time slot. Henceforth, when a recommendation is issued, the system finds the corresponding profile based on the time stamp of the request. Then, the similarity of video v and each video v' in the corresponding profile is calculated based on Pearson correlation coefficient between them, as defined in Equations 3.5.

$$similarity(v, v') = \frac{\sum_{u' \in U} (r_{u'v} - \overline{r_v}) * (r_{u'v'} - \overline{r_{v'}})}{\sqrt{\sum_{u' \in U} (r_{u'v} - \overline{r_v})^2} * \sqrt{\sum_{u' \in U} (r_{u'v'} - \overline{r_{v'}})^2} } \tag{3.5}$$

3.5.2 Social Context in TV content Recommendation

Several studies (such as Groh et al. (2012), Lathia et al. (2008) and Groh & Ehmig (2007)) proved that social filtering (e.g. group recommendation) is an efficient approach to cope with the sparseness problem in collaborative filtering. This is considerably for taste related domain, such as TV, cinema and music, which are strongly influenced by friends. Moreover, in contrast to other domains, profiling recommended TV shows is a hard task since TV is usually shared by a group of person such as family, what make the recommendation addressed to the group instead of individuals. Some previous works focused on the strength of the social connections, while others studied trust and the influence of opinion leaders.

Since users in the same group have doubtlessly different preferences, and might also have different tolerance levels to accept other members suggestions, Sun et al. (2017) introduced a novel approach based on experts : persons which their characteristics can largely influence the preference of the whole group. Because users are interested in social media content generated by their followees, the designed framework is composed of three components : a preference model for social groups, a personal tolerance model, and a followee-based preference model that uses external experts social behaviors, such as microblogs they posted and their relationship with the group members.

Barragáns-Martínez et al. (2009) introduced a personalized TV program recommendation system. To solve first-rater, cold-start, sparsity and overspecialization problems, they proposed a hybrid approach that combines content-filtering techniques with those based on collaborative filtering and provides advantages of any social networks such as comments, tags and ratings. They used vector space model to generate content-based recommendations. They used SVD (Singular Value Decomposition) to reduce the dimension of the active item′s neighborhood, and to execute the item-based filtering with this low rank representation to generate its predictions.

(Liu, Cao, Zhao & Yang 2010) proposed a social network-based movie recommendation technique. They used both collective matrix factorization and regularized matrix factorization. The collective matrix factorization is used in order to jointly factorize the rating matrix R ($R = U^T.V$) and the binary matrix G ($R = Z^T.V$) where $g_{uj} = 1$ if u and j are friends, based on the following training objective function :

$$\begin{aligned} \min_{U,V,Z} \sum_{i,u}(r_{ui} - u_i^T.v_u)^2 + \alpha \sum_{j,i}(g_{uj} - z_j^T.v_u)^2 \\ +\lambda(||U||_F^2 + ||V||_F^2 + ||Z||_F^2) \end{aligned} \tag{3.6}$$

Where α is a parameter used to control the importance of the matrix G.

On the other hand, the regularized matrix factorization is used to incorporate similarity between users and their friends based on the following network regularized matrix factorization function :

$$\begin{aligned} \min_{U,V} \sum_{i=1}^{m}\sum_{u=1}^{n}(r_{ui} - u_i^T.v_u)^2 \\ +\lambda_1(||U||_F^2 + ||V||_F^2) + \sum_{i,j} S_{i,j}||v_i - v_j||_F^2 \end{aligned} \tag{3.7}$$

$S_{i,j}$ equals to 1 if users i and j are friends, else $S_{i,j}$ equals to 0.

3.5.3 Preferences in TV content Recommendation

User preferences are passing into the fundamental ingredient of recommender systems in modern Web-based data-intensive applications. Since preferences are expressed differently from one user to another and in one system from another, modeling user preferences has been largely studied in recent years. Unfortunately, there is a need for more work and improvement to obtain the preferences, and the way they are integrated into recommendation systems.

Antonelli et al. (2009) proposed a content-based recommender approach using the textual descriptors associated to TV contents extracted from newspaper articles. They used matrix factorization technique to associate textual descriptors to TV contents. Chang et al. (2013) presented a TV program recommender framework integrating TV program content analysis module (e.g. TV program basic content information, watching statistics information, etc.), user profile analysis module (e.g. demographic information, watching histories, preferences) and user preference learning module (e.g. preferences of user implicit and explicit network).

To filter available TV shows based on the user interests and preferences, the work of Chang et al. (2013) came up with a framework that consists of TV program content analysis module, user profile analysis module and user preference learning module that collects and extracts users demographic information, watching histories, preference/Interest and social relationship from social media and relevant organization. Users interests are deduced from his experience using content-based filtering methods, implicit network (Users with similar preference) and explicit network (friends/family/colleague) using the collaborative filtering methods.

3.6 Conclusion

The concept of context-aware approaches in RS has been studied for several years. However, most of the work on context-aware RS has been conceptual, where a certain method have been developed and tested on some (often limited) data. The key issue here is the lack of contextual data.

In this chapter, we focused on Context-Aware RS. We introduced existing methods for modeling the context in RS, we draw on how contextual information are obtained and how data sparsity or lack of data are tackled. Then, we highlighted the limitations of existing Context-Aware RS :

- All the possible contexts used in Context-Aware RS are non-personalized which can lead to inaccurate recommendations
- Existing context-aware approaches consider only the current time and the user location

and ignore any additional contextual information on which users' preferences may depend.
- Existing Context-Aware RS can not deal with data sparsity and can not generate accurate recommendations on sparse data.
- Existing session-based Context-Aware RS can not achieve accurate user modeling and accurate context modeling at the same time.
The limitations of CARS motivates for better context-aware recommendation methods. In the following chapters, we propose new approach in order to improve existing Context-Aware RS.

Chapitre 4

A Personalized Context-based Approach

Contents

4.1 Introduction

In the previous chapter, we introduced the basic concepts and existing methods in CARS. We presented the dimensions of the context used in the literature and draw on how they are obtained and exploited.
Under these considerations, such needs remain unmet. How could we define the contextual information on which users' preferences depend ? How could we exploit such information for improving personalized recommendation amongst a huge number of items ?
In this chapter, we propose a context-based approach that captures and models the current context of the user for improving personalized items recommendation. In respect with this captured context, a probabilistic model is proposed in order to estimate the relevance of items.
The rest of this chapter is organized as follows. First, we formalize the problems and the limits related to context-aware approaches. Second, we present the proposed context-based approach for improving items recommendation. Finally, we describe the conducted experiments on real dataset crawled from a social TV platform, and the obtained results for highlighting the effectiveness evaluation of the proposed approach.

4.2 Problem Formulation and Positioning

In this section, we define the problem of context-aware recommendation and we provide an overview of terminology, techniques, and limitations related to the different types of context-aware approaches.

4.2.1 Problem Formulation

The sheer volume of the available items often undermines the user ability to choose the content that best fits her interests and that are perfectly adapted to her contexts. As described in Section 3.2.3, CARS seem to be natural solution for this problem. However, differently from other classic recommendation scenarios (e.g. books), there are other systems that require a more personalized recommendation in which the relevance of items is sensitive to several contextual information or context elements.
This is considerably important for RS in which the relevance of the items is sensitive to several contexts, and in which content-based recommendation is not accurately predicted since the content of the same item is changing daily.
In CARS field, the context is generally defined as a set of conditions or circumstances which affect the decision of the user in order to improve items recommendation (McKechnie 1983,

Bazire & Brézillon 2005*a*).

Example 4.2.1. In this example, we are interested in TV recommendation. Therefore, we define the *current viewing context* as the set of circumstances related to the actual environment of the user that may influence his/her preferences. For example, on weekdays morning a user might prefer to watch world news (e.g. CNN or BBC) in the morning, the stock market report on weekends, and movies' reviews on Friday night.
In our work, we define the *current context* as the set of circumstances related to the actual environment of the user and that may influence his/her preferences. Therefore, we argue that contextual information consists of the following attributes :
- Time : indicates when the movie can be or has been seen,
- Location : represents the actual location of the viewer.
However, TV content recommendation is arguably more challenging, since TV programs content is changing daily. Though being interested with the whole program, a viewer might not prefer the actual content. In such a way, the relevance of TV content is sensitive to several contexts.
Further, there are several contextual information that may influence the viewer preferences such as :
- Weather : represents the actual weather of the viewer,
- Occasion : represents the event existing in the calendar of the viewer in the actual time slot (e.g. workout, meeting, Christmas),
- DayOfWeek : has values Mon, Tue, Wed, Thu, Fri, Sat, Sun.

In order to integrate these contexts, we have to consider two properties :
- **Genericity (or Genericness) :** The genericity means in this case the possibility for a model to provide parametrized modules or types. The genericity of the context elements must also be considered. In such a way, we can integrate any additional contextual attribute in generic way.
- **Independence :** Each contextual information can have a complicated structure reflecting complex nature. Therefore, the independence between the context elements must be treated in the recommender process.
The question that arises here in this case is : How to integrate independently and in a generic way any additional contextual information ?

Otherwise, user-cold start problem occurs when the system that does not have enough information (e.g. ratings, and browsing history) about a new user or a new item, and thus it is not able to provide the user with accurate recommendations or to reliably recommend the new item to any user (See. Section 2.4). Actually, a more challenging task is how to improve the recommendation accuracy for the new (or rarely rated) items and the new users (Schein et al. 2002).

For CARS, this problem is named context-cold start problem. For the newly released items and the old ones that are rarely viewed by users in such a context, it is difficult for the standard recommendation approaches such as collaborative filtering approach to provide accurate recommendations.
On the other hand, sparsity problem is also considered as the most challenging problems in RS. If the recommender system is based on explicit data, each user has to rate a sufficient number of items before the system can learn the user's preferences. However, in reality, most users are reluctant to provide ratings, and typically rate only a small proportion of the available items. Therefore, the dataset is sparse.
Consequently, having new contextual situations or not enough contextual information, a crucial question is : How accurate context-aware recommendations can be produced in the despite the cold-start situations and the data sparsity ?

4.2.2 Limits of Context-Aware Approaches

As we have already shown in Section 3.2.3, many context-aware approaches have been proposed in the literature such as the pre-filtering and post-filtering approaches.

Unfortunately, most of these approaches have been conceptual, where certain methods have been developed and tested on some and often limited data.
Obviously, most existing context-aware approaches (Ricci et al. 2015, Macedo et al. 2015, Turrin et al. 2014, Hariri et al. 2014) did not exploit all the elements of the context to predict users' preferences. They consider only two- dimensional representation – in every case only the current time and location are considered. However, as highlighted above there are several contextual information on which users' preferences undoubtedly depend (e.g. the actual weather and occasion).

Example 4.2.2. Taking Spotify [1] as an example, it is one of the most popular music recommendation systems. Spotify is a Swedish music, podcast, and video streaming service, launched in October 2008, that provides digital rights managementprotected content from record labels and media companies.
The user playlists vary from week to week, presumably reflecting the shifting musical preferences of the user. Since contextual data is becoming more and more important in the refining of musical recommendations, Spotify recommendations is based also on the user's mood (or listening occasion) such as workout, exams period and party. To each mood, the system assigns a playlist.
However, as highlighted in Figure 4.1, this is the user who must manually set her moods in order to reach the associated playlist.

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

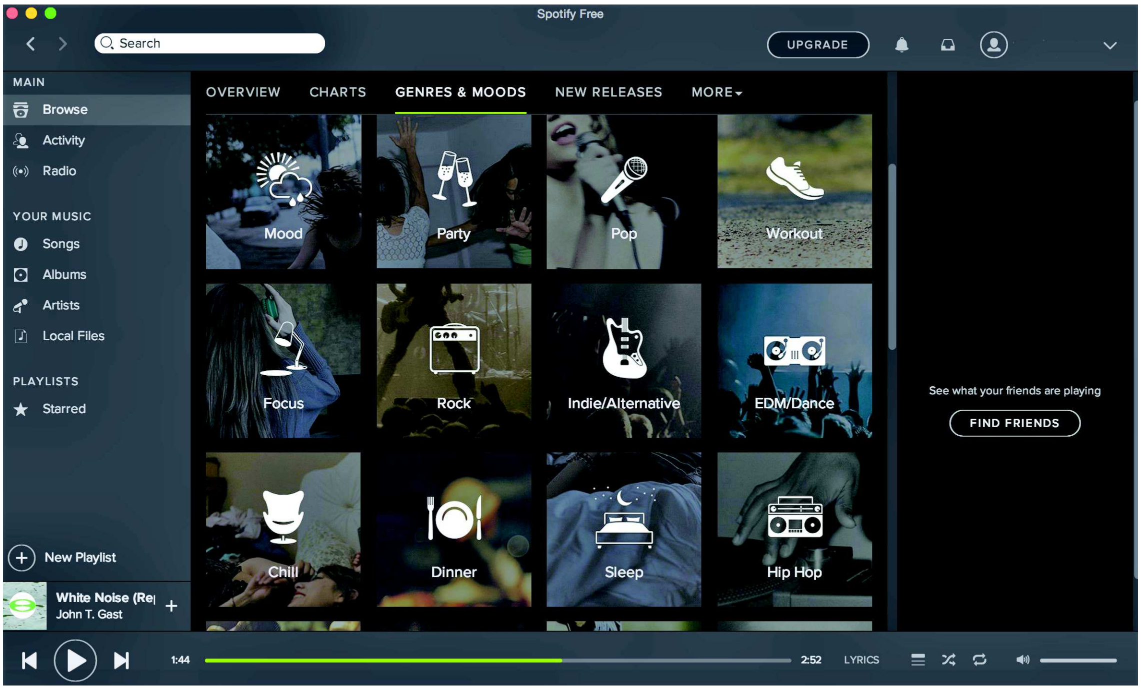

FIGURE 4.1: An example of a recommendation based on users' moods generated by Spotify

On the one hand, most of the proposed approaches (Oh et al. 2012, Antonelli et al. 2009, Barragáns-Martínez et al. 2009, Chang et al. 2013, Asabere 2012) draw on collaborative- based filtering and content-based techniques. They also have few consideration about solving recommender problems (e.g. no items seen by a new user known as "cold-start problem", and not enough co-rated items with other users with similar preferences known as "sparsity problem").

Overall, while there is a substantial amount of research on cold-start in traditional RS, only little research has been conducted on context-start problems in CARS. In the field of traditional RS, most of the proposed approaches (Pyo et al. 2013, Turrin et al. 2014, Oh et al. 2012, Antonelli et al. 2009, Chang et al. 2013, Asabere 2012) solved this problem by applying collaborative methods with latent factors, such as matrix factorization.
However, in CARS, these methods are not always effective since other recommender problems might occur (e.g. no items seen by a new user or a user with new context known as "cold-start problem", and not enough no similar contexts known as "sparsity problem").

4.2.3 Research Questions

In this research, we propose a personalized context-based approach that captures and models the current context of the user. In respect with this context, the proposed probabilistic

model estimates the relevance of available items. This model integrates several contextual information independently and in a generic way.
To the best of our knowledge, this is the first approach providing the integration of several context elements and enabling the integration of any additional contextual information. Our approach considers also context-cold start and context sparsity problems.
This research explores probabilistic models field to answer to these questions :
1. How to integrate independently and in a generic way any additional information related to the current context of the target user in order to improve items recommendation ?
2. How to overcome context-cold start and data (or context) sparsity problems ?

4.3 Context-based Model

In this section, we present the proposed context-based model which aims to exploit the actual contextual information of the target user in order to improve items recommendation. To explain our model in a simple and unified way, we take the case of movies recommendation.
In this case, we refer to the set of viewers as U and to the set of videos as V. $V(u)$ represents the set of videos viewed by user u, and $U(v)$ represents the set of users which have viewed video v.
L may link a viewer u and a video v. L represents the context in which user u has viewed video v. Let C be the set of elements of the viewing context, $c_{uv} \in C$ is the viewing context of u in which he viewed video v.
A model describes data that one could observe from a system. Therefore, our first aim is to model the context of the user. In our case, the viewing context c_{uv} is represented as a set of properties $c_{uv}\{c_{1uv}, \ldots, c_{muv}\}$ (i.e. *time slot*, *the location*, *the week day*, *the weather* and *the occasion*).

Example 4.3.1. For example, Jessica regularly watched romantic movies on weekend night. However, this weekend she woke up early and decided to watch TV. Intuitively, Jessica's viewing rating is decided by both her preferences and her current context. Her current context is $c_{Jessica}$ = {**location**= London Soho, **time slot**= 08-09, **weekday**= Saturday, **weather**= 9°, **occasion**= weekend night}. Jessica's interests will be reflected from other users' preferences in similar contexts.

Consequently, our second aim is to estimate the relevance (i.e. the rating) of items in respect with the actual context of the user. In our case, we aim to predict the relevance of a video v in respect with the actual context of Jessica based on ratings of users which watched v in similar contexts.

Under these considerations, a probabilistic model is defined as a statistical analysis tool that estimates, on the basis of past or historical data, the probability of an event occurring again (Robertson & Sparck Jones 1988, Robertson & Zaragoza 2009). It is a formalism of Information Retrieval useful to derive ranking functions that aim to rank matching items according to their relevance in respect with a given user needs.

Accordingly, we propose a probabilistic model that estimates the probability of finding if an item v is relevant to a context c_u. In other words, the model quantitatively captures contextual information in order to mine users' preferences in certain contexts. The goal of this model is to estimate the relevance of the target video v for the target user u given his current context $c_u\{c_{1u}, \ldots, c_{mu}\}$. Then, we aim to predict $Pr(r_{uv} = k|c = c_u)$ which is the conditional probability that the rating of the target user u on video v is equal to value k, given the current context c_u of user u. c_u represents the current context of viewer u. In such a way, videos with high probabilities will be recommended to viewer u.
By considering all dimensions of the context, the probability $Pr(r_{uv} = k|c = c_u)$ can be written as $Pr(r_{uv} = k|c = c_u\{c_{1u}, \ldots, c_{mu}\})$.

We consider the dynamic change of viewer preferences according to his current context considering the context-based model. We aim to estimate the relevance of video v for user u given his current context c_u. This probability can be estimated as the relevance of video v for viewers having approximately the same viewing context than u.
Inverse probability (i.e. Bayes rule) allows us to infer unknown quantities, adapt our models and make predictions from data related to contextual information. Therefore, based on Bayes rule, $Pr(r_{uv} = k|c = c_u)$ could be factorized as follows :

$$\begin{aligned} & Pr(r_{uv} = k|c = c_u\{c_{1u}, \ldots, c_{mu}\}) \\ = & \frac{Pr(c = c_u\{c_{1u}, \ldots, c_{mu}\}|r_v = k) \times Pr(r_v = k)}{Pr(c = c_u\{c_{1u}, \ldots, c_{mu}\})} \end{aligned} \tag{4.1}$$

Because this probability depends on the context elements' values rather than the user u, we drop the subscript u in r_{uv} for simplification.

