Towards Context-Aware Personalized Recommendations in an Ambient Intelligence Environment

by

Mohammed F. Alhamid

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Due to the rapid increase of social network resources and services, Internet users are now overwhelmed by the vast quantity of social media available. By utilizing the user's context while consuming diverse multimedia contents, we can identify different personal preferences and settings. However, there is still a need to reinforce the recommendation process in a systematic way, with context-adaptive information. This thesis proposes a recommendation model, called HPEM, that establishes a bridge between the multimedia resources, user collaborative preferences, and the detected contextual information, including physiological parameters. The collection of contextual information and the delivery of the resulted recommendation is made possible by adapting the user's environment using Ambient Intelligent (AmI) interfaces. Additionally, this thesis presents the potential of including a user's biological signal and leveraging it within an adapted collaborative filtering algorithm in the recommendation process. First, the different versions of the proposed HPEM model utilize existing online social networks by incorporating social tags and rating information in ways that personalize the search for content in a particular detected context. By leveraging the social tagging, our proposed model computes the hidden preferences of users in certain contexts from other similar contexts, as well as the hidden assignment of contexts for items from other similar items. Second, we demonstrate the use of an optimization function to maximize the Mean Average Prevision (MAP) measure of the resulted recommendations.

We demonstrate the feasibility of HPEM with two prototype applications that use contextual information for recommendations. Offline and online experiments have been conducted to measure the accuracy of delivering personalized recommendations, based on the user's context; two real-world and one collected semi-synthetic datasets were used. Our evaluation results show a potential improvement to the quality of the recommendation when compared to state-of-the-art recommendation algorithms that consider contextual information. We also compare the proposed method to other algorithms, where user's context is not used to personalize the recommendation results. Additionally, the results obtained demonstrate certain improvements on cold start situations, where relatively little information is known about a user or an item.

Acknowledgements

I would like to acknowledge my sincere gratitude to my Ph.D supervisor Prof. Abdulmotaleb El Saddik, who always encouraged, supported, and guided me through my research. He was, and still is, my mentor, keeping me motivated with opportunities and inspiring ideas while I work toward the completion of this thesis. I am thankful for all his advice, not only in the field of research, but also to build a professional future career.

I am also thankful to all the members of the Multimedia Research Laboratory (MCR-Lab), and the Distributed and Collaborative Virtual Environment Research Laboratory (Discover Lab), with whom I worked for the past few years. I had the opportunity to collaborate with Dr. Majdi Rawashdeh, Dr. Heung-Nam Kim, Dr. Mohamad Eid, and Dr. Haiwei Dong. I acknowledge their feedback and assistance on various aspects during the progress of this work.

Words cannot express my heartfelt appreciation, thanks, and love to my wife Hind. Without her unconditional support, care, and encouragement, I would not have finished this thesis. I also especially thank my sweet daughter Aljazi, my beloved parents, my brother, and my sisters for their love and patience.

Lastly, I want to express my gratitude to my friends and colleagues at the University of Ottawa. I have been fortunate to know and learn from J. Arteaga, A. Alshareef, A. Barnawi, and B. Hafidh.

Contents

List of Tables

List of Figures

List of Acronyms

AmI Ambient Intelligence

AU Active Users

CBF Content-based Filtering

CCbT Collaborative and Content-based Technique

CF Collaborative Filtering

ECG Electrocardiography

ERD Entity Relationship Diagram

GFREC Graph-Based Flexible Recommendation

HF High Frequency

HPEM Hidden Preferences Explorer Model

HRV Heart Rate Variability

JMF-MS Joint Matrix Factorization with Mood Specific

LF Low Frequency

LSA Latent Semantic Analysis

MAP Mean Average Precision

MRR Mean Reciprocal Rank

NU Normal Users

PI Popular Item

PLSA Probabilistic Latent Semantic Analysis

PM Probabilistic Model

PU Passive Users

RS Recommender System

Chapter 1

Introduction

Users of music, movies and other media recommender systems want to find their preferred content easily and efficiently, with minimal effort. The rapid increase in the use of social media has changed the way users interact with the resources available online. There are many online multimedia channels (Netflix[1], Last.fm[2]), online stores (Amazon[3]), and social networks (Facebook[4]), which present and recommend different types of multimedia contents. Much effort has been made, through these online resources, to estimate interest and manipulate the user's input, in order to deliver a list of recommended items. These applications deal with the user's interest in different ways; some require an explicit interest input, such as item rating scores, in order for the application's algorithm to analyze the preferences from the already consumed items. Amazon is an important example, worth mentioning [62]. The user and item relationship is taken into careful consideration in order to personalize the website for an easy and powerful way to sell items online.