Moreover, we assume that context properties are all independent from each others. In probability theory, two variables are independent if the realization of one does not affect the probability distribution of the other.
The concept of independence expands to dealing with collections of more than two random variables, in such a way they are pairwise independent if each pair is independent of each other, and they are mutually independent if each event is independent of each other combination of events.
Two variables A and B are independent if their joint probability equals the product of their probabilities :

$$Pr(A \cap B) = Pr(A) \times Pr(B) \tag{4.2}$$

Then, based on the probabilistic independence theory, $Pr(r_{uv} = k|c_u\{c_{1u}, \ldots, c_{mu}\})$ can be written as follows :

$$\begin{aligned} &Pr(r_{uv} = k|c = c_u\{c_{1u}, \ldots, c_{mu}\}) \\ = &\prod_{i=1}^{m} \frac{Pr(c_i = c_{iu}|r_v = k) \times Pr(r_v = k)}{Pr(c_i = c_{iu})} \end{aligned} \tag{4.3}$$

where c_i is the element i of the current context.
We assume that $Pr(c_i = c_{iu})$ is uniform. Thus this probability can be estimated as follows :

$$\begin{aligned} &Pr(r_{uv} = k|c = c_u\{c_{1u}, \ldots, c_{mu}\}) \\ \propto &\prod_{i=1}^{m} Pr(c_i = c_{iu}|r_v = k) \times Pr(r_v = k) \end{aligned} \tag{4.4}$$

On the one hand, $Pr(c_i = c_{iu}|r_v = k)$ represents the conditional probability that the video v was watched within a context element c_i equals to the actual user context c_{ui}, knowing that the rating given for v is equal to k.
As described in Equation 4.5, this probability could be estimated by calculating the ratio between the number of times the video v was watched views where ratings on video v that equal to k and the context element c_i equals to the actual user context c_{ui}, and the number of times the video v was watched where ratings on video v are equal to k.

$$\prod_{i=1}^{m} Pr(c_i = c_{iu}|r_v = k) = \prod_{i=1}^{m} \frac{|r_v = k, c = c_{iu}|}{|r_v = k|} \tag{4.5}$$

where $|r_v = k, c_i = c_{iu}|$ represents the number of ratings equal to k given to video v and where u's viewing context item is c_{iu}, and $|r_v = k|$ is the number of views of v where the ratings are equal to k regardless the viewing context.
On the other hand, $Pr(r_v = k)$ represents the probability of having a rating equal to k for video v. It can be estimated as the ratio between the number of ratings equal to k given to video v, and the total number of ratings on video v.

$$Pr(r_v = k) = \frac{|r_v = k|}{|r_v|} \tag{4.6}$$

where $|r_v = k|$ is the number of ratings on video v that equal to value k, and $|r_v|$ is the number of ratings on v.

However, $Pr(r_{uv} = k|c = c_u)$ could be equal to 0 if there are no views' ratings equal to k for all context elements, and $Pr(r_{uv} = k|c = c_u)$ could be equal to 1 if all ratings are equal to k in all contexts. In this case, it is required to not assign low probability (zero probability) to unseen contexts or ratings, or strong (probability = 1) probability with all seen contexts.

Smoothing the maximum likelihood model is extremely important when estimating a model based on a limited amount of data. The term smoothing refers to the adjustment of the maximum likelihood estimator in order to produce more accurate probabilities and to solve data sparsity (Zhai & Lafferty 2001, Chen & Goodman 1996). The name smoothing is centered around the fact that these techniques tend to make distributions more uniform by adjusting probabilities and improving the accuracy of the model.

In the literature, there are several works that focused on the issue of smoothing accuracy, such as Jelinek-Mercer (Jelinek & Mercer 1980), Dirichlet (Smucker & Allan 2005), and Laplace (Chandra & Gupta 2011).

A smoothing method may be as simple as adding an extra count, which is called additive smoothing or Laplace smoothing. Recent studies have proven that additive smoothing technique (Chandra & Gupta 2011) is more effective than other methods in several retrieval tasks such as language models and RS (Hazimeh & Zhai 2015, Valcarce et al. 2016), for solving strong probabilities problem particularly for small size of training samples.

In our case, we choose to use this technique, in such a way, we pretend we have seen each $n - gram$ times more than we have. Therefore, for estimating $Pr(c_i = c_{iu}|r_v = k)$, we assume that we have seen each '$r_v = k, c = c_{iu}$' one more time, and that we have seen '$r_v = k$' nc_i more times, as shown in Equation 4.7.

$$\prod_{i=1}^{m} Pr(c_i = c_{iu}|r_v = k) = \prod_{i=1}^{m} \frac{|r_v = k, c = c_{iu}| + 1}{|r_v = k| + nc_i} \tag{4.7}$$

Where nc_i is the number of possible values for each context element. For estimating $Pr(r_v = k)$, we assume that we have seen each '$r_v = k$' one more time, and we have seen 'r_v' nr more times, as shown in Equation 4.8.

$$Pr(r_v = k) = \frac{|r_v = k| + 1}{|r_v| + nr} \tag{4.8}$$

where nr is the number of possible rating values.

4.4 Experimental Setup

In this section, we introduce the experiments conducted on real data set crawled from a Social TV platform. First, we present the objectives behind these experiments. Second, we define the framework evaluation of our approach. Then, we present the evaluation protocols in which we define the evaluation metrics and the baseline models to which our approach is compared. Afterward, we interpret the obtained results. Finally, we describe and perform the carried out studies on context elements impact and on recommendation problems resolution.

The aims of these experiments are :

1. Evaluating the effectiveness of the proposed context-based approach ;
2. Evaluating the impact of each context element ;

3. Evaluating the ability of the proposed approach to solve context cold start and sparsity problems ;
4. Evaluating the effectiveness of integrating several contextual information by drawing up a comparison study between the proposed approach and some approaches of state-of-the art.

4.4.1 Evaluation Framework

In this section, we evaluate the effectiveness of the proposed context-based approach for improving items recommendation. Particularly, we conduct an effectiveness evaluation using real data collected from Pinhole [2] social TV platform. This dataset includes viewer-video access history and viewers' friendship networks.
We expanded an existing social TV system based on the proposed approach in order to perform the personalized content recommendation in a context-aware environment.
Pinhole is an innovative platform for social television to improve the TV viewing experience and help producers reach more audience online.
The idea of the startup was conceived in 2012, with the goal of delivering a technology that enables the user to interact with live shows. Officially, the startup launched its product after 7 months of software development. It has accumulated more than 400K users during its first 3 months.
Pinhole offers a social media application layer which enables the connection with established social media platforms such as Facebook, Twitter, Google+ [3], etc. It enables the users to watch VOD (Video On Demand) content and to interact with shows and TV programs. On the other hand, it allows marketers to target users through ads and coupons directly relating to the content seen on TV and VOD.
Over the last two years, Pinhole has also started pitching its product to potential advertising clients with great feedback. With such large success in Tunisia, Pinhole management is looking outside the country for expansion, with the goal of reaching the whole MENA region by 2014.
The number of people who are in the process of watching television on their second screen (Laptop / Tablet) is currently increasing in the Arab world, which justifies the interest to address this kind of audience and provide them relevant contents perfectly adapted to their consumption patterns.
We collected a sample of data consisting of 16,000 users, 81,000 TV shows and 721,121,391 views. The statistics of these data are displayed in Table 4.1.

2. www.pinhole.tn
3. https ://plus.google.com/

Statistics	Quantity
Number of users	16,000
Number of TV shows	81,000
Number of views	721,121,391
Number of social interactions	3,440,218
Average number of friends per user	124

Table 4.1: Statistics of the dataset used in the conducted experiments

4.4.2 Evaluation Protocol

In Pinhole system, the user can rate a viewed video from 1 to 5 representing the number of stars. However, almost of viewers do not rate all viewed videos. Since an unknown rating implies that we have no explicit information about the user's preferences, we could not rely on explicit ratings of viewers to measure the relevance of each item.
Therefore, we assume that the relevance of a video v watched by a viewer u could be estimated based on the amount of time he/she has watched v, as opposed to videos that he/she clicks and then abandon. We estimate the user rating r_{uv} according to the time spent watching a TV show :

$$r_{uv} = \frac{Number\ of\ minutes\ viewer\ u\ watched\ video\ v}{Number\ of\ minutes\ in\ video\ v} \tag{4.9}$$

In order to evaluate the effectiveness of the proposed approach, we adopt cross-validation technique and focus on two recommendation tasks.
The first one is the video list recommendation task where we evaluate MAP. The second one is a video rating prediction task in which the accuracy metrics are MAE and $RMSE$.

On the other hand, we conducted experiments in order to study the impact of context elements in improving recommendation. We considered the location, time slot, weekday, weather and occasion in our context-based model. We studied and tested also the ability of our model to solve cold-start and sparsity problems.
We compare our approach with Time-aware Collaborative Filtering (TCF) (Liu, Cao, Zhao & Yang 2010) and Time-Dependent Profile (TDP) (Oh et al. 2012).

4.4.3 Effectiveness of the Context-based Approach

In order to evaluate the effectiveness of the proposed approach, we adopt cross-validation technique. We used the Mean Average MAP (MAP), MAE and RMSE evaluation metrics. We perform 5-fold cross-validation. In each fold, 80% of videos was randomly selected as the training set and remaining 20% as the testing set. The evaluation focused on two

recommendation tasks.
The first one is the video list recommendation task where we evaluate the MAP. Then, we assess if the recommended videos with high probabilities were really viewed by the target user u. We used MAP of top x ($x = 5, 10, 15, 20$) recommendations.
The second one is a video rating prediction task in which the accuracy metrics are MAE and $RMSE$ computed respectively in Equation 2.12 and Equation 2.13 in Section 2.5.2.

Table 4.2 represent the obtained evaluation results of the proposed model in terms of $MAP@x$ ($x = 5, 10, 15, 20$), MAE and $RMSE$.
The different values of $MAP@x$ demonstrate that our model accurately predicts the relevance of videos. Obviously, $MAP@x$ are all more than 60%, which means that most of the relevant videos are ranked among the top x ones.
MAE is 0.616 (less than 100%), which means that there are little differences between the predicted ratings and real ones. The $RMSE$ 0.62 (less than 100%), which means that there are not large emphasized errors.

MAP@5	**MAP@10**	**MAP@15**	**MAP@20**	**MAE**	**RMSE**
0.622	0.641	0.653	0.68	0.616	0.628

Table 4.2: Results on the effectiveness of our approach in terms of MAP@x (x=5, 10, 15, 20), MAEs and RMSEs

4.4.4 Context Elements Impact

We study the impact of each element of the viewing context. In other words, we evaluate the importance of each context element in prediction performance. In this study, we realize two experimental tasks :

(1) The first one is based on removing only one context element : We implement 5 instances of our model. In each instance, we removed one context element. Then, we evaluate $MAP@10$ of each model instance.

The first column of Table 4.3 indicates the context element that was removed in each instance, the second column indicates the new $MAP@10$ calculated after removing it, and the third column represents the impact calculated as follows :

$$Impact = \frac{New\ MAP@10 - Ancient\ MAP@10}{Ancient\ MAP@10} \times 100 \tag{4.10}$$

Table 4.3 shows that all the context elements are essential for the prediction model. The most important ones are location, time slot and occasion. We notice that if we eliminate

Context element	MAP@10	Impact %
Location	0.251	-61
Time Slot	0.28	-56
Week day	0.384	-40
Weather	0.373	-42
Occasion	0.31	-52

Table 4.3: Studying the impact of eliminating each context element on the prediction performance of our context-based model in terms of MAP@10

Context element	MAP@10	Impact %
Location	0.052	-91
Time Slot	0.064	-90
Week day	0.017	-97
Weather	0.03	-95
Occasion	0.022	-96

Table 4.4: Studying the impact of keeping only one context element on the prediction performance in terms of MAP@10

the location or the time slot from the model, MAP decreases by more than 13%.
However, MAP decreases at most by 3% for week day, the weather and the occasion. Therefore, we note that viewers' preferences are more context-sensitive to location and time slot.

(2) The second task consists in keeping only one context element : In each instance, we keep only one context element. Then, we evaluate $MAP@x$ for each model instance.
Table 4.4 reveals the obtained $MAP@10$ and the impacts for each considered instance. The first column appoints the context element that was dropped. The second one indicates the obtained $MAP@10$ after eliminating the corresponding element. The third column indicates the impact of keeping only a context element on MAP (Equation 4.10).
The negative results (impacts) demonstrate that all the context elements are significant for improving the prediction model. The most important ones are location and time slot. Obviously, if we keep only the *location* or the *time slot* from the model, MAP decreases by more than 34%.
However, MAP decreases at most by 39% for week day, the weather and the occasion. Therefore, we note that viewers' preferences are more context-sensitive to *location* and *time slot*.

	MAE	RMSE
10%	0.617	0.62
20%	0.623	0.628
30%	0.628	0.72
40%	0.66	0.73
50%	0.71	0.76
60%	0.73	0.80
70%	0.782	0.815

TABLE 4.5: Evaluating the performance of the proposed context-based approach in terms of MAEs and RMSEs at different sizes of testing set

4.4.5 Resolution of Data Sparsity Problem

The quality of collaborative filtering recommendations is extremely dependent on the sparsity of available data which encounters when there are many missing values. Generally, data sparsity arises due to the fact that users only rate a small portion of items (See. Section 2.4). In this study, we aim to evaluate the effectiveness of our model at various levels of data sparsity. Thus, we randomly divided the viewer/video pairs of our dataset into $n = 10$ groups. We then vary the portion of sets to be considered for training data from 10% to 70%. These sets are randomly selected. Finally, we measured the MAE and the $RMSE$ for each set.

Table 4.5 compares the obtained MAE and $RMSE$ when testing sets from 10% to 70%. As it is clearly shown the MAEs and $RMSE$s increase at a much slower space, and they are not affected by data sparsity.

4.4.6 Resolution of Cold-start Problem

Cold-start refers to the issue that accurate recommendations are expected for new users whereas they often rate only a few items that are difficult to reveal their preferences (See. Section 2.4). We conducted experiments to test the ability of our model to solve viewer cold start recommendation problem. The cold start problem occurs when a new user has no seen videos. We simulate the cold start for each user in the dataset.
We did not take into account the target viewer ratings in the training set. However, we considered the actual contextual information of the user.
The obtained MAE and $RMSE$ for this test are respectively 0.714 and 0.738. The obtained results show the significant improvement of our model in resolving cold-start problem.

4.4.7 Comparison Results

We compare the proposed approach with Time-aware Collaborative Filtering (TCF) (Liu, Cao, Zhao & Yang 2010) and Time-Dependent Profile (TDP) (Oh et al. 2012) in terms of MAP, MAE and $RMSE$. These approaches are described in details in Section /refContextRS. In the following, we explain how we implemented these approaches.

- Time-aware Collaborative Filtering (TCF) (Liu, Cao, Zhao & Yang 2010) : We implemented a collaborative model based a matrix factorization technique. Then, we incorporated the temporal relevance which is associated in our case to the time stamp of viewing a video (time context).

- Time-Dependent Profile (TDP) (Oh et al. 2012) : To implement this approach, we build for each user a time-dependent profile. The construction of the user profile is based on splitting each watch log into time slots and generating a time-dependent profile for each time slot. Therefore, when a recommendation is issued, the system finds the corresponding profile based on the time stamp of the request. In other words, we return the list of videos having the same categories of the those watched in this time stamp.

Figure 4.2 and Figure 4.3 report the MAPs, MAEs and $RMSE$s of all comparison models discussed above. As shown in Figure 4.2, the proposed model significantly outperforms all compared approaches in terms of MAP of top 5 to top 20. It outperforms TCF model by more than 40% and outperforms the TDP model by 36%.

Figure 4.2 presents results on accuracy of rating prediction in terms of $RMSE$ and MAE. We note that our model also outperforms the prediction accuracy of all baseline models in terms of MAE and $RMSE$. Our model outperforms the prediction accuracy of TCF model by more than 0.15 and outperforms the TDP model by 0.20.

In the case of the $RMSE$, the result implies that our recommendation approach performs better personalized recommendation to viewers who are faced with specific contexts, compared to the time-aware models. We note that our recommendation approach, compared to the time-aware models, provides better personalized recommendation to viewers who face with specific contexts. These results demonstrate that, in temporal-based models, the ratings' matrix are more sparse and that using only a specific temporal feature (time context) increases the sparsity problem.

On the one hand, we test the ability of the other approaches to solve data sparsity comparing to the Context-based approach. Table 4.6 compares the MAE and $RMSE$ of our model when testing sets vary from 10% to 70%. As it can be expected, the general behavior of all the approaches is the same. The effectiveness of the other approaches is correlated with

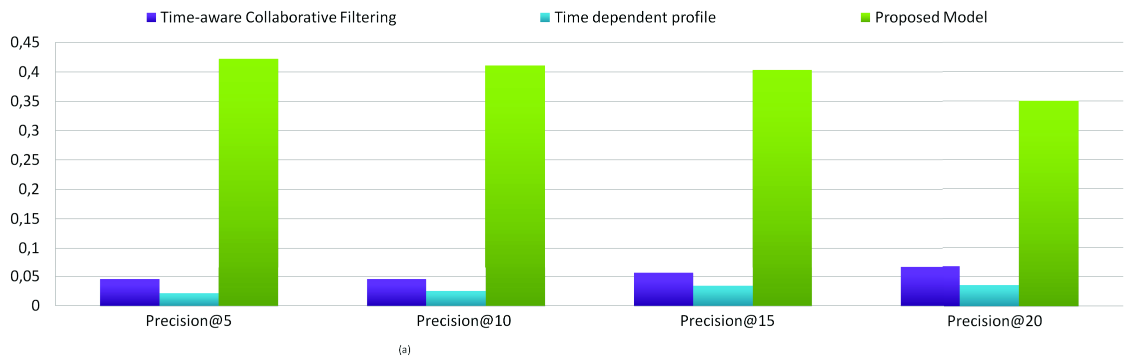

FIGURE 4.2: Comparing the performance of the proposed context-based model with time-aware models in terms of MAP

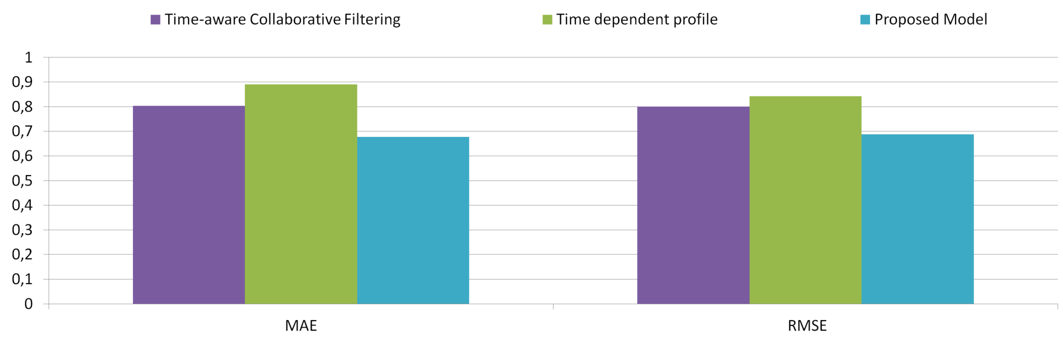

FIGURE 4.3: Comparing the performance of the proposed context-based model with time-aware models in terms of MAEs and RMSEs

	Time-aware Collaborative Filtering (TCF)		Time-Dependent Profile (TDP)		Context-based approach	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
10%	0.792	0.76	0.85	0.84	**0.617**	**0.62**
20%	0.802	0.80	0.89	0.842	**0.623**	**0.628**
30%	0.921	0.83	0.93	0.86	**0.628**	**0.72**
40%	0.93	0.84	0.96	0.87	**0.66**	**0.73**
50%	0.97	0.921	0.97	0.871	**0.71**	**0.76**
60%	0.98	0.98	0.97	0.95	**0.73**	**0.80**
70%	0.98	0.982	0.98	0.98	**0.782**	**0.815**

TABLE 4.6: Comparing the performance of the proposed context-based approach with time-aware models in terms of MAEs and RMSEs at different sizes of testing set

	MAE	RMSE
Time-aware Collaborative Filtering (TCF)	0.803	0.81
Time-Dependent Profile (TDP)	0.89	0.842
Context-based approach	**0.714**	**0.738**

Table 4.7: Comparing our approach effectiveness in resolving context-cold start problem with different baselines in terms of MAEs and RMSEs

the size of the training data.
However, the results show clearly that the MAEs of our model are consistently lower than those of baseline models. In addition, we observe that matrix factorization techniques are highly affected by data sparsity.
For instance, the MAEs of TCF model increases by 0.17 from 0.61 when the testing set increases from 10% to 70%, whereas the MAEs and $RMSE$s of our model increases at a much slower space. During the test of sparsity in TCF model, we noted that the ratings' matrix is sparse since there are missing ratings. Therefore, using only temporal feature (i.e. time) to predict users' preferences alleviates the sparsity problem against missing temporal data.

On the other hand, we compare the ability of the our model with the other approaches to resolve cold-start and sparsity problems. Table 4.7 shows the significant improvement of our model compared to the TCF and TDP models in terms of MAE and $RMSE$. This is due to the fact that CF techniques cannot make recommendation to new viewers because they cannot find similar viewers.
This is due to the fact that CF techniques cannot make recommendation to new viewers because they cannot find similar viewers.
Additionally, our model outperforms $SNMF$ model in terms of MAE and $RMSE$ by more than 0.15. This is due to the fact that matrix factorization techniques can not integrate features other than temporal feature, which increases to solve cold start problem.

4.5 Conclusion

In this chapter, we presented the proposed context-based approach. The aim of our approach is to estimate the relevance of items in respect with the actual context of the user. In our work, we defined the user's context as the set of circumstances related to the actual environment of the user and that may influence his/her preferences. Therefore, we exploit and integrate several contextual information namely time, location, occasion and weekday independently and in generic way. We proposed a probabilistic model in order to predict the relevance of items in respect with these contextual information.

We evaluated the effectiveness of the Context-based approach using real data and implemented our approach in an existing social TV recommender system. We studied the impact of each element of the viewing context on the prediction accuracy.
Moreover, we conducted experiments in order to test the ability of our approach to solve data sparsity and cold start problems. The obtained results through these studies show the effectiveness of our model despite a slower space and missing data.
We compared our approach with *Time-aware Collaborative Filtering (TCF)* and *Time-Dependent Profile (TDP)* which integrate only temporal context. The evaluation provided encouraging results comparing to time-aware methods in terms of MAP, MAE and $RMSE$.
Our approach is different from previous related work in at least three aspects :
- We integrate several context elements in the contrast of previous approaches that used only time and location contexts.
- The additional contextual information are integrated in generic way independently of their complex and different structures.
- We deal with user cold start and sparsity problems using smoothing techniques.

In the previous section, we have tried to integrate social influence among users in the proposed approach. We evaluated the impact of integrating social influence and the context in improving recommendation.

Chapitre 5

Integrating Current Context and Social Influence

Contents

5.1 Introduction

In the previous chapter, we proposed a probabilistic model that integrates the actual context of the viewer in order to improve TV content recommendation. The aim was to integrate several context elements independently and in a generic way.
However, another important context dimension that may play an important role in recommendation. Obviously, social relationships are found beneficial for such RS (Ma 2014, Krishnan et al. 2014, Yang et al. 2014, Groh & Ehmig 2007). Obviously, friends may influence each other and may tend to exhibit similar preferences.

For instance, in the context of TV recommendation, a user might prefer to watch world news (e.g. CNN [1] or BBC [2]) in the morning with colleagues, and movies recommended by friends on weekends.
In our work, we refer to *social context* the influence of the crowd (e.g. interactions with friends or family members presence) around a user.

In this chapter, we propose to integrate the social influence with the proposed context-aware model. Our approach differs from the previous works in at least two aspects :
- We jointly integrate social influence with contextual information.
- We deal with social cold start and social sparsity problems.

We conduct a comprehensive effectiveness of our model on a real dataset. Then, we introduce the social influence realm in RS. In Section 5.2, we formalize the problems behind integrating social aspects in the recommendation process and define the limits of related work. In Section 5.3, we present the proposed social-based model and the whole approach. Then, we describe the conducted experiments on real dataset crawled from a social TV platform, and the obtained results. Finally, we describe our elaborated studies on the effect of social influence and on solving recommender problems.

5.2 Problem Formulation and Positioning

In this section, we present the social realm in RS. Then, we provide some limits and issues behind using social context into the recommender process.

5.2.1 Motivation

Due to the emergence and the prevalence of social networks, users can now rate items, comment and suggest them to friends through social networks. These social interactions

1. www.cnn.com/
2. www.bbc.com/news

are found beneficial for such RS.
Ascribed in Section 2.3.4, Groh et al. (2012) and Groh & Ehmig (2007) found that there is a correlation between items selected by a user and those selected by her friends (i.e. friends share some common interests or friends may influence each other) and propose to exploit these interactions, known as social context, for item recommendations.

This realm commonly appears in taste domains such as TV, music and cinema, where users' preferences are highly influenced by their social environment or their social context. The social context considers the crowd (e.g. friends' interactions) around the user. For example, a viewer may often invite his/her friends to watch TV content together.

As mentioned in Section 2.3.4, many studies (such as (Groh et al. 2012), (Jameson 2004*b*), (Lathia et al. 2008), (Ma 2014), (Krishnan et al. 2014), (Yang et al. 2014) and (Groh & Ehmig 2007)) proved that SF techniques, which exploit the social context in order to improve prediction models. Therefore, these studied the social influence, a social filtering technique, and proved that it may definitely cope with collaborative filtering issues and improve items recommendation.
In our work, we define the *social context* the influence of the crowd (e.g. interactions with friends or family members presence) around a viewer on his preferences.

The questions, that arise here, are : How can we measure the social influence between users and their friends based on social interactions ? Does this social influence depend only on the number of social interactions between such a user and her friends ? How can we trust this measured social influence ? How can we consider and integrate the *social context* in the recommendation process, and particularly, in our probabilistic approach ?

On the other hand, using SF techniques might breed to what is called "'New social connection"' or "'Social cold-start"' problem. As mentioned in Section 2.4, this problem occurs when recommendations may get biased if a user has a very small social network or if she has no connections. Therefore, every social recommendation would be generated based on the activity of just one user.
Another problem occurs when integrating social aspects, called "'Social Sparsity"' problem which occurs where the user have no sufficient connections or interactions.
Consequently, such need remains unmet. How to overcome to social sparsity and social-cold start problems ?

5.2.2 Limits of Social Filtering Techniques

As previously shown (see. Section 2.3.4), many works (Zhu et al. 2011, Brocco, Groh & Forster 2010, Brocco & Groh 2009, Liu & Aberer 2013, Ma et al. 2011) have been proposed to incorporate the social influence into recommendation models. Unfortunately, most of these approaches used commonly matrix factorization techniques incorporate social influence in heuristic way.

Despite the fact that matrix factorization techniques are commonly used by almost works, they are considered as the most complex techniques. This is due to their major drawback related to the non-convexity scheme. As a result, there is in general no algorithm that is guaranteed to compute the desired factorization.

In addition,, we are in the presence of different possible contexts that the user may experience (e.g. interactions). Although the wealth of information gathered by users in social networks, most existing systems did not exploit real-time users' interactions (e.g. social interactions on TV shows with friends) to predict their preferences.
The proposed approaches are restricted to exploit similar ratings with other users in a static way. Matrix factorization techniques fail to consider the structure in the data, such as the nature of interactions between users and their friends (Aleksandrova et al. 2014, Lazar & Doncescu 2009, Porteous et al. 2010).

Moreover, when the formed groups (or networks) are large, the strong assumption of pair-wise influence in a group may not be true. The existing approaches used by these systems have also few considerations about solving recommender problems (e.g. no information about a new user known as "cold-start problem") which are the major cause of reducing RS performance. Then, users' preferences may change according to his/her interactions with friends. Obviously, a viewer might like horror movies while being with his friends, whereas, he might prefer comedy when being with his family.

On the other hand, the matrix factorization techniques fail to jointly exploit contextual information and social information.
Consequently, an effective recommender system must exploit the rich environment of users, capture all their social behaviors, and analyze all the user context and its change by considering cold start and sparsity problems.

Ref.	Content-based filtering	Collaborative-filtering	Social-filtering	Matrix factorization	Integrating temporal or location factor	Integrating other context elements	Sparsity problem	Cold start problem
(Liu, Cao, Zhao & Yang 2010, Gantner et al. 2010)	−	+	−	+	+	−	−	[illegible]
Martinez2009	−	+	−	+	+	−	−	[illegible]
Turrin et al. (2014), Oh et al. (2012)	+	−	−	−	+	−	−	[illegible]
(Pyo et al. 2013)	−	+	−	−	+	−	−	−
(Barragáns-Martínez et al. 2009)	+	+	−	+	−	−	−	−
(Chang et al. 2013)	−	+	−	−	−	−	−	−
(Pálovics et al. 2014, Forsati et al. 2015)	−	+	+	+	−	−	−	−
(Ma 2013, Liu, Cao, Zhao & Yang 2010)	−	+	+	+	−	−	−	−
(Zhao et al. 2013, Chaney et al. 2015, Ye et al. 2012)	+	−	+	+	−	−	−	−
(Macedo et al. 2015)	+	−	+	−	+	−	+	+
Context and Social-based Approach	−	+	+	−	+	+	+	+

Table 5.1: Comparison of Context and Social-based Approach against related work approaches.

The comparison of related work approaches is described in Table 5.1.

5.2.3 Research Questions

In this research, we define a new social-based model that estimates the relevance of items based using social influence between users and their friends. The social influence technique is based not only on the number of social interactions between the target user and her friends, but also on the response of the user towards these interactions (or a trust-measure).
We proposed also a probabilistic approach that unifies jointly the proposed context-based and social-based models in order to improve items recommendation. In particular, we propose a probabilistic approach that aims to predict the relevance of items based on the user's current context and the social influence.

Accordingly, we establish our key research questions in this work :
- How can we measure the social influence based on the interactions between users in order to estimate items relevance ?
- How can cope with social sparsity and social-cold start problems ?
- How can we jointly exploit the contextual information and the social influence in order to improve items recommendation ? and How can we unify the context-based and the social-based models into one model ?

5.3 Jointly Leveraging Context-based and Social Influence Models

In the previous chapter, we proposed a probabilistic approach that estimates the relevance of items in respect with the current context of the user. In this Section, we propose a generative model that captures quantitatively social influence between friends and employs social influence to mine the personal preference of users. Then we describe how we jointly incorporate the context-based model previously proposed with the social influence in order to improve relevance prediction.

We restate the same example described in Section 4.3.
We consider a graph $G = (N, L)$ where N represents nodes (Viewers and Videos) and L represents links between nodes. We refer to the set of viewers as U and to the set of videos as V. $V(u)$ represents the set of videos viewed by user u and $U(v)$ represents the set of users which have viewed video v.

L may link viewer u and a video v or viewer u with his friend f. In the first case, L represents the context in which user u has viewed video v. Let C be the set of all viewing contexts,

$c_{uv} \in C$ is the viewing context of u in which he has viewed video v. A viewing context c_{uv} is represented by a set of properties $c_{uv}\{c_{1uv},..,c_{muv}\}$ including the *time slot*, *the location*, *the week day*, *the weather* and *the occasion*. The second link type refers to viewer's social network.

The social network of a viewer u consists of all other viewers (e.g. friends, family members or colleagues) whom u watches or interacts with (receive or make a recommendation) while watching such a video v. We refer to the set of viewers the user u interacts with as $F(u)$. We represent each interaction between viewer u and his friend f by an edge I_{uf} labeled by the nature of interaction (*recommend to*, *watch with*, *tweet the same show* or *tag a friend in a show page*) and the identifier of the video they interact on.

The problem of recommending new videos (previously unseen) to a user u can be addressed by estimating the probability for u to select an item i (i.e. $Pr(i|u)$). Candidate items with the highest aforementioned probability are recommended to u. We note that $Pr(i|u)$ can be computed by estimating the relevance of the item i in respect not only the current context of the user u, but also the his/her social context.

Therefore, the goal of our probabilistic approach is to estimate the relevance of the target video v for the target user u given his current context $c_u\{c_{1u},..,c_{mu}\}$ and his social context. In this case, we define the *social context* as the influence of the preferences (or ratings) of the friends of the viewer on her preferences.

We argue that social influence can provide useful information to predict users' preferences. The aim is to model the potential effect of social relationships on user' ratings. Therefore, we extend the context-based probability $P(r_{uv} = k|c = c_u$ by integrating the social influence. We aim to predict $P(r_{uv} = k|c = c_u, SI_{F(u)v})$ which is the conditional probability that the target viewer u's rating on video v equals the to the value k, given the current context c_u of viewer u and the social influence $SI_{F(u)v}$. $SI_{F(u)v}$ represents the social influence of $F(u)$' ratings.

We assume that the social influence and the viewing context are independent. Then, we used naive Bayes assumption which simplifies the correlation between the viewing context and the social influence. Thus, after using Bayes rule, $Pr(r_{uv} = k|c = c_u, SI_{F(u)v})$ can be written as follows :

$$Pr(r_{uv} = k|c = c_u, SI_{F(u)v}) \propto \frac{Pr(r_{uv} = k|c = c_u) \times Pr(r_{uv} = k|SI_{F(u)v})}{Pr(SI_{F(u)v})} \tag{5.1}$$

We assume that $Pr(SI_{F(u)v})$ is uniform. Then, $Pr(r_{uv} = k|c = c_u, SI_{F(u)v})$ can be estimated as follows :

$$\begin{aligned} & Pr(r_{uv} = k|c = c_u, SI_{F(u)v}) \\ \propto \; & Pr(r_{uv} = k|c = c_u) \times Pr(r_{uv} = k|SI_{F(u)v}) \end{aligned} \tag{5.2}$$

As described in Section 4.3, c_u consists of the current time slot, location, week day, weather and occasion. The aim was to estimate the relevance of video v for user u given his current context c_u. $Pr(r_{uv} = k|c = c_u)$ estimates the conditional probability that the rating r_{uv} given by user u on video v equals the to the value k given the u's current viewing context c_u. This probability represents viewer u preferences given his current context c_u.
It can be estimated as the relevance of video v for viewers having approximately the same viewing context than u.

$$\begin{aligned} & Pr(r_{uv} = k|c = c_u\{c_{1u}, \ldots, c_{mu}\}) \\ \propto & \prod_{i=1}^{m} Pr(c_i = c_{iu}|r_v = k) \times Pr(r_v = k) \end{aligned} \tag{5.3}$$

On the other hand, we argue that the ratings of friends can provide useful information to predict users' preferences. The aim is to model the potential effect of social relationships on user ratings. $Pr(r_{uv} = k|SI_{F(u)v})$ is the probability that viewer u gives a rating equals to k to v given the influence of the $F(u)v$'s ratings. We consider that social influence $SI_{F(u)v}$ measures the effect of the preferences of u's friends on the relevance of video v. We detail in the next section the estimation of the two probabilities. In the following subsections, we present the proposed social influence model.

5.3.1 Social Influence based on Friends' Ratings

We aim to estimate the video relevance by considering the social influence. Hence, we simulate the process that how viewer u picks video v, considering how his friends F influence the relevance of v for the viewer u.

$Pr(r_{uv} = k|SI_{F(u)v})$ is the conditional probability that u' s rating on video v equal to value k given the social influence. We assume that friends may have similar preferences and similar ratings. $Pr(r_{uv} = k|SI_{F(u)v})$ can be estimated by considering friends' ratings on the video v against their ratings that are equal to value k on other videos v' (Equation 5.4).

$$Pr(r_{uv} = k|SI_{F(u)v}) \propto \frac{\sum_{i=1}^{|F(u)\cap U(v)|} |r_{f_i v} = k|}{\sum_{i=1}^{|F(u)\cap U(v)|} \sum_{v'_i} |r_{f_i v'} = k|} \tag{5.4}$$

$Pr(r_{uv} = k|SI_{F(u)v})$ is the conditional probability that there is social influence when u's rating on v equal to value k. It could be estimated as the mean average of the sum of the number of u's friends ratings on video v equal to k against the sum of the number of the friends' ratings equal k. That is $|F(u) \cap U(v)|$ is the number of the friends of user u who viewed video v, $|r_{f_i v} = k|$ is the number of ratings of f_i on video v that equal to k, and $|r_{f_i} = k|$ is the number of ratings of f_i that equal to k.