As people share their preferences and information on different popular social media sites, they share important information about their consumed multimedia contents, including their descriptive experience. For instance, a vast quantity of users on Last.fm, Youtube, and Facebook upload media content with annotation tags that describe their uploaded media. The challenge is personalizing such content in order to match the user's needs.

The prevalence of smart devices has increased the accessibility to different multimedia

[1]http://www.netflix.com
[2]http://www.last.fm/
[3]http://www.amazon.com/
[4]http://www.facebook.com/

contents through the social web. Also, the ability to embed our living environment with small and inexpensive sensors enables and accelerates the development of context-aware recommender systems that can obtain the most desirable content in a specific context. The combination of real-time access to different multimedia contents and overwhelming content options available on the social web increases the challenge of building a personalized search model. For these reasons, this field of study, related to helping users browse and consume multimedia contents, has been a challenge for researchers over the past few years [94, 111].

Different users have different personalities [75]; likewise, a user's choice of which multimedia content to consume varies with different parameters, including emotions and various physiological conditions [67]. It is therefore essential to integrate user preferences and physiological context in the development of a recommendation algorithm. Using today's technology, we can obtain a user's various physiological functions and monitor them using biomedical sensors, with the help of computer systems. Certain types of multimedia contents are used for relaxation, stress management, and mood modification therapies; researchers in the field of recommender systems became interested in predicting the user's context and the environment of the interactions, in order to enrich the user's experience and satisfaction.

Context-aware recommendations help us predict the interest or the preferences of a certain user, for a particular item. They collect more information than just the rating values given to resources by consumers. Such information helps overcome the difficulties of understanding the score values obtained from the user-item relationship. Detecting the user context and extracting useful contextual information improve traditional recommendation techniques and help to reduce the sparseness of the user-item datasets [91]. In addition, a user's context helps to associate the user-item relationship with the items' direct or indirect effects on the consumer. For instance, Reynolds et al. [85] experimented with the types of activities that affect the listener's mood. The experiment was conducted with 750 people, and studied their choices of music. Another study indicates that users perceive and interact with different multimedia systems, and their satisfaction level is not easily predictable. However, it does confirm the clear dependency of the user's choices on the context of the interactions [39].

Ambient Intelligent (AmI) refers to a set of promising technological approaches all aimed at improving the quality of life [27]. Our understanding of the AmI, according to the ISTAG (the European Commission Advisory Group) [27], is the ability to have an electronic environment that is sensitive, and responsive to the presence of people, and

that can work in an invisible way. With an AmI system, we are able to read the user's environment using numerous sensors, make reasoned judgments about the user context, and adopt several mechanisms to modify the surrounding environment using actuation interfaces.

There are many ways in which AmI can assist us in our daily lives, for example, environmental monitoring, safety monitoring, smart homes, education and workspace developments [21]. Specifically, one promising application for the AmI technology is supporting the interactions between users and multimedia entertainment systems. In a home environment, each individual has his/her own preferences on how to interact with a multimedia content. A person may like watching a movie in the living room with a certain degree of light. However, if that person is watching a movie with his/her kids, then the preference might change to accommodate other peoples' preferences. As another example, a person might like listening to loud background music in their room, but enjoy low volume relaxing music when working in their office. Therefore, the following questions become pertinent: Can a recommender system detect such preferences, and if so, how can such preferences be utilized?

The interaction between the user and the AmI is the key to enhancing the user experience with a recommender system, due to the variety of possible research applications and their related benefits. Homes, offices and other environments are already equipped with a number of devices that are able to interact and react with users. The availability of different smart sensors and their functional accuracy have enabled the development of different types of AmI-based applications. Moreover, these applications can also provide automatic interactions, where an application does not have to continuously rely on explicit inputs from users. In all the cases mentioned above, the combination of AmI technologies and multimedia systems provides immense potential for enhancing user experience and comfort level. Accordingly, we need an interactive and intelligent model that combines AmI with other sensors to enhance the user's satisfaction, and ultimately provide implicit inputs to the recommender systems, to create a responsive multimedia environment. To the best of our knowledge, no prior work has considered such comprehensive integration.

Using an AmI environment, we can collect and incorporate multidimensional contextual information into the recommendation process. A user's interest for multimedia content can be predicted based on the evaluation of the available contextual information and it's correlation with the user. User and collaborative user preferences are also considered. The proposed recommendation models facilitate the context assignments on items

by incorporating social tagging to discover the relevance of the item to the user's needs. Consequently, the user annotates and describes resources that can be used to retrieve context-based recommendation results, which is the main advantage of utilizing contextual information. Then, by finding the similarities between the user's context and various contexts and items, we can determine the appropriate items for a particular context. We then map the context on the items based on that particular user, and recommend the most relevant item for his/her needs.