5.3.2 User-User Social Trust

We argue that information from social relationships has potential influence in viewers' preferences. Thus, we assume that social influence depends not only on friends' ratings but also on social similarity between users. The user-friend social similarity information may be established based on social interactions between users and their friends. Therefore, we argue that social similarity could be integrated in Equation 5.5 using similarity measure Sim_{uf}.

$$Pr(SI_{F(u)v}|r_{uv} = k) = \frac{\sum_{i=1}^{|F(u)\cap U(v)|}(|r_{f_i v} = k| \times Sim_{uf_i})}{\sum_{i=1}^{|F(u)\cap U(v)}(|r_{f_i} = k| \times Sim_{uf_i})} \tag{5.5}$$

The similarity Sim_{uf_i} between user u and his friend f_i could be considered as the exponential of the distance between their ratings. Sim_{uf_i} is related to similarity and interpersonal interactions between u and f and the degree of agreement between them. Sim_{uf_i} is measured using Equation 5.6.

$$Sim_{uf_i} = e^{-d_{uf_i}} \tag{5.6}$$

The distance d_{uf_i} between u and f is the difference between their ratings on same videos they interact with. It is measured as the aggregation of absolute values of the difference between their ratings on each video they interact on.

$$d_{uf_i} = \sum_{j=1}^{|I_{uf_i}|} |r_{uv'_j} - r_{f_i v'_j}| \tag{5.7}$$

I_{uf_i} is the social interaction between user u and his friend f_i (i.e. receive, make recommendation or tag on a TV program page) and v'_j is the video they interact on. $|I_{uf_i}|$ refers to

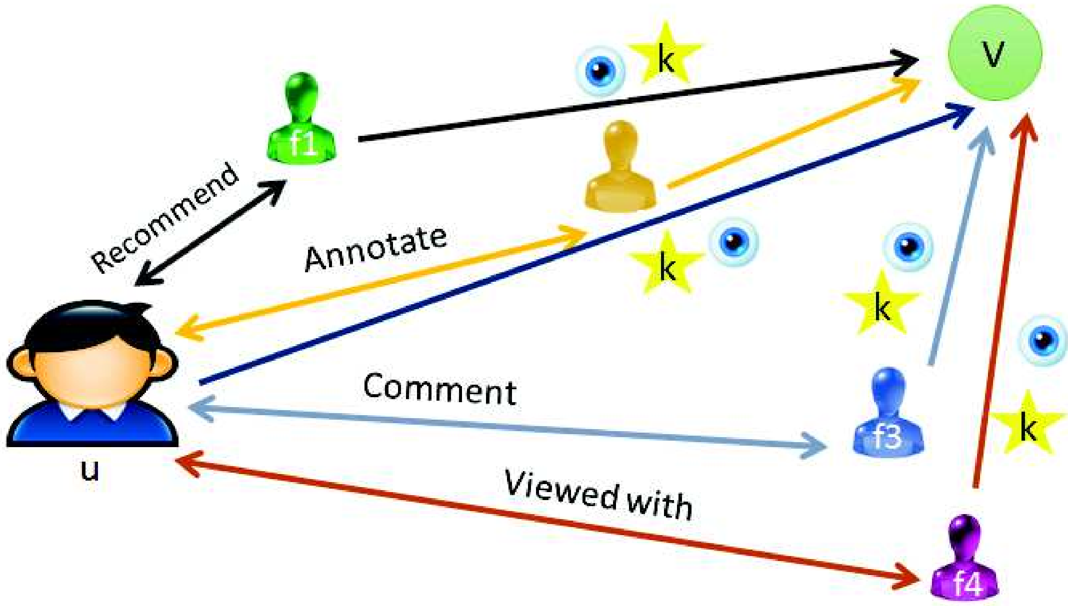

FIGURE 5.1: An example showing interactions with different natures between user u and her friends on video v with ratings equal to k. These interactions indirectly influence the relevance of video v to user u (she viewed v with the same rating after these interactions).

the number of interactions between f_i and u and $|r_{uv'_j} - r_{f_i v'_j}|$ is the absolute value of the difference between ratings of f_i and u given an interaction on video v'_j.

However, $|F(u) \cap U(v)|$ could be null where no friends' views for video v. Therefore, $Pr(SI_{F(u)v}|r_{uv} = k)$ could be equal to 0. In this case, it is required to not assign low probability (zero probability) to no found friends watching the target video, or strong (probability = 1) probability where all friends have watched the target video.

Smoothing strong or low probabilities is extremely significant when estimating a model based on a limited amount of data (e.g. no associated ratings). The term smoothing refers to adjusting probability in order to produce more accurate estimation and to solve data sparsity (Zhai & Lafferty 2001, Chen & Goodman 1996). The name smoothing is centered around the fact that these techniques tend to make distributions more uniform by adjusting probabilities and improving the accuracy of the model.
There are several works in the literature that focused on the issue of smoothing accuracy, such as Jelinek-Mercer (Jelinek & Mercer 1980), Dirichlet (Smucker & Allan 2005), and Laplace (Chandra & Gupta 2011).
A smoothing method may be assign a probability proportional to the ratings occurrence in the collection to unseen ratings (Smucker & Allan 2005). Dirichlet is a conjugate prior for Multinomial distribution, it means that the prior has the same functional form as the likelihood. μ is a parameter used to control smoothing.

Then, in order to avoid zero frequency problem and to penalize lower similarity measures, we used Dirichlet smoothing technique Smucker & Allan (2005). It determines also the amount of smoothing according to friends number and to come more from an implicit prior favoring important social influence. In our work, Dirichlet smoothing technique advantage is that we can integrate additional information in order to avoid zero frequency problem. In this case, we integrate $Pr(r_v = k)$ which measures the probability that the video v was rated with value k. The Equation 5.8 can be reformulated as :

$$\begin{aligned} & Pr(SI_{F(u)v}|r_v = k) \\ = & \frac{\sum_{i=1}^{|F(u)\cap U(v)|} |r_{f_i v} = k| \times Sim_{uf_i} + \mu \times Pr(r_v = k)}{\sum_{i=1}^{|F(u)\cap U(v)|} |r_{f_i} = k| \times Sim_{uf_i} + \mu} \end{aligned} \tag{5.8}$$

Where μ is a parameter used to control smoothing probabilities.

5.4 Experiments

In this section, we conduct an effectiveness evaluation using real data collected from Pinhole social TV platform. This dataset includes viewer-video accessing history and viewers' friendship networks. In addition, we collect contextual information for each viewer-video accessing history captured by the platform system.

In our evaluation, we adopt the time-dependent profile approach (Oh et al. 2012) and the approaches proposed by Liu, Cao, Zhao & Yang (2010) as baseline models. Besides, we propose to study the effectiveness of each element of the viewing context (i.e. location, time slot, weekday, weather and occasion) considered in our context-based model. We propose also to study the social influence (SI) impact on the effectiveness of our model.

The aims of these experiments are :
1. Evaluating the effectiveness of the proposed context-based and social influence approach ;
2. Evaluating the impact of contextual information after integrating social context ;
3. Evaluating the effectiveness of jointly integrating the social influence with context-based approach ;
3. Evaluating the effectiveness of the social trust in improving the recommendation process ;
5. Evaluating the ability of the proposed approach to solve context cold start and sparsity problems ;
6. Evaluating the effectiveness of the proposed approach by drawing up a comparison study between the proposed approach and some approaches of state-of-the art.

MAP@5	MAP@10	MAP@15	MAP@20	MAE	RMSE
0.622	0.63	0.653	0.68	0.71	0.73

Table 5.2: Results on the effectiveness of the Context and Social-based Approach in terms of MAP@x (x=5, 10, 15, 20), MAEs and RMSEs

5.4.1 Evaluation Protocol

In order to conduct our experiments, we used a subset of Pinhole[3] data. The used dataset is described in Section 4.4.1. The dataset includes 3, 440, 218 social interactions and the average number of friends per user is 124. Throughout social networks and TV programs pages, viewers can interact (e.g. recommend, tag, comment or share) on TV contents with their friends.

5.4.2 Effectiveness of the Context and Social-based Approach

In order to evaluate the effectiveness of the Context and Social-based Approach, we adopt the same technique, recommendation tasks and evaluation metrics described in the previous Chapter (See. Section 4.4).
Based on the empirical study of this work, μ= 500 is the best setting for estimating $Pr(SI_{F(u)v}|r_v = k)$.

Table 5.2 represents the obtained evaluation results of the Context and Social-based Approach in terms of $Precison@x$ ($x = 5, 10, 15, 20$), MAE and $RMSE$.
The different values of $MAP@x$ demonstrate that our model accurately predicts the relevance of videos. Obviously, even after integrating social influence model, $MAP@x$ are all more than 60%, which means that most of the relevant videos are ranked among the top x ones.
MAE is 0.71 (less than 100%), which means that there are little differences between the predicted ratings and real ones. The $RMSE$ 0.73 (less than 100%), which means that there are not large emphasized errors.

5.4.3 Impact of Context Elements After Integrating Social Influence

We use here the statements of the study enunciated in Section 4.4.4 to highlight the impact of each element of the context but with considering social influence. In other words, we evaluate the importance of each context element in prediction performance of the whole approach.

3. www.pinhole.tn

We implement 5 instances of our model. In each instance, we removed a context element. Then, we evaluate *MAP@10* of each model instance. Table 5.3 indicates that all the context

Removed context element	**MAP@10**	**Impact**
Location	**0.451**	**39,68**
Time Slot	**0.480**	**31,25**
Week day	0.584	7,8
Weather	0.573	9,94
Occasion	0.581	8,43
With all context elements	0,641	

TABLE 5.3: Studying the impact of eliminating each context element on the prediction performance in terms of MAP@10 after the integration jointly the context and the social influence

elements are essential for the prediction model. The most important ones are location and time slot. Obviously, if we eliminate the location or the time slot from the model, MAP decreases by more than 16%. However, MAP decreases at most by 7% for week day, the weather and the occasion. Therefore, we note that viewers' preferences are more context-sensitive to location and time slot. In the case of the *RMSE*, the result implies that our recommendation approach, compared to the time-aware models, performs better personalized recommendation to viewers who are faced with specific contexts.

5.4.4 Social Influence Study

In this section, we study the role of social influence among viewers and their friends in improving prediction of TV contents relevance.

We compared the effectiveness of our model with and without incorporating similarity measure. When comparing our model with and without considering social influence, we note that the performance of using similarity measure among viewers and their friends is considerably better.
The experimental results in Table 5.3 show that using similarity measure is very effective at improving traditional recommender techniques. *MAP@10* increases from 43,1% to 64%. The *MAE* increases from 71,13% to 71,16%, which is only a 3% difference. The increase of *MAE* is due to errors that may inescapably occur when considering user-friend similarity measure.

5.4.5 Resolution of Data Sparsity Problem

The quality of collaborative filtering recommendations is extremely dependent on the sparsity of available data which encounters when there are many missing values. Generally, data

	MAP@10	MAE
With similarity measure	**0,641**	0,716
Without similarity measure	0.431	**0,713**

Table 5.4: Our approach effectiveness with and without considering social influence in terms of MAP@10 and MAE

	MAE	RMSE
10%	0.617	0.62
20%	0.716	0.728
30%	0.713	0.80
40%	0.71	0.81
50%	0.81	0.81
60%	0.81	0.813
70%	0.83	0.82

Table 5.5: Evaluating the performance of the Context and Social-based Approach in terms of MAEs and RMSEs at different sizes of testing set

sparsity arises due to the fact that users only rate a small portion of items (See. Section 2.4). In this study, we aim to evaluate the effectiveness of our model at various levels of social sparsity. Thus, we randomly divided the viewer/friends pairs of our dataset into ten groups. Then, we randomly selected n sets as testing set and the rest as training set. We measured the *MAE* and the *RMSE* for each value of n.

Table 5.4 compares the *MAE* and *RMSE* of our model when testing sets vary from 10% to 70%.

Table 5.7 compares the obtained MAE and $RMSE$ when testing sets from 10% to 70%. As it is clearly shown the MAEs and $RMSE$s increase at a much slower space, and they are not affected by data sparsity.

5.4.6 Resolution of Cold-start Problem

Cold-start refers to the issue that accurate recommendations are expected for new users whereas they often rate only a few items that are difficult to reveal their preferences (See. Section 2.4).

Now we conduct experiments to test the ability of our model to solve viewer cold-start recommendation problem. In our case, the social cold-start problem occurs when the user has no connected social network (no friends).

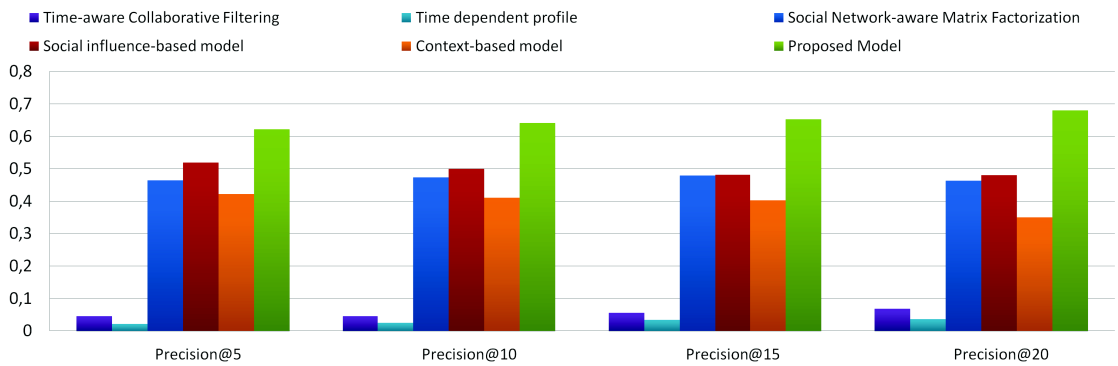

FIGURE 5.2: Comparing our approach performance with different baselines in terms of MAP

We simulate the social cold-start for each user in the dataset based on the following experiments settings : 1) We assume that the target user has no friends. So, we did not take into account the ratings of her friends on the training set. 2) However, we considered the actual context of the target user. We conducted experiments to test the ability of our model to solve viewer cold start recommendation problem. The cold start problem occurs when a new user has no seen videos. We simulate the cold start for each user in the dataset.
We did not take into account the ratings of the friends of the target user in the training set. However, we considered the actual contextual information of the user.
The obtained MAE and $RMSE$ for this test are respectively 0,724 and 0,738. The obtained results show the significant improvement of our model in resolving cold-start problem.

5.4.7 Comparison Results

We compare the Context and Social-based Approach with with time-dependent profile approach (Oh et al. 2012) and the approaches proposed by Liu, Cao, Zhao & Yang (2010) which integrates separately the temporal context and the social network in recommendation process. These approaches are described in details in Section 3.5.

In order to highlight the importance of integrating jointly the viewing context and the social influence, we evaluated separately the *Social influence-based model* and the *Context-based model* of our approach.

Figure 5.2 and Figure 5.3 report MAP, MAE and $RMSE$ of all comparison models discussed above.

As shown in Figure 5.2, the Context and Social-based Approach significantly outperforms all compared approaches in terms of MAP of top 5 to top 20. It outperforms social network-based model by 20% which demonstrates the effectiveness of incorporating user-friend similarity measure based on social interactions to improve recommendation performance.

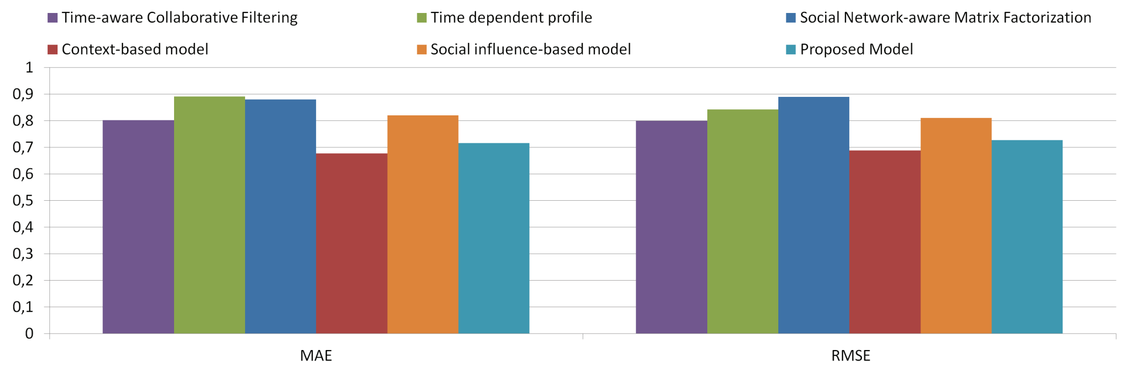

Figure 5.3: Comparing our approach performance with baseline models in terms of MAE and RMSE

	TCF		TDP		SNMF		Context and Social-based Approach	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
10%	0,792	0.76	0.85	0.84	0,75	0,70	**0.61**	**0.62**
20%	0.802	0.80	0.89	0.842	0,88	0,82	**0.716**	**0.728**
30%	0.921	0.83	0.93	0.86	0,89	0,87	**0.713**	**0.80**
40%	0.93	0.84	0.96	0.87	0,91	0,87	**0.71**	**0.81**
50%	0.97	0.921	0.97	0.871	0,96	0,918	**0.81**	**0.81**
60%	0.98	0.98	0.97	0.95	1,108	0,93	**0.81**	**0.813**
70%	0.98	0.982	0.98	0.98	1,107	1,06	**0.83**	**0.82**

Table 5.6: Comparing our approach effectiveness with different baselines in terms of MAEs and RMSEs at different sizes of testing set

Additionally, the context-aware model outperforms time-aware models by more which highlights the importance of integrating several contextual information other than temporal features.

Figure 5.3 presents results on accuracy of rating prediction in terms of *RMSE* and *MAE*. We note that our model outperforms Social influence-based model by more than 10% which demonstrates that exploiting jointly contextual and social information is better than considering them separately.

From Figure 5.3, we note that our model improves also the prediction accuracy of all baseline models in terms of *MAE* and *RMSE*. Our model improves the prediction accuracy of *TCF* model by more than 40% and outperforms the *TDP* model by 26%.

We find that using social influence indeed improves recommendation performance comparing to time-aware models.
In this study, we aim to test the ability of the other approaches to solve social and context sparsity comparing to our approach.

	MAE	RMSE
Time-aware Collaborative Filtering (TCF)	0,803	0,81
Time-Dependent Profile (TDP)	0,89	0,842
Social Network-aware Matrix Factorization (SNMF)	0,97	0,96
Context and Social-based Approach	**0,724**	**0,738**

TABLE 5.7: Comparing our approach effectiveness in resolving social-cold start problem with different baselines in terms of MAEs and RMSEs

Table 5.6 compares the *MAE* and *RMSE* of our model when testing sets vary from 10% to 70%. As it can be expected the general behavior of these approaches is the same. The effectiveness of the other approaches are correlated with the size of the training data. However, the results show clearly that the *MAE*s of our model are consistently lower than those of the baseline models. In addition, we observe that matrix factorization techniques are highly affected by data sparsity. For instance, the *MAE*s of *TCF* model increases by 18,8% from 0.79 when the testing set increases from 10% to 70%, whereas the *MAE*s and *RMSE*s of our model increases at a much slower space.

As showed in Table 5.7, the resulting *MAE* is 72,4% and *RMSE* is 73%. The results show significant improvement compared with the *TCF* and *TDP* model in terms of *MAE* and *RMSE*. This is due to the fact that *CF* techniques can not make recommendation to new viewers because they can not find similar viewers. Additionally, our model outperforms *SNMF* model in terms of *MAE* and *RMSE* by more than 20%. This is due to the fact that matrix factorization techniques can not integrate features other than ratings' similarity to solve cold start problem.

5.5 Conclusion

In the previous chapter, we proposed a context-based approach which estimates the items relevance in respect with the current context of the user. A proposed probabilistic model integrates several context elements in order to improve prediction and to recommend more personalized items. The proposed probabilistic approach enables integrating any additional information in a generic way and independently of their structures. The Context and Social-based Approach succeeded also to cope with context-cold start and data sparsity problems.

In this chapter, we proposed a social-based approach that predicts the ratings of the target user based on the social influence of the ratings of her friends on the relevance of the target item. The social influence is estimated based on the social interactions between the target user and her friends. However, this estimation is not based only on the number of the interactions between them but also on a trust measure. The trust measure is based on the response of the target user towards these interactions.

We proposed also an approach that unifies the introduced context-based and the social-based models into one model. This approach aims to estimate the items relevance based simultaneously on the current context of the target user and the social influence around her.

In addition, we have collected data on real viewing histories and social interactions crawled from an existing Social TV platform. We conduct several experiments in order to evaluate the effectiveness of the Context and Social-based Approach in terms of MAP, MAE and $RMSE$. We tested also the ability of our approach to solve data sparsity and cold-start problems.

We compare our approach to time-aware approaches and a social-based approach. In our experimental studies, the Context and Social-based Approach achieves the best results comparing to different baselines. As social influence is a hidden and not directly observable factor, its incorporation in a predictive model is considered more challenging than incorporating items' contents. In the sparsity and cold start tests, our approach returns consistently accurate predictions at different values of testing tests thanks to the used smoothing techniques.

The encouraging results open several future directions such as enriching TV shows' profiles based on keywords related to viewers interactions and using other evaluation metrics such as the serendipity and the diversity in order to better evaluate recommendation accuracy.

Chapitre 6

Implementation

Contents

6.1 Introduction

Our research work falls within MobiDoc program [1], which is hosted by PASRI [2] (Project Supporting Research and Innovation Systems) and funded by the European Union.
This program has allowed us to integrate the new professional environment, and to carry out our experiments in the Tunisian company Pinhole.

As explained in the next chapters, we we conduct an effectiveness evaluation using real data collected from Pinhole [3] social TV platform. This dataset includes viewer-video access history and viewers' friendship networks. It enables the users to watch VOD (Video On Demand) content and to interact with their friends on shows and TV programs.
We expanded an existing social TV system based on the proposed approach in order to perform the personalized content recommendation in a context-aware environment.

In Section 6.2, we present the system architecture of the platform Pinhole. In Section 6.3, we demonstrate how we contribute in database conception, and the transformation of data to graph-based data. In Section 6.4, we present and describe some user interfaces on Pinhole Platform.

6.2 System Architecture

As highlighted in Figure 6.6, the main concern of Pinhole is to provide popular TV content to the viewers, by collaborating with boxes of Tunisian, Arab, Turkish and international production in order to integrate their movie, show or series in Pinhole.

Figure 6.1 illustrates the enhanced client-server architecture, in which the context-based approach was used to develop the recommendation module.

The work flow of context-aware recommendation is that the server side of the system uses the recommendation module to produce a candidate list from the currently available videos. Then the system exploits the context module to re-rank the candidate videos, as shown in Figure 6.1. The context module consists of the actual context elements of the actual viewer. The re-ranking is achieved based a context-aware model. As described in Section 4.3, we consider contextual information (time, location, occasion, weekday) collected on the client side. This recommendation list is send to the user for his reference regarding relevant video selection.

1. http ://www.pasri.tn/mobidoc-doctorant
2. http ://www.pasri.tn/pr%C3%A9sentation
3. www.pinhole.tn

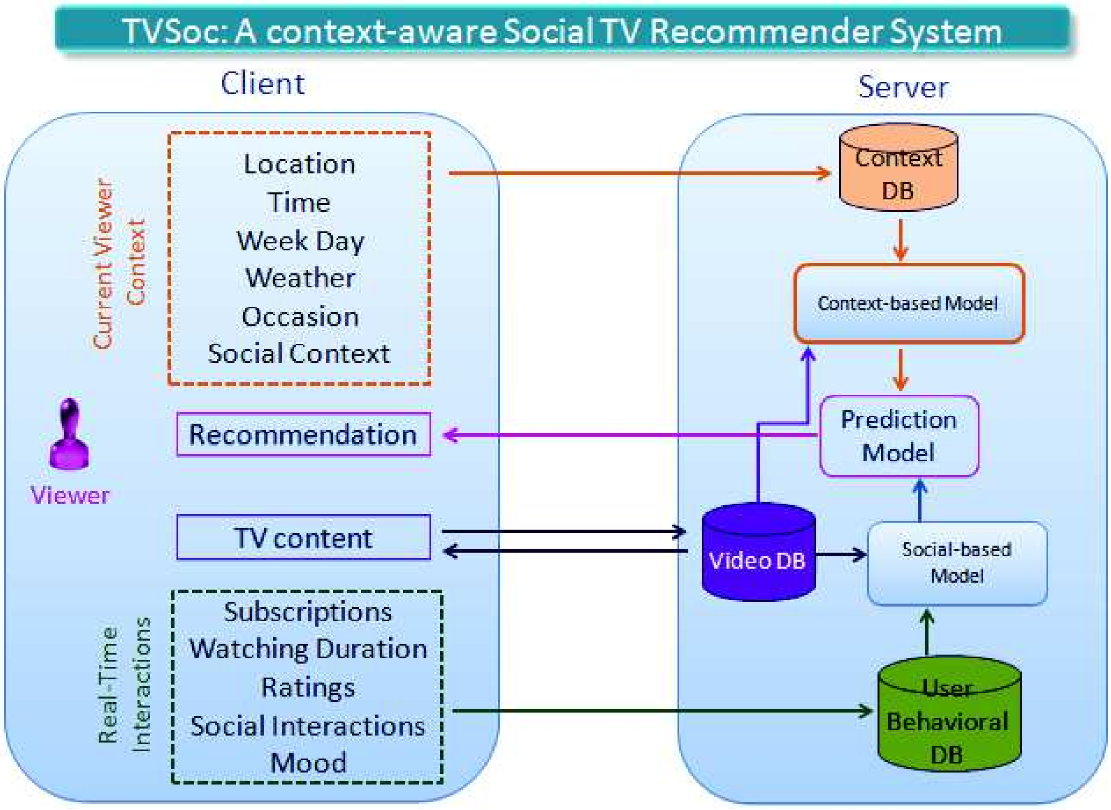

FIGURE 6.1: The architecture of the proposed context-based TV recommender system (TVSoc)

To identify the context, system embedded in the most popular client devices (e.g. Windows NT, Mac OS, Linux and Solaris for non-mobile devices and iPhone Mac OS, and Linux for mobile devices) in order to recognize the type of the user device accordingly.

The GPS or GSM cellular system can be used with an electronic map to provide detailed location context. To identify the location, the system integrates the positioning-system-based location and time information into our system in our current implementation.
The weekday and the occasion are captured from the viewer schedule (e.g. anniversary, workout, party, and meeting). The weather is captured according to the detected location and the time slot.

Consequently, the system collects contextual information for each viewer-video access history captured by the platform system. The platform system captures and records the last contextual information that the viewer faced while watching such a video.

Once contextual and social information are collected and modeled, the system appeal the prediction model to make recommendation.

	usr_id bigint	ext_id bigint	show_id bigint	uew_viewdate timestamp without time zone	evt_id bigint
1	122	1710	93495	2013-07-09 21:45:09.537	75226
2	122	1009	92664	2013-07-09 21:45:56.626	75132
3	122	2117	93898	2013-07-09 23:17:44.599	75328
4	122	1710	93495	2013-07-10 07:13:40.523	75226
5	122	2131	93912	2013-07-10 07:34:11.7	75328
6	122	2131	93912	2013-07-10 07:34:18.234	75328
7	122	2117	93898	2013-07-10 07:39:52.124	75328
8	122	2117	93898	2013-07-10 07:39:56.804	75328

FIGURE 6.2: An example of an entity in relational databases of Pinhole TV platform

6.3 Graph-based Data Model Transformation

The data are represented in relational data base. For example, the shows watched by each viewer are saved in the table $pin_{u}ser_{e}xtra_{w}atch$ as described in Figure6.2 and Figure 6.3 .

In relational databases, references to other rows and tables are indicated by referring to their primary-key attributes via foreign-key columns. For example, to identify the category of each TV program watched by user 122, we have to use the foreign key to access to the other table. This is achievable with constraints, but only when the reference is never optional. Joins are computed at query time by matching primary- and foreign-keys of the many rows of the to-be-joined tables.

Unfortunately, the use of relational database will make the operations more compute- and memory-intensive and have an exponential cost. In addition, relational databases do not intrinsically contain the idea of fixed relationships between records. However, related data is linked to each other by storing one record's unique key in another record's data.

In order to simplify the representation and the exploitation of data, we transformed data form relation to graph-based data base.
Figure 6.4 represents an sample of a graph-based data set.

We used Neo4j [4] which is an open-source $NoSQLgraph$ database implemented in Java and Scala. $Neo4j$ implements the Property Graph Model efficiently down to the storage level.

The advantages of graph-based database are :

4. www.Neo4J.com

evt_id bigint	cmt_comment text	cmt_creationdate timestamp without time zone	version bigint	cmt_isdelete boolean	show_id bigint	cmt_parent_ bigint	cmt_parent_ character va
78336	ou est l'episode 1 2 3 ?	2014-08-28 20:30:00.632	1	FALSE	111052		
76988	انـــــــــــــــد	2014-08-28 20:35:55.174	1	FALSE	101967	6173	twitter
76988	انـــــــــــــــد	2014-08-28 20:38:24.04	1	FALSE	101967	6173	twitter
76988	و اند كمان	2014-08-28 20:40:33.708	1	FALSE	101967	5203	twitter
76988	و اند زعلان؟	2014-08-28 20:41:13.675	1	FALSE	101967	5203	twitter
76067	cool http://www.pinhole.tn/fariha-2/8062?t=34	2014-08-31 19:06:03.348	1	FALSE	101231		
77074	rien http://www.pinhole.tn/al-agha-al-saghir/9913?t	2014-09-02 12:33:24.436	1	FALSE	102883		

Figure 6.3: Some interactions made by users on Pinhole TV platform presented in the relational Database

- Performance

Graph databases improve performance by several orders of magnitude for intensive data relationship handling. In relational databases, relationship queries will come to a grinding halt as the number and depth of relationships increase. However, graph database performance stays constant even as your data grows over the time.

- Flexibility

In graph databases, data architect teams move at the speed of business because the structure of a graph model flexes as applications change. Rather than exhaustively modeling a domain ahead of time, data teams can add to the existing graph structure without endangering current functionality. This allows us to easily integrate additional context elements.

- Agility

Developing with graph databases aligns perfectly with agility, test-driven development practices. This allows our graph database to evolve in step with any changing applications requirements. Modern graph databases are equipped for frictionless development and graceful systems maintenance.

The property graph contains connected entities (the nodes) which can hold any number of attributes. In addition to contextualizing node and relationship properties, labels may serve to attach metadata (index or constraint information) to certain nodes.

The steps adopted for the transformation of our dataset are :

- Each entity table is represented by a set of nodes.

- Each row in an entity table is a node. In our case, each viewer and each video is represented by a node.

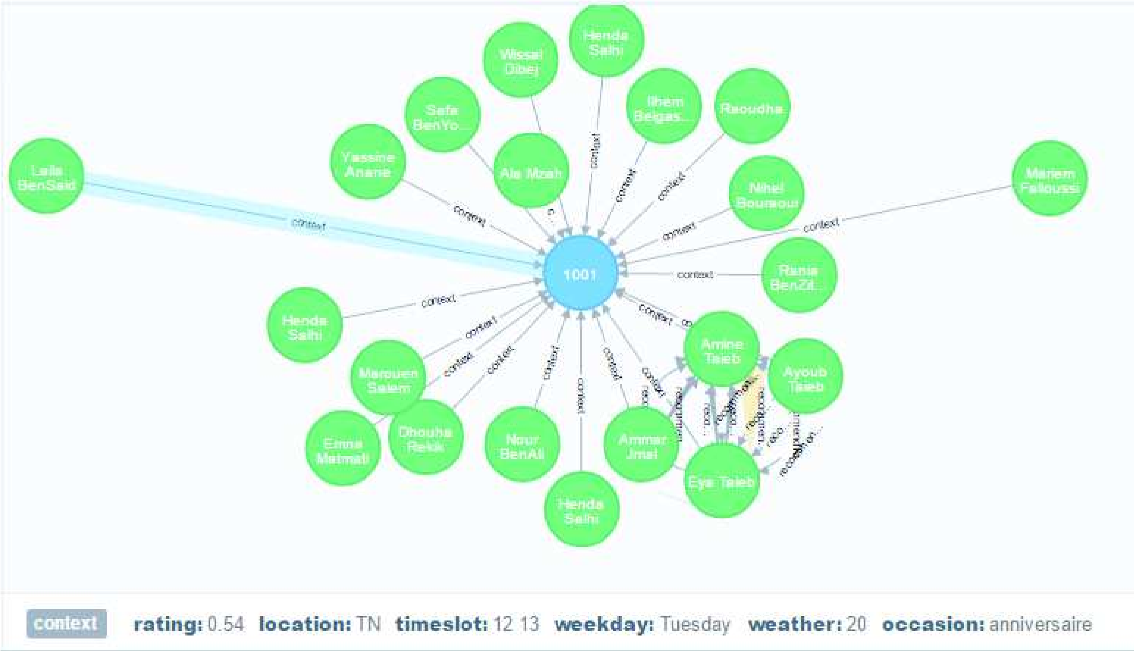

FIGURE 6.4: A captured sample of a sub graph-based data on Pinhole data base

- Columns on those tables become node properties. The columns of the context associated to each viewing history are transformed as the properties of the arc that links the viewer and the video. Each edge that links nodes u and v represents viewing context within which the user u watched the show v. Each viewing context represents a set of properties.
We considered the following context elements : the location, the time slot, the weekday, the weather and the occasion. These properties are captured by the context detector of Pinhole system.

- Remove technical primary keys, keep business primary keys ;
- Replace foreign keys with relationships to the other table, remove them afterwards ;
- Remove data with default values, no need to store those ;
- Data in tables that is denormalized and duplicated might have to be pulled out into separate nodes to get a cleaner model ;
- Indexed column names, might indicate an array property (like category1, category2, etc) ;
- Join tables are transformed into relationships, columns on those tables become relationship properties ;

The edges created between user u and his friend f represent interactions between them on such a video. Each edge that links nodes u and v represents viewing context within which the user u watched the show v.
Each viewing context represents the location, the time slot, the weekday, the weather and the occasion. These properties are captured by social TV system. However, viewer do

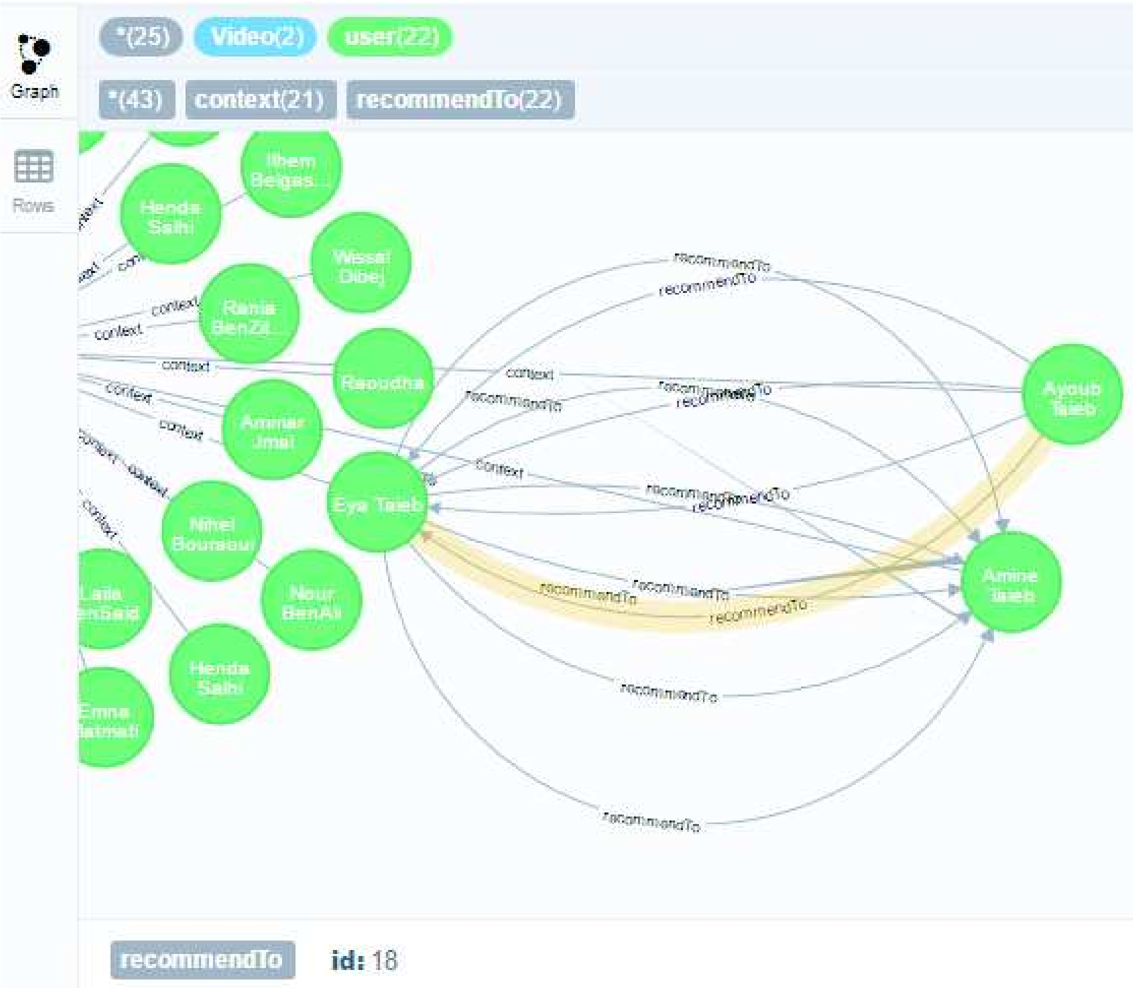

FIGURE 6.5: A subgraph representing relations (or interactions) between users

not rate explicitly TV programs. We estimate his rating r_{uv} according to the time spent watching a TV show.