1.1 Motivation

Nowadays, we consume different types of multimedia contents using a substantial variety of devices and platforms. These devices run independently, and have content sharing limitations. Considerable efforts are made to enhance the user experience. On the one hand, individual people have different reasons for browsing or consuming multimedia content, including entertainment, maintaining their well-being, or learning. On the other hand, smart engines, such as those built to provide recommendations, browse and manipulate the multimedia collections for a variety of business purposes such as advertising, studying consumer behavior, and other business related objectives. It remains challenging for both parties to select the most suitable, preferred, or appropriately related content using contextual information from the traditional user-item datasets [94]. As a result, it is worthwhile to examine context in a way that can be beneficial to personalized recommendation. However, lack of available contextual information is an important challenge when designing any context-based recommendation. The method of interactions between the user and recommender systems that require contextual information to provide feedback, is still a research problem that needs to be tackled. The estimation of suitability of a multimedia content to a given type of context is another issue to address. Such an estimation depends on the suitable selection of contextual dimensions to be considered in the recommendation process.

We argue that contextual information is an important factor during the recommendation process. Adomavicius et al. [3] place emphasis on the importance of considering the user's context for recommendations: "contextual information does matter in recommender systems: it helps to increase the quality of recommendations in certain settings". We also argue that in the near future, we will have multimedia contents that are already tagged and categorized, based on different contextual conditions available to the con-

sumers. For example, Sourcetone[5] is a music classification company that sorts music into different contextual categories such as music suitable for activity, health, or mood. We also benefit from the online social networks and the ability to have contents that are already tagged and rated with different emotional annotations, in order to simplify the search for the right contents for a particular context. Some previous related work has already shown potential results towards improving the user experience by incorporating contextual information, in filtering items [9], social networks [72, 113], biomedical sensors [64], user behavior and emotions [41, 52], and incorporating multiple contextual dimensions [94, 90, 42].

1.2 Motivating Scenarios

This section provides a few motivating scenarios that point out the benefits of the proposed model. These scenarios are referred to throughout this thesis, and are later used as target examples to evaluate the applicability and performance of the proposed recommendation model.

Let us take a typical family of four in a smart home environment as an example; there is a husband, a wife, and their two kids. The family occasionally gathers in front of the television to watch a movie or a television show. When the time comes to choose a movie, each family member has their preferences and their favorite movie stars, which they are interested in watching. Using a camera, we can detect who is standing or sitting in front of the television, and implicitly feed a multimedia server that is already installed in the house, with the identity of the person who is there to watch the movie. When one of the family members starts browsing the available list of movies, the multimedia server begins filtering out some movie suggestions based on the received contextual information, which, in this case, is the profile of each family member. The question is, would such information be enough to make good predictions and help the family pick the right movie? The answer is no, since they might also be interested in watching some Christmas movies if the date is late December, and if we know that they watched similar holiday movies last year around this time.

Today's smartphone accelerated the development of different context-aware recommender systems, particularly in observing the user's contexts such as weather, location, activity, and time. Smartphones also enable enormous access to multimedia collections

[5] http://www.sourcetone.com/

and act as a bridge between different sensors and the recommendation engine. Hence, we developed a smartphone application that can effectively be used in a home environment. Let us assume a user usually listens to music while studying and who has an exam the next morning. At home, the user employed the application's Electrocardiography (ECG) interface to capture their heart signal. Depending on the user's stress level, the application will suggest relaxing music according to his/her preferences. Additionally, as we explain in the design of the proposed recommendation model (Chapter 4), the application also explores the user's friends on social networks and their preferences. In this scenario, the application can not only recommend music, but it can also adapt the recommendation result to fit a user's context. Furthermore, the application can also modify the environment of the user's room, according to their preferences. It can make the room more comfortable for studying by adjusting the volume of the music and the level of light, using AmI interfaces.

1.3 Research Goals and Problem

Recommender Systems (RSs) should not only consider the properties of the content, the user profile, or the correlation between them, but should also consider different contextual information. In addition, when it comes to making a recommendation, the system should not be limited to recommending something new; it should also be capable of recommending the same content many times. The best example is music, where a recommender system could detect the user's interest in a song and keep playing that favourite song from time to time, depending on the context. Therefore, the recommender system does not have to recommend new items or a content that has never been consumed before in order to capture the user's interest.

In this research, we address the issue of how to improve the recommendation and the quality of the user experience by analyzing the contextual aspect of the users, at the time when they wish to consume multimedia content. We therefore focus on incorporating the contextual information collected through the AmI environment into the recommendation process. This thesis addresses the following requirements:

- Support the personalization of the multimedia recommendations according to a given context. This has been addressed in Chapter 4.
- Provide a recommendation mechanism to improve the user experience and satisfaction with the use of a biosignal in the recommendation process. This has been

addressed in Chapter 3, Section 3.2.

- Match the contents with the physiological and the environmental conditions of the user. This has been addressed in Chapter 4.
- Capture the user's preferences in an implicit way. This has been addressed in Chapter 3 (Section 3.1.2) and Chapter 5 (Section 5.1.4).