6.4 User Interfaces on Pinhole Platform

Figure 6.6 represents an example of an recommendation interface on Pinhole platform.

As shown in Figure 6.7, users can watch their favorite TV programs, recommend them to their friends, and interact (e.g. comment) with on social networks (i.e. Facebook and Twitter). The used dataset is described in Section 4.4.1.

Figure 6.8 represents an interface showing how a user can receive notifications offering her a view over the behavior (or actions) of her friends on the platform.

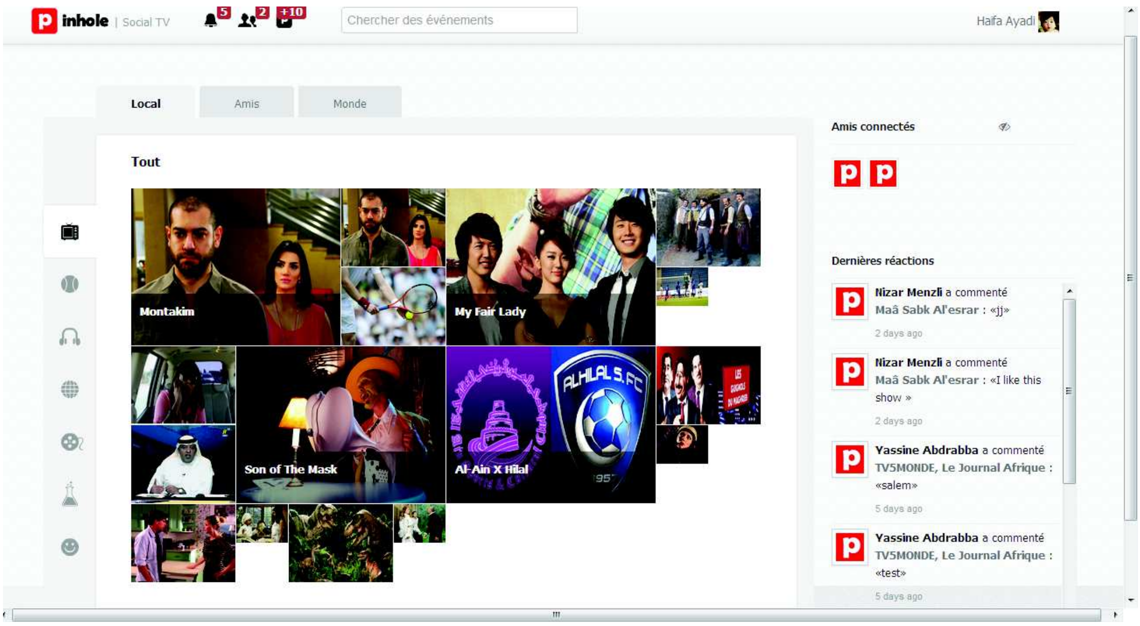

FIGURE 6.6: An example of a recommendation interface on Pinhole social TV platform

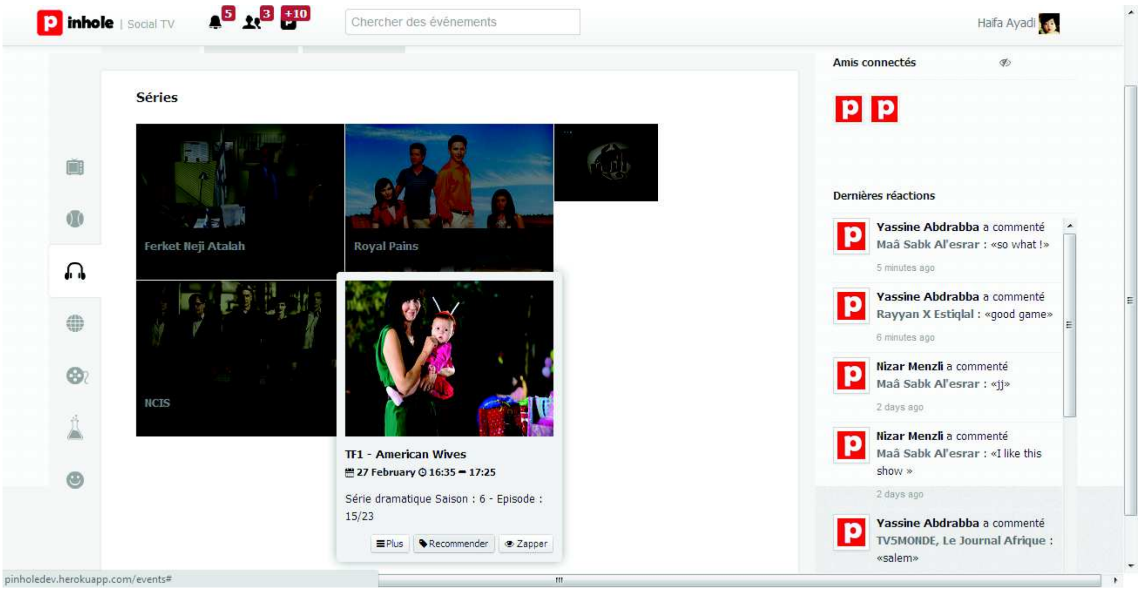

FIGURE 6.7: An interface showing social interactions (e.g. comment and recommend) on TV contents

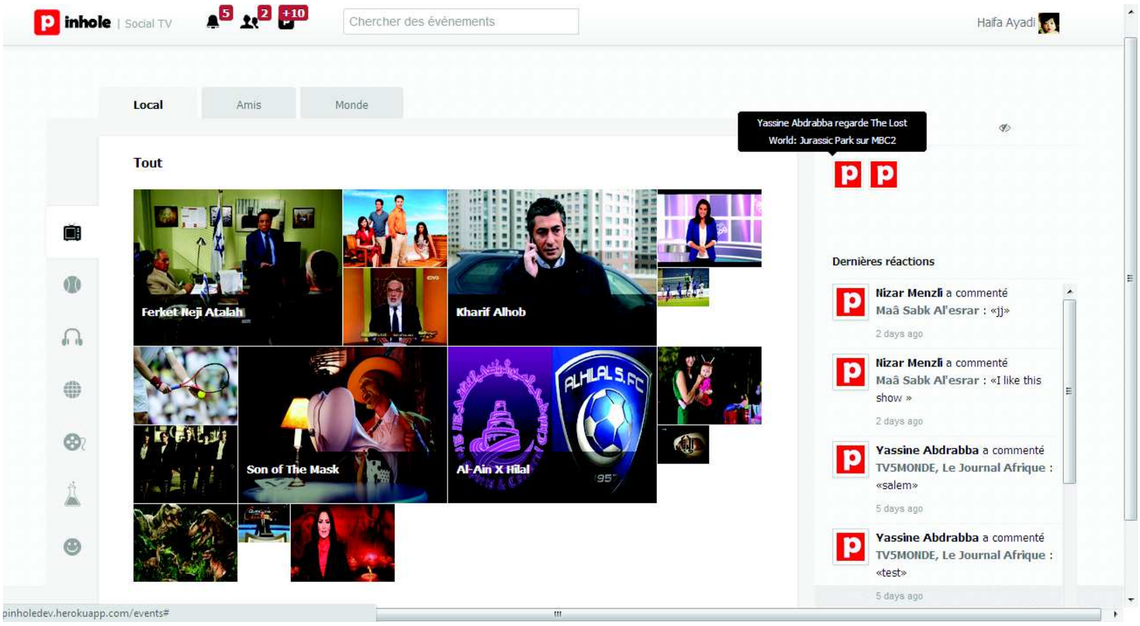

FIGURE 6.8: A user interface showing notifications from Pinhole on friends behavior

6.5 Conclusion

In this Chapter, we present the different tasks made in Pinhole Company in order to prepare development environment to realize the experiments. These tasks are considered essential for experiments realization under favorable conditions. The data base transformation and the architecture definition were primordial to improve time execution and organizing the project stages.

Chapitre 7

Conclusions

This dissertation has made a number of contributions towards the goal of improving items recommendation and predicting users' preferences based on current context of the user and the social influence around her. We define the *current user's context* as the set of circumstances related to the actual environment of the user on which his/her preferences undoubtedly depend. On the other hand, we refer to *social influence* to the measure of the effect of the preferences of the target user's friends on the relevance of an item towards this user.

Key contributions proposed for current contextual information and social influence integration are : the use of a probabilistic context-based model that integrates several current context elements of the user in order to improve items recommendation and adapting prediction to user's environment changes, the proposition of a social-based model to estimate social influence on items relevance, the use of a social trust measure between users and their friends based on their interactions, the proposition of a new approach that jointly integrates the context-based and the social-based models, and the consideration of cold start and the social sparsity problems based on smoothing techniques.

The proposed context-based approach is able to capture and model the current context of the target user. In respect with this captured contexts, a proposed probabilistic model carried out to estimate the relevance of items. It integrates also several context elements such as the weather, the occasion and the weekday for providing more personalized recommendations. The flexibility of our model enables the integration of any additional contextual information in a generic way and independently of their complex and different structures.

The conducted experiments on data collected from Pinhole social TV platform enable us to evaluate the effectiveness of our approach on real data and ensures relevant data access enriched with sufficient information to implement our context-based model. The evaluation of our context-based approach provided encouraging results comparing to time-aware methods in terms of $Precision$, MAE and $RMSE$.

Furthermore, our conducted study on the impact of each context element on the prediction accuracy proved that all the context elements are essential and complementary for the prediction model, and demonstrate the effectiveness of integrating a wide variety of contextual information for improving personalized recommendation.

This dissertation has also explored the use of social filtering techniques to improve items recommendation. The proposed social-based model allows us to model the potential effect of social relationships on user' preferences. The proposed probabilistic social-based model captures quantitatively social interactions between the target user and his/her friends, employs the response of the user for these social interactions, and estimates the social influence on the relevance of the items.

The use of the user-friend similarity measure between the user and his/her friends provided an straightforward and a smooth way to estimate the degree of agreement between them for each interaction.
The study the role of social influence among viewers and their friends in improving prediction proved that using social influence improves recommendation performance comparing to time-aware models. When comparing our model with and without considering social influence, we note that the performance of using similarity measure among viewers and their friends is considerably better.

Jointly integrating the current context and social influence is able to unify the proposed context-based model and the social-based model into one predictive model. The model succeeded to jointly integrate several contextual information and the social influence in order to improve personalized recommendation.
Obviously, the conducted experiments showed that jointly integrating contextual information and social influence is effective at improving recommendation comparing to approaches that treat these two aspects separately.

The proposed approach can avoid the weaknesses of conventional social filtering and collaborative filtering techniques while taking advantage of their strengths. Obviously, testing the ability of our model to solve data sparsity and user cold start recommendation problems demonstrated the effectiveness of using smoothing techniques into the proposed predictive models. The purpose of smoothing techniques is to avoid strong probabilities which are very prominent where missing data occur (e.g., no existing same contexts and no friends have watched the target movie) in the recommendation process. In the sparsity and cold start tests, our model returns consistently accurate predictions at different values of data sparsity.

Overall, these contributions are major advancements in the research of Information Retrieval and Recommender Systems. To the best of our knowledge, this is the first dissertation focusing on jointly exploiting the current contextual information and the social influence between users for improving personalized recommendation. The hope is that, such contributions provide the basis of the development of efficient recommender system predicting and ranking items for the benefit of end users in respect with their contexts and social interactions.

Future Work

There are many directions to proceed in the work presented in this dissertation.
In terms of the context-based model, it can be enriched by integrating other contextual information into the prediction model, such as the user mood. For instance, when users experience a negative emotional state, they tend to watch competition programs to experience excitement and happiness from the program. However, when users are in a positive mood, they tend to choose action programs. It can be estimated also based on real-time user's behavior (i.e., a set of actions or activities like real-time recommendations, tags or comments). This correlation might be an efficient predictor for items recommendation.

Moreover, more databases aside from TV programs accessing history could be collected for building more robust models, integrate more different and accurate contextual information and evaluate our model in various domains' collections. For instance, we can test our model into an event or music recommendation systems where recommendation depend on several users' contexts elements (e.g., Do you listen to music in the same way during exams period or holidays ?).

One possible improvement is to estimate social influence not only between immediate friends but also between non-immediate ones. Because there are a large number of items in some recommender systems, immediate friends of the target user may not have re-viewed the target item.
Therefore, the influences from those friends cannot be used. In order to solve this problem, incorporating the influences from distant friends via extending the social influence among immediate friends. A classification technique could be used in order to identify non-immediate friends whose preferences could influence those of the target user.

It is also possible to employ sentiment analysis techniques in order to estimate the acceptance or the agreement of the user for such a recommendation received by a friend. In this context, the social influence is not estimated based only on the number of interactions between the target user and his/her friends, but also on the degree of the acceptance of the user for these interactions (e.g., the target user comment "I like it " on a movie recommended by his/her friend).

In terms of performance evaluation, we argue for using other evaluation metrics that depending on the goal of the recommender system itself. For instance, we aim to evaluate the effectiveness of our approach in terms of serendipity (i.e, the experience of discovering an unexpected and fortuitous item) for recommender systems in which items are considered as relevant where it is novel and interesting for users, and in terms of diversity for recommender systems where a more diverse recommendation list can lead to higher user satisfaction.

Bibliographie

Adomavicius, G., Sankaranarayanan, R., Sen, S. & Tuzhilin, A. (2005), 'Incorporating contextual information in recommender systems using a multidimensional approach', *ACM Trans. Inf. Syst.* **23**(1), 103–145.

Adomavicius, G. & Tuzhilin, A. (2005), 'Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions', *IEEE Trans. on Knowl. and Data Eng.* **17**(6), 734–749.

Adomavicius, G. & Tuzhilin, A. (2008), Context-aware recommender systems, *in* 'Proceedings of the 2008 ACM Conference on Recommender Systems', ACM, pp. 335–336.

Agrawal, R., Gollapudi, S., Halverson, A. & Ieong, S. (2009), Diversifying search results, *in* 'Proceedings of the Second ACM International Conference on Web Search and Data Mining', ACM, pp. 5–14.

Aizenberg, N., Koren, Y. & Somekh, O. (2012), Build your own music recommender by modeling internet radio streams, *in* 'Proceedings of the 21st International Conference on World Wide Web', ACM, pp. 1–10.

Akrivas, G., Wallace, M., Andreou, G., Stamou, G. & Kollias, S. (2002), Context-sensitive semantic query expansion, *in* 'Artificial Intelligence Systems, 2002. (ICAIS 2002). 2002 IEEE International Conference on', pp. 109–114.

Aleksandrova, M., Brun, A. & Boyer, A. (2014), What about Interpreting Features in Matrix Factorization-based Recommender Systems as Users ?, *in* '25th ACM Conference on Hypertext and Social Media - Workshop on Social Personalisation ', santiago du chili, Chile.

Ali, K. & Van Stam, W. (2004), Tivo : Making show recommendations using a distributed collaborative filtering architecture, *in* 'Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 394–401.

Ansari, A., Essegaier, S. & Kohli, R. (2000), Internet recommendation systems, *in* 'Journal of Marketing Research', pp. 363–375.

Antonelli, F., Francini, G., Geymonat, M. & Lepsøy, S. (2009), Dynamictv : A culture-aware recommender, *in* 'Proceedings of the Third ACM Conference on Recommender Systems', ACM, New York, NY, USA, pp. 257–260.

Ardissono, L., Goy, A., Petrone, G., Segnan, M. & Torasso, P. (2003), Intrigue : Personalized recommendation of tourist attractions for desktop and handset devices, *in* 'Applied Artificial Intelligence', Taylor and Francis, pp. 687–714.

Arias, J. J. P., Vilas, A. F. & Redondo, R. P. D. (2012), Recommender systems for the social web, *in* 'Recommender Systems for the Social Web', Vol. 32, Springer.

Asabere, N. (2012), 'A survey of personalized television and video recommender systems and techniques', *Information and Communication Technology Research* **2**(7), 602–608.

Balabanović, M. & Shoham, Y. (1997), 'Fab : Content-based, collaborative recommendation', *Commun. ACM* **40**(3), 66–72.

Baltrunas, L. & Ricci, F. (2010), Context-dependent recommendations with items splitting, *in* 'IIR 2010 - Proceedings of the First Italian Information Retrieval Workshop, Padua, Italy, January 27-28, 2010', pp. 71–75.

Bambia, M., Boughanem, M. & Faiz, R. (2016), Exploring current viewing context for tv contents recommendation, *in* 'Proceedings of the International Conference on Web Intelligence', IEE/WIC/ACM, pp. 272–279.

Bambia, M., Faiz, R. & Boughanem, M. (2015), Context-awareness and viewer behavior prediction in social-tv recommender systems : Survey and challenges, *in* 'New Trends in Databases and Information Systems', pp. 52–59.

Bar, G. D. & Glinansky, O. (2004), 'FIT-recommending TV programs to family members', *Computers and Graphics* **28**, 149–156.

Barragáns-Martínez, A. B., Arias, J. J. P., Vilas, A. F., Duque, J. G. & Nores, M. L. (2009), 'What's on tv tonight ? an efficient and effective personalized recommender system of tv programs.', *IEEE Trans. Consumer Electronics* **55**(1), 286–294.

Basu, C., Cohen, W. W., Hirsh, H. & Nevill-Manning, C. G. (2011), 'Technical paper recommendation : A study in combining multiple information sources', *CoRR* **abs/1106.0248**.

Bazire, M. & Brézillon, P. (2005*a*), 'Understanding context before using it', pp. 29–40.

Bazire, M. & Brézillon, P. (2005*b*), Understanding context before using it, *in* 'Proceedings of the 5th International Conference on Modeling and Using Context', Springer-Verlag, Berlin, Heidelberg, pp. 29–40.

Belkin, N. J. & Croft, W. B. (1992), 'Information filtering and information retrieval : Two sides of the same coin ?', *Commun. ACM* **35**(12), 29–38.

Bell, R., Koren, Y. & Volinsky, C. (2007), 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', ACM, pp. 95–104.

Bell, R. M. & Koren, Y. (2007), Scalable collaborative filtering with jointly derived neighborhood interpolation weights, *in* 'Proceedings of the 2007 Seventh IEEE International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 43–52.

Bernhaupt, R., Wilfinger, D., Weiss, A. & Tscheligi, M. (2008), 'An ethnographic study on recommendations in the living room : Implications for the design of itv recommender systems', pp. 92–101.

Birnkammerer, S. & Wolfgang Woerndl, G. G. (2009), Recommending for groups in decentralized collaborative filtering, Technical report, Tech. Report TUM-I0927, Institut fuer Informatik, Technische Universitaet Muenchen.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *J. Mach. Learn. Res.* **3**, 993–1022.

Bolger, N., Davis, A. & Rafaeli, E. (2003), 'Diary Methods : Capturing Life as it is Lived', *Annual Review of Psychology* **54**(1), 579–616.

Bonnefoy, D., Bouzid, M., Lhuillier, N. & Mercer, K. (2007), "'more like this" or "not for me" : Delivering personalised recommendations in multi-user environments', pp. 87–96.

Boughareb, D. & Farah, N. (2014), Context in information retrieval, *in* '2014 International Conference on Control, Decision and Information Technologies (CoDIT)', pp. 589–594.

Bradley, K. & Smyth, B. (2001), Improving Recommendation Diversity., *in* 'Proceedings of the 12th National Conference in Artificial Intelligence and Cognitive Science', Maynooth, Ireland, pp. 75–84.

Breese, J. S., Heckerman, D. & Kadie, C. (1998), Empirical analysis of predictive algorithms for collaborative filtering, *in* 'Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence', Morgan Kaufmann Publishers Inc., pp. 43–52.

Brocco, M. & Groh, G. (2009), Team recommendation in open innovation networks, *in* 'Proceedings of the 2009 ACM Conference on Recommender Systems, RecSys 2009, New York, NY, USA, October 23-25, 2009', pp. 365–368.

Brocco, M., Groh, G. & Forster, F. (2010), 'A meta model for team recommendations', pp. 35–50.

Brocco, M., Groh, G. & Kern, C. (2010), On the influence of social factors on team recommendations, *in* 'Workshops Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA', pp. 270–277.

Budzik, J. & Hammond, K. J. (2000), User interactions with everyday applications as context for just-in-time information access, *in* 'Proceedings of the 5th International Conference on Intelligent User Interfaces', ACM, pp. 44–51.

Buriano, L., Marchetti, M., Carmagnola, F., Cena, F., Gena, C. & Torre, I. (2006), The Role of Ontologies in Context-Aware Recommender Systems, *in* 'MDM '06 : Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)', IEEE Computer Society, Washington, DC, USA.

Burke, R. (2002), 'Hybrid recommender systems : Survey and experiments', *User Modeling and User-Adapted Interaction* **12**(4), 331–370.

Canny, J. (2002), Collaborative filtering with privacy via factor analysis, *in* 'Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 238–245.

Cantador, I., Bellogín, A. & Castells, P. (2008), News hand : A semantic web approach to recommending news, *in* 'Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems', Springer-Verlag, pp. 279–283.

Cantador, I., Bellogín, A. & Vallet, D. (2010), Content-based recommendation in social tagging systems, *in* 'Proceedings of the Fourth ACM Conference on Recommender Systems', ACM, pp. 237–240.

Card, S. K., Robertson, G. G. & Mackinlay, J. D. (1991), The information visualizer, an information workspace, *in* 'Proceedings of the SIGCHI Conference on Human Factors in Computing Systems', ACM, pp. 181–186.

Celma, O. & Herrera, P. (2008), A new approach to evaluating novel recommendations, *in* 'Proceedings of the 2008 ACM Conference on Recommender Systems', ACM, pp. 179–186.

Chai, T. & Draxler, R. R. (2014), 'Root mean square error (rmse) or mean absolute error (mae) ? arguments against avoiding rmse in the literature', *Geoscientific Model Development* **7**(3), 1247–1250.

Champin, P.-A., Briggs, P., Coyle, M. & Smyth, B. (2010), 'Coping with noisy search experiences', pp. 5–18.

Chandar, P. & Carterette, B. (2010), Diversification of search results using webgraphs, *in* 'Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 869–870.

Chandra, B. & Gupta, M. (2011), 'Robust approach for estimating probabilities in naïve-bayes classifier for gene expression data', *Expert Syst. Appl.* **38**(3), 1293–1298.

Chaney, A. J., Blei, D. M. & Eliassi-Rad, T. (2015), A probabilistic model for using social networks in personalized item recommendation, *in* 'Proceedings of the 9th ACM Conference on Recommender Systems', ACM, pp. 43–50.

Chang, N., Irvan, M. & Terano, T. (2013), A tv program recommender framework, *in* 'KES'13', pp. 561–570.

Chen, G. & Kotz, D. (2000), A survey of context-aware mobile computing research, Technical report, Dartmouth College, Hanover, NH, USA.

Chen, S. F. & Goodman, J. (1996), An empirical study of smoothing techniques for language modeling, *in* 'Proceedings of the 34th Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, pp. 310–318.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Büttcher, S. & MacKinnon, I. (2008), Novelty and diversity in information retrieval evaluation, *in* 'Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 659–666.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. & Sartin, M. (1999), 'Combining content-based and collaborative filters in an online newspaper'.

Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J. & Suri, S. (2008), Feedback effects between similarity and social influence in online communities, *in* 'Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 160–168.

Crossen, A., Budzik, J. & Hammond, K. J. (2002), Flytrap : Intelligent group music recommendation, *in* 'Proceedings of the 7th International Conference on Intelligent User Interfaces', ACM, pp. 184–185.

Das, A. S., Datar, M., Garg, A. & Rajaram, S. (2007), Google news personalization : Scalable online collaborative filtering, *in* 'Proceedings of the 16th International Conference on World Wide Web', ACM, pp. 271–280.

De Gemmis, M., Lops, P., Semeraro, G. & Basile, P. (2008), Integrating tags in a semantic content-based recommender, *in* 'Proceedings of the 2008 ACM Conference on Recommender Systems', ACM, pp. 163–170.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science* **41**(6), 391–407.

Dey, A. K., Abowd, G. D. & Salber, D. (2001), 'A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications', *Hum.-Comput. Interact.* **16**(2), 97–166.

Domingues, M. A., Jorge, A. M. & Soares, C. (2011), 'Using contextual information as virtual items on top-n recommender systems', *CoRR* **abs/1111.2948**.

Dourish, P. (2004), 'What we talk about when we talk about context', *Personal Ubiquitous Comput.* **8**(1), 19–30.

Ebrahimi, F. & Golpayegani, S. A. H. (2016), Personalized recommender system based on social relations, *in* '2016 24th Iranian Conference on Electrical Engineering (ICEE)', pp. 218–223.

Fasli, M. (2006), On the relationship between roles and power : Preliminary report, *in* 'Proceedings of the 2006 ACM Symposium on Applied Computing', ACM, pp. 313–318.

Felfernig, A. (2005), Koba4ms : Selling complex products and services using knowledge-based recommender technologies, *in* 'CEC '05 : Proceedings of the Seventh IEEE International Conference on E-Commerce Technology', IEEE Computer Society, Washington, DC, USA, pp. 92–100.

Feng, W. & Wang, J. (2012), Incorporating heterogeneous information for personalized tag recommendation in social tagging systems, *in* 'Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 1276–1284.

Fischer, G. (2001), 'User modeling in human computer interaction', *User Modeling and User-Adapted Interaction* **11**(1-2), 65–86.

Fleder, D. M. & Hosanagar, K. (2007), Recommender systems and their impact on sales diversity, *in* 'Proceedings of the 8th ACM Conference on Electronic Commerce', ACM, pp. 192–199.

Forsati, R., Barjasteh, I., Masrour, F., Esfahanian, A.-H. & Radha, H. (2015), PushTrust : An Efficient Recommendation Algorithm by Leveraging Trust and Distrust Relations, *in* 'Proc. of RecSys' 15', ACM, pp. 51–58.

Gantner, Z., Rendle, S. & Schmidt-Thieme, L. (2010), Factorization models for context-/time-aware movie recommendations, *in* 'Proceedings of the Workshop on Context-Aware Movie Recommendation', ACM, pp. 14–19.

Ge, M., Delgado-battenfeld, C. & Jannach, D. (2010), Beyond accuracy : evaluating recommender systems by coverage and serendipity, *in* 'In RecSys 10', p. 257.

George, T. & Merugu, S. (2005), A scalable collaborative filtering framework based on co-clustering, *in* 'Proceedings of the Fifth IEEE International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 625–628.

Golbeck, J. (2006), Generating predictive movie recommendations from trust in social networks, *in* 'Proceedings of the 4th International Conference on Trust Management', Springer-Verlag, pp. 93–104.

Groh, G. (2007), Groups and group-instantiations in mobile communities detection, modeling and applications, in, *in* 'In Proceedings of the International Conference on Weblogs and Social Media'.

Groh, G., Birnkammerer, S. & Köllhofer, V. (2012), Social recommender systems, *in* 'Recommender Systems for the Social Web', Springer Berlin Heidelberg, pp. 3–42.

Groh, G. & Daubmeier, P. (2010), State of the art in mobile social networking on the web, Technical report, TUM - Boltzmanns University.

Groh, G. & Ehmig, C. (2007), Recommendations in taste related domains : Collaborative filtering vs. social filtering, *in* 'Proceedings of the 2007 International ACM Conference on Supporting Group Work', ACM, pp. 127–136.

Guo, D., Xu, J., Zhang, J., Xu, M., Cui, Y. & He, X. (2017), 'User relationship strength modeling for friend recommendation on instagram', *Neurocomput.* **239**(C), 9–18.

Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S. & Ofek-Koifman, S. (2009), Personalized recommendation of social software items based on social relations, *in* 'Proceedings of the Third ACM Conference on Recommender Systems', RecSys '09, ACM, pp. 53–60.

Hanani, U., Shapira, B. & Shoval, P. (2001), 'Information filtering : Overview of issues, research and systems', *User Modeling and User-Adapted Interaction* **11**(3), 203–259.

Hariri, N., Mobasher, B. & Burke, R. (2012*a*), Context-aware music recommendation based on latenttopic sequential patterns, *in* 'Proceedings of the Sixth ACM Conference on Recommender Systems', ACM, pp. 131–138.

Hariri, N., Mobasher, B. & Burke, R. (2012*b*), Context-aware music recommendation based on latenttopic sequential patterns, *in* 'Proceedings of the Sixth ACM Conference on Recommender Systems', RecSys '12, ACM, New York, NY, USA, pp. 131–138.

Hariri, N., Mobasher, B. & Burke, R. (2013), Query-driven context aware recommendation, *in* 'Proceedings of the 7th ACM Conference on Recommender Systems', ACM, pp. 9–16.

Hariri, N., Mobasher, B. & Burke, R. D. (2014), Context adaptation in interactive recommender systems, *in* 'Proc. of RecSys' 14', ACM, pp. 41–48.

Harman, D. (1994), Overview of the third text retrieval conference (TREC-3), *in* 'Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994', pp. 1–20.

Hauver, D. & French, J. (23-24 Nov. 2001), 'Flycasting : using collaborative filtering to generate a playlist for online radio', *Web Delivering of Music, 2001. Proceedings. First International Conference on* pp. 123–130.

Hazimeh, H. & Zhai, C. (2015), Axiomatic analysis of smoothing methods in language models for pseudo-relevance feedback, *in* 'Proceedings of the 2015 International Conference on The Theory of Information Retrieval', ACM, pp. 141–150.

Herlocker, J. L., Konstan, J. A., Borchers, A. & Riedl, J. (1999), An algorithmic framework for performing collaborative filtering, *in* 'Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 230–237.

Herlocker, J. L., Konstan, J. A. & Riedl, J. (2000), Explaining collaborative filtering recommendations, *in* 'Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work', ACM, pp. 241–250.

Herlocker, J. L., Konstan, J. A., Terveen, L. G. & Riedl, J. T. (2004), 'Evaluating collaborative filtering recommender systems', *ACM Trans. Inf. Syst.* **22**(1), 5–53.

Huang, J., Cheng, X., Guo, J., Shen, H. & Yang, K. (2010), Social recommendation with interpersonal influence., *in* 'ECAI', Vol. 215 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 601–606.

Isinkaye, F., Folajimi, Y. & Ojokoh, B. (2015), 'Recommendation systems : Principles, methods and evaluation', *Egyptian Informatics Journal* **16**(3), 261 – 273.

Iwata, T., Saito, K. & Yamada, T. (2007), Modeling user behavior in recommender systems based on maximum entropy, *in* 'WWW '07 : Proceedings of the 16th international conference on World Wide Web', ACM Press, New York, NY, USA, pp. 1281–1282.

Jamali, M. & Ester, M. (2009), Trustwalker : A random walk model for combining trust-based and item-based recommendation, *in* 'Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 397–406.

Jambor, T. & Wang, J. (2010), Optimizing multiple objectives in collaborative filtering, *in* 'Proceedings of the Fourth ACM Conference on Recommender Systems', ACM, pp. 55–62.

Jameson, A. (2004*a*), More than the sum of its members : Challenges for group recommender systems, *in* 'Proceedings of the Working Conference on Advanced Visual Interfaces', ACM, pp. 48–54.

Jameson, A. (2004*b*), More than the sum of its members : Challenges for group recommender systems, *in* 'Proceedings of the Working Conference on Advanced Visual Interfaces', ACM, pp. 48–54.

Jameson, A., Baldes, S. & Kleinbauer, T. (2004), Two methods for enhancing mutual awareness in a group recommender system, *in* 'Proceedings of the Working Conference on Advanced Visual Interfaces', ACM, pp. 447–449.

Jameson, A. & Smyth, B. (2007), The adaptive web, Springer-Verlag, Berlin, Heidelberg, chapter Recommendation to Groups, pp. 596–627.

Jelinek, F. & Mercer, R. L. (1980), Interpolated estimation of Markov source parameters from sparse data, *in* 'Proceedings, Workshop on Pattern Recognition in Practice', North Holland, Amsterdam, pp. 381–397.

Ji, K., Sun, R., Shu, W. & Li, X. (2015), 'Next-song recommendation with temporal dynamics', *Know.-Based Syst.* **88**(C), 134–143.

Jin, R., Si, L. & Zhai, C. (2006), 'A study of mixture models for collaborative filtering', *Information Retrieval* **9**(3), 357–382.

Jones, G. J. F. & Brown, P. J. (2002), Challenges and opportunities for context-aware retrieval on mobile devices, *in* 'In : Proceedings of the SIGIR workshop on Mobile Personal Information Retrieval', ACM, pp. 47–56.

Jones, K. S. (1972), 'A statistical interpretation of term specificity and its application in retrieval', *Journal of Documentation* **28**, 11–21.

Kathuria, A., Jansen, B. J., Hafernik, C. T. & Spink, A. (2010), 'Classifying the user intent of web queries using k-means clustering.', *Internet Research* **20**(5), 563–581.

Keim, T., Weitzel, T. & Färber, F. (2003), An automated recommendation approach to selection in personnel recruitment, *in* 'Americas Conference on Information Systems 2003'.

Kim, B. M., Li, Q., Park, C. S., Kim, S. G. & Kim, J. Y. (2006), 'A new approach for combining content-based and collaborative filters', *Journal of Intelligent Information Systems* **27**(1), 79–91.

Kim, D. & Yum, B.-J. (2005), 'Collaborative filtering based on iterative principal component analysis', *Expert Syst. Appl.* **28**(4), 823–830.

Kim, S. & Kwon, J. (2007), 'Effective context-aware recommendation on the semantic web', *International Journal of Computer Science and Network Security* pp. 154–159.

Kohavi, R., Longbotham, R., Sommerfield, D. & Henne, R. M. (2009), 'Controlled experiments on the web : survey and practical guide', *Data Mining and Knowledge Discovery* **18**(1), 140–181.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R. & Riedl, J. (1997), 'GroupLens : applying collaborative filtering to Usenet news', *Communications of the ACM* **40**(3), 77–87.

Konstas, I., Stathopoulos, V. & Jose, J. M. (2009), On social networks and collaborative recommendation, *in* 'Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 195–202.

Koren, Y. (2008), Factorization meets the neighborhood : A multifaceted collaborative filtering model, *in* 'Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, New York, NY, USA, pp. 426–434.

Krishnan, S., Patel, J., Franklin, M. J. & Goldberg, K. (2014), A methodology for learning, analyzing, and mitigating social influence bias in recommender systems, *in* 'Proceedings of the 8th ACM Conference on Recommender Systems', ACM, pp. 137–144.

Kumar, R., Novak, J. & Tomkins, A. (2006), Structure and evolution of online social networks, *in* 'Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 611–617.

Lang, K. (1995), Newsweeder : Learning to filter netnews, *in* 'in Proceedings of the 12th International Machine Learning Conference (ML95'.

Lathia, N., Hailes, S. & Capra, L. (2008), Trust-based collaborative filtering, *in* 'Trust Management II', Vol. 263 of *The International Federation for Information Processing*, Springer US, pp. 119–134.

Lathia, N., Hailes, S., Capra, L. & Amatriain, X. (2010), Temporal diversity in recommender systems, *in* 'Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 210–217.

Lawrence, S. (2000), 'Context in web search', *IEEE Data Engineering Bulletin* **23**, 25–32.

Lazar, C. & Doncescu, A. (2009), Non negative matrix factorization clustering capabilities ; application on multivariate image segmentation., *in* 'CISIS', IEEE Computer Society, pp. 924–929.

Lieberman, H. (1995), Letizia : An agent that assists web browsing, *in* 'Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1', Morgan Kaufmann Publishers Inc., pp. 924–929.

Linden, G., Smith, B. & York, J. (2003), 'Amazon.com recommendations : Item-to-item collaborative filtering', *IEEE Internet Computing* **7**(1), 76–80.

Liu, F. & Lee, H. J. (2010), 'Use of social network information to enhance collaborative filtering performance', *Expert Systems with Applications* **37**(7), 4772 – 4778.

Liu, N. N., Cao, B., Zhao, M. & Yang, Q. (2010), Adapting neighborhood and matrix factorization models for context aware recommendation, *in* 'Proceedings of the Workshop on Context-Aware Movie Recommendation', ACM, pp. 7–13.

Liu, N. N., Zhao, M., Xiang, E. & Yang, Q. (2010), Online evolutionary collaborative filtering, *in* 'Proceedings of the Fourth ACM Conference on Recommender Systems', RecSys '10, ACM, New York, NY, USA, pp. 95–102.

Liu, P., Ma, J., Wang, Y., Ma, L. & Huang, S. (2016), *A Context-Aware Method for Top-k Recommendation in Smart TV*, Springer International Publishing, Cham, pp. 150–161.

Liu, X. & Aberer, K. (2013), Soco : A social network aided context-aware recommender system, *in* 'Proceedings of the 22Nd International Conference on World Wide Web', ACM, pp. 781–802.

Lops, P., de Gemmis, M. & Semeraro, G. (2011), Content-based recommender systems : State of the art and trends, *in* 'Recommender Systems Handbook', Springer US, Boston, MA, pp. 73–105.

Ma, H. (2013), An experimental study on implicit social recommendation, *in* 'Proc. of SIGIR' 13', ACM, pp. 73–82.

Ma, H. (2014), On measuring social friend interest similarities in recommender systems, *in* 'Proc. of SIGIR '14', pp. 465–474.

Ma, H., King, I. & Lyu, M. R. (2009), Learning to recommend with social trust ensemble, *in* 'Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 203–210.

Ma, H., Lyu, M. R. & King, I. (2009), Learning to recommend with trust and distrust relationships, *in* 'Proceedings of the Third ACM Conference on Recommender Systems', ACM, pp. 189–196.

Ma, H., Zhou, D., Liu, C., Lyu, M. R. & King, I. (2011), Recommender systems with social regularization, *in* 'Proceedings of the Fourth ACM International Conference on Web Search and Data Mining', ACM, pp. 287–296.

Maamar, Z., Benslimane, D. & Narendra, N. C. (2006), 'What can context do for web services ?', *Commun. ACM* **49**(12), 98–103.

Macedo, A. Q., Marinho, L. B. & Santos, R. L. (2015), Context-aware event recommendation in event-based social networks, *in* 'Proceedings of the 9th ACM Conference on Recommender Systems', ACM, pp. 123–130.

Magara, M. B., Ojo, S., Ngwira, S. & Zuva, T. (2016), Mplist : Context aware music playlist, *in* '2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)', pp. 309–316.

Malinowski, J., Keim, T., Weitzel, T. & Wendt, O. (2005), Decision support for team building : Incorporating trust into a recommender-based approach, *in* 'Pacific Asia Conference on Information Systems, PACIS 2005, Bangkok, Thailand, July 7-10, 2005', p. 49.

Manning, C. D., Raghavan, P. & Schütze, H. (2008), *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA.

Massa, P. & Avesani, P. (2004), Trust-aware collaborative filtering for recommender systems, *in* 'In Proc. of Federated Int. Conference On The Move to Meaningful Internet : CoopIS, DOA, ODBASE', pp. 492–508.

Massa, P. & Avesani, P. (2007), Trust-aware recommender systems, *in* 'Proceedings of the 2007 ACM Conference on Recommender Systems', ACM, pp. 17–24.

Masthoff, J. (2004), 'Group modeling : Selecting a sequence of television items to suit a group of viewers', *User Modeling and User-Adapted Interaction* **14**(1), 37–85.

McKechnie, J. (1983), 'Webster's new twentieth century dictionary of the english language, unabridged : Based upon the broad foundations laid down by noah webster'.

McNee, S. M., Riedl, J. & Konstan, J. A. (2006), Making recommendations better : An analytic model for human-recommender interaction, *in* 'CHI '06 Extended Abstracts on Human Factors in Computing Systems', ACM, pp. 1103–1108.

Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P. & Bhattacharjee, B. (2007), Measurement and analysis of online social networks, *in* 'Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement', ACM, pp. 29–42.

Missaoui, S. & Faiz, R. (2014), *A Preferences Based Approach for Better Comprehension of User Information Needs*, Springer International Publishing, Cham, pp. 94–103.

Mooney, R. J. & Roy, L. (2000), Content-based book recommending using learning for text categorization, *in* 'Proceedings of the Fifth ACM Conference on Digital Libraries', ACM, pp. 195–204.

Narducci, F., de Gemmis, M. & Lops, P. (2015), A general architecture for an emotion-aware content-based recommender system, *in* 'Proceedings of the 3rd Workshop on Emotions and Personality in Personalized Systems 2015', EMPIRE '15, ACM, New York, NY, USA, pp. 3–6.

Noguera, J., Barranco, M. J., Segura, R. & Martínez, L. (2012), 'A mobile 3d-gis hybrid recommender system for tourism', *Information Sciences* **215**, 37–52.

O'Connor, M., Cosley, D., Konstan, J. A. & Riedl, J. (2001), Polylens : A recommender system for groups of users, *in* 'Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work', Kluwer Academic Publishers, pp. 199–218.

O'Donovan, J. & Smyth, B. (2005), Trust in recommender systems, *in* 'Proceedings of the 10th International Conference on Intelligent User Interfaces', ACM, pp. 167–174.

Oh, J., Sung, Y., Kim, J., Humayoun, M., Park, Y.-H. & Yu, H. (2012), Time-dependent user profiling for tv recommendation, *in* 'Cloud and Green Computing', pp. 783–787.

Oku, K., Nakajima, S., Miyazaki, J. & Uemura, S. (2006), Context-aware svm for context-dependent information recommendation, *in* 'Proceedings of the 7th International Conference on Mobile Data Management', IEEE Computer Society, Washington, DC, USA, pp. 109–.

Onuma, K., Tong, H. & Faloutsos, C. (2009), TANGENT : a novel, 'Surprise me', recommendation algorithm, *in* 'KDD '09 : Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, New York, NY, USA, pp. 657–666.

Ostuni, V. C., Di Noia, T., Di Sciascio, E. & Mirizzi, R. (2013), Top-n recommendations from implicit feedback leveraging linked open data, *in* 'Proceedings of the 7th ACM Conference on Recommender Systems', ACM, pp. 85–92.

Pálovics, R., Benczúr, A. A., Kocsis, L., Kiss, T. & Frigó, E. (2014), Exploiting temporal influence in online recommendation, *in* 'Proceedings of the 8th ACM Conference on Recommender Systems', ACM, pp. 273–280.

Panayiotou, C., Andreou, M., Samaras, G. & Pitsillides, A. (2005), Time based personalization for the moving user, *in* 'International Conference on Mobile Business (ICMB'05)', pp. 128–136.

Papadogiorgaki, M., Papastathis, V., Nidelkou, E., Kompatsiaris, Y., Waddington, S., Bratu, B. & Ribiere, M. (2007), 'Distributed user modeling for personalized news delivery in mobile devices', *Semantic Media Adaptation and Personalization, International Workshop on* pp. 80–85.

Paterek, A. (2007), 'Improving regularized Singular Value Decomposition for collaborative filtering', *Proceedings of KDD Cup and Workshop* .

Pazzani, M. & Billsus, D. (1997), 'Learning and revising user profiles : The identification of interesting web sites', *Machine Learning* **27**(3), 313–331.

Pazzani, M. J. & Billsus, D. (2007), 'The adaptive web', pp. 325–341.

Pennock, D. M., Horvitz, E. & Giles, C. L. (2000), Social choice theory and recommender systems : Analysis of the axiomatic foundations of collaborative filtering, *in* 'Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.', pp. 729–734.

Popescul, A., Ungar, L. H., Pennock, D. M. & Lawrence, S. (2001), Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments, *in* 'Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 437–444.

Porteous, I., Asuncion, A. U. & Welling, M. (2010), Bayesian matrix factorization with side information and dirichlet process mixtures., *in* 'AAAI', AAAI Press.

Prahalad, C. (2004), Beyond crm : Ck prahalad predicts customer context is the next big thing, *in* 'American Management Association McWorld'.

Pyo, S., Kim, E. & Kim, M. (2013), 'Automatic and personalized recommendation of tv program contents using sequential pattern mining for smart tv user interaction.', *Multimedia Syst.* **19**(6), 527–542.

Radlinski, F., Bennett, P. N., Carterette, B. & Joachims, T. (2009), 'Redundancy, diversity and interdependent document relevance', *SIGIR Forum* **43**(2), 46–52.

Radlinski, F., Kleinberg, R. & Joachims, T. (2008), Learning diverse rankings with multi-armed bandits, *in* 'Proceedings of the 25th International Conference on Machine Learning', ACM, pp. 784–791.

Rafiei, D., Bharat, K. & Shukla, A. (2010), Diversifying web search results, *in* 'Proceedings of the 19th International Conference on World Wide Web', ACM, pp. 781–790.

Resnick, P. & Varian, H. R. (1997), 'Recommender systems', *Commun. ACM* **40**(3), 56–58.

Ricci, F., Rokach, L. & Shapira, B. (2011), Introduction to recommender systems handbook, *in* 'Recommender Systems Handbook', Springer US, Boston, MA, pp. 1–35.

Ricci, F., Rokach, L. & Shapira, B. (2015), 'Recommender systems handbook'.

Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (2010), 'Recommender systems handbook'.

Richardson, M., Agrawal, R. & Domingos, P. (2006), Trust management for the semantic web, *in* 'The Semantic Web — ISWC 2003', Springer-Verlag, pp. 351–368.

Robertson, S. E. & Sparck Jones, K. (1988), 'Document retrieval systems', pp. 143–160.

Robertson, S. & Zaragoza, H. (2009), 'The probabilistic relevance framework : Bm25 and beyond', *Found. Trends Inf. Retr.* **3**(4), 333–389.

Roth, M., Ben-David, A., Deutscher, D., Flysher, G., Horn, I., Leichtberg, A., Leiser, N., Matias, Y. & Merom, R. (2010), Suggesting friends using the implicit social graph, *in* 'Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining'.

Salakhutdinov, R., Mnih, A. & Hinton, G. (2007), Restricted boltzmann machines for collaborative filtering, *in* 'Proceedings of the 24th International Conference on Machine Learning', ACM, pp. 791–798.

Salter, J. & Antonopoulos, N. (2006), 'Cinemascreen recommender agent : Combining collaborative and content-based filtering', *IEEE Intelligent Systems* **21**(1), 35–41.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2000*a*), Analysis of recommendation algorithms for e-commerce, *in* 'Proceedings of the 2Nd ACM Conference on Electronic Commerce', ACM, pp. 158–167.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2001), Item-based collaborative filtering recommendation algorithms, *in* 'Proceedings of the 10th International Conference on World Wide Web', ACM, pp. 285–295.

Sarwar, B. M., Karypis, G., Konstan, J. A. & Riedl, J. T. (2000*b*), Application of dimensionality reduction in recommender system – a case study, *in* 'IN ACM WEBKDD WORKSHOP'.

Schein, A. I., Popescul, A., Ungar, L. H. & Pennock, D. M. (2002), Methods and metrics for cold-start recommendations, *in* 'Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 253–260.

Schilit, B., Adams, N. & Want, R. (1994), Context-aware computing applications, *in* 'Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications', IEEE Computer Society, Washington, DC, USA, pp. 85–90.

Schilit, B. N. & Theimer, M. M. (1994), 'Disseminating active map information to mobile hosts', *Netwrk. Mag. of Global Internetwkg.* **8**(5), 22–32.

Semeraro, G., Degemmis, M., Lops, P. & Basile, P. (2007), Combining learning and word sense disambiguation for intelligent user profiling, *in* 'Proceedings of the 20th International Joint Conference on Artifical Intelligence', Morgan Kaufmann Publishers Inc., pp. 2856–2861.

Seo, Y.-D., Kim, Y.-G., Lee, E. & Baik, D.-K. (2017), 'Personalized recommender system based on friendship strength in social network services', *Expert Systems with Applications* **69**, 135 – 148.

Shani, G. & Gunawardana, A. (2011), 'Evaluating recommendation systems', *Recommender Systems Handbook* pp. 257–297.

Sheth, B. & Maes, P. (1993), Evolving agents for personalized information filtering, *in* 'Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on', pp. 345–352.

Sieg, A., Mobasher, B. & Burke, R. (2007), 'Representing context in web search with ontological user profiles', pp. 439–452.

Sinha, R. R. & Swearingen, K. (2001), Comparing recommendations made by online systems and friends., *in* 'DELOS Workshop : Personalisation and Recommender Systems in Digital Libraries'.

Smucker, M. D. & Allan, J. (2005), An investigation of dirichlet prior smoothing's performance advantage, Technical report.

Smyth, B., Balfe, E., Boydell, O., Bradley, K., Briggs, P., Coyle, M. & Freyne, J. (2005), A Live-User Evaluation of Collaborative Web Search., *in* 'Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)', Edinburgh, Scotland, pp. 1419–1424.

Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M. & Boydell, O. (2004), 'Exploiting query repetition and regularity in an adaptive community-based web search engine', *User Modeling and User-Adapted Interaction* **14**(5), 383–423.

Steurer, M. & Trattner, C. (2013), Predicting interactions in online social networks : An experiment in second life, *in* 'Proceedings of the 4th International Workshop on Modeling Social Media', ACM, pp. 5 :1–5 :8.

Sun, L., Wang, X., Wang, Z., Zhao, H. & Zhu, W. (2017), 'Social-aware video recommendation for online social groups', *IEEE Transactions on Multimedia* **19**(3), 609–618.

Sun, Z., Han, L., Huang, W., Wang, X., Zeng, X., Wang, M. & Yan, H. (2015), 'Recommender systems based on social networks', *J. Syst. Softw.* **99**(C), 109–119.

Takács, G., Pilászy, I., Németh, B. & Tikk, D. (2007), 'Major components of the gravity recommendation system', *SIGKDD Explor. Newsl.* **9**(2), 80–83.

Terry, D. B. (1993), A tour through tapestry, *in* 'Proceedings of the Conference on Organizational Computing Systems', ACM, pp. 21–30.

Tong, H., Faloutsos, C. & Pan, J.-Y. (2006), Fast random walk with restart and its applications, *in* 'Proceedings of the Sixth International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 613–622.

Turrin, R., Pagano, R., Cremonesi, P. & Condorelli, A. (2014), Time-based TV programs prediction, *in* '1st Workshop on Recommender Systems for Television and Online Video at ACM RecSys' 14'.

Umyarov, A. & Tuzhilin, A. (2011), 'Using external aggregate ratings for improving individual recommendations', *ACM Trans. Web* **5**(1), 3 :1–3 :40.

Valcarce, D., Parapar, J. & Barreiro, Á. (2016), Additive smoothing for relevance-based language modelling of recommender systems, *in* 'Proceedings of the 4th Spanish Conference on Information Retrieval, Granada, Spain, June 14-16, 2016', p. 9.

Vargas, S. & Castells, P. (2011), Rank and relevance in novelty and diversity metrics for recommender systems, *in* 'Proceedings of the Fifth ACM Conference on Recommender Systems', ACM, pp. 109–116.

Verbert, K., Manouselis, N., Ochoa, X., Wolpers, M., Drachsler, H., Bosnic, I. & Duval, E. (2012), 'Context-aware recommender systems for learning : A survey and future challenges', *IEEE Trans. Learn. Technol.* **5**(4), 318–335.

Wang, J., de Vries, A. P. & Reinders, M. J. T. (2008), 'Unified relevance models for rating prediction in collaborative filtering', *ACM Trans. Inf. Syst.* **26**(3), 16 :1–16 :42.

Wang, J., Robertson, S., de Vries, A. P. & Reinders, M. J. T. (2008), 'Probabilistic relevance ranking for collaborative filtering', *Information Retrieval* **11**(6), 477–497.

Weng, L. T., Xu, Y., Li, Y. & Nayak, R. (2007), Improving Recommendation Novelty Based on Topic Taxonomy, *in* 'Web Intelligence and Intelligent Agent Technology Workshops, 2007 IEEE/WIC/ACM International Conferences on', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, pp. 115–118.

Wörndl, W., Mühe, H. & Prinz, V. (2009), Decentral item-based collaborative filtering for recommending images on mobile devices, *in* 'MDM 2009, Tenth International Conference on Mobile Data Management, Taipei, Taiwan, 18-20 May 2009', pp. 608–613.

Yang, X., Guo, Y., Liu, Y. & Steck, H. (2014), 'A survey of collaborative filtering based social recommender systems', *Computer Communications* **41**, 1–10.

Yang, X., Steck, H. & Liu, Y. (2012), Circle-based recommendation in online social networks, *in* 'Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '12, ACM, New York, NY, USA, pp. 1267–1275.

Ye, M., Liu, X. & Lee, W.-C. (2012), Exploring social influence for recommendation : A generative model approach, *in* 'Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 671–680.

Yu, Z., Zhou, X., Zhang, D., Chin, C.-Y., Wang, X. & Men, J. (2006), 'Supporting context-aware media recommendations for smart phones', *IEEE Pervasive Computing* **5**(3), 68–75.

Yuan, Q., Cong, G., Ma, Z., Sun, A. & Thalmann, N. M. (2013), Time-aware point-of-interest recommendation, *in* 'Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '13, ACM, New York, NY, USA, pp. 363–372.

Zhai, C. & Lafferty, J. (2001), A study of smoothing methods for language models applied to ad hoc information retrieval, *in* 'Proceedings of the 24th Annual International

ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 334–342.

Zhang, M. & Hurley, N. (2009), Statistical modeling of diversity in top-n recommender systems, *in* 'Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01', IEEE Computer Society, Washington, DC, USA, pp. 490–497.

Zhang, Y., Callan, J. & Minka, T. (2002), Novelty and redundancy detection in adaptive filtering, *in* 'Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 81–88.

Zhao, T., Li, C., Li, M., Ding, Q. & Li, L. (2013), Social recommendation incorporating topic mining and social trust analysis, *in* 'Proceedings of the 22nd ACM International Conference on Information Knowledge Management', ACM, pp. 1643–1648.

Zheleva, E., Guiver, J., Rodrigues, E. M. & Milic-Frayling, N. (2010), Statistical models of music-listening sessions in social media, *in* 'The 19th International World Wide Web Conference (WWW2010), April 26-30, 2010, Raleigh NC, USA', Association for Computing Machinery, Inc.

Zhou, T., Kuscsik, Z., Liu, J., Medo, M., Wakeling, J. & Zhang, Y. (2010), 'Solving the apparent diversity-accuracy dilemma of recommender systems', *Proceedings of the National Academy of Sciences* **107**(10), 4511–4515.

Zhu, H., Huberman, B. A. & Luon, Y. (2011), 'To switch or not to switch : Understanding social influence in recommender systems', *CoRR* **abs/1108.5147**.

Ziegler, C. N. & Lausen, G. (2004), Analyzing correlation between trust and user similarity in online communities, *in* 'Proceedings of Second International Conference on Trust Management', Springer-Verlag, pp. 251–265.

Zong, S., Kveton, B., Berkovsky, S., Ashkan, A., Vlassis, N. & Wen, Z. (2017), 'Does weather matter ? causal analysis of TV logs', *CoRR* **abs/1701.08716**.