This thesis addresses the following research questions:

- What are the effects of adding a bio dimension to the delivery of multimedia contents to the user?
- How do we predict the user's preferences toward content, while incorporating contextual information into the recommendation process in order to improve the user's satisfaction?

1.4 Thesis Contributions

In this thesis, we focus on recommending multimedia content to users based on their instantaneous requirements, which can be represented by categorizing their contexts into physiological and environmental parameters, in order to enhance their experience and comfort level. Additionally, we propose a model to bridge the gap between the user's context and the available multimedia recommendation options. The model personalizes the multimedia recommendations according to the given context. We also propose two multi-modal applications for a feasibility demonstration of the proposed model. The proposed applications use interactive multi-modal interfaces to collect physiological and environmental data in real-time from the user, and provide the means for proportional recommendation responses.

The proposed recommendation model benefits from the large number of existing online users who contribute to the classification and annotation of different media. This model is more effective than just relying on feature extraction since, for example, our model avoids the unrealistic need to analyze innumerable items on the Internet in order to detect the context. Accordingly, we propose a feasible solution to find relevant media content based on the detected contextual information.

The proposed recommendation model makes the following contributions toward recommendation models: First, the model identifies the hidden preferences of users in specific contexts, the hidden preferences of items toward contexts, as well as the hidden

preferences of users toward new and consumed items. The proposed model takes into consideration the need to have an easily applicable model to fit the scope of specific context-aware recommendation applications. Additionally, the model considers the contextual information by reflecting on the social tags available online to explore the hidden preferences between users, items, and contexts. Specifically, the model has the ability to incorporate not only the preferences of a single user in a given context, but also the choices other consumers made in a similar context. Second, we propose a rating algorithm for context-based items, which bridges the gap between the media resources, users' individual and joint preferences, and the detected contextual information. Third, we experimentally show how the proposed recommendation model can be optimized to maximize the Mean Average Precision (MAP), and how it identifies relevant items for the user's query. The contributions of this thesis can be summarized as follows:

- **Designing and developing a context-based recommendation model:** The proposed model captures and exploits contextual information from the user's environment and employs social tagging to maximize the benefit of the extracted contextual information in the recommendation process. The contributions of the proposed model include:
  - Designing three different methods of discovering hidden preferences: the context-boosted, the context-user-boosted, and the context-item-boosted.
  - Designing an optimization method to utilize the explored hidden preferences.
- **Modeling and analyzing a recommendation algorithm to improve user satisfaction and experience:** We propose a recommendation algorithm to improve the user experience and satisfaction by using a biosignal in the recommendation process. In particular, we highlight the importance of the physiological aspect of the user's context during the recommendation process.
- **Analyzing the effectiveness of the proposed recommendation model on two real-world datasets:** We examine how to enhance the quality of the recommendation with two main problems: the cold start of new users and items, and the recommendation of items to a group of users and to individuals. The contribution also includes:
  - Designing and developing two prototype applications, proposed in order to collect the required contextual information and evaluate the recommendation performance.

1.5 Scholarly Output

Research Resulted in Refereed Journals

- **Mohammed F. Alhamid**, Majdi Rawashdeh, Haiwei Dong, and Abdulmotaleb El Saddik, "RecAm: A collaborative context-aware framework for multimedia recommendations in an ambient intelligence environment" *Multimedia Systems*, 2015. (Under review process)

- **Mohammed F. Alhamid**, Majdi Rawashdeh, and Abdulmotaleb El Saddik, "Towards Context-Aware Media Recommendation Based on Social Tagging" *Journal of Intelligent Information Systems*, 2015. (Major revision)

- **Mohammed F. Alhamid**, Majdi Rawashdeh, Hussein Al Osman, M. Shamim Hossain, and Abdulmotaleb El Saddik, "Towards context-sensitive collaborative media recommender system" *Multimedia Tools and Applications*, 2014. (Published on September, 2014)

Research Resulted in Refereed Conferences

- Majdi Rawashdeh, **Mohammed F. Alhamid**, Heung-Nam Kim, Awny Alnusair, Vanessa Maclsaac, and Abdulmotaleb El Saddik. "Graph-based personalized recommendation in social tagging systems." Proceedings of IEEE International Conference on Multimedia and Expo Workshops, 2014.

- M. Anwar Hossain, Atif Alamri, **Mohammed F. Alhamid**, Majdi Rawashdeh, and Awny Alnusair. "Collaborative Recommendation of Ambient Media Services." Proceedings of IEEE International Conference on Multimedia and Expo Workshops, 2014.

- **Mohammed F. Alhamid**, Majdi Rawashdeh, and Abdulmotaleb El Saddik, "Towards context-aware recommendations of multimedia in an ambient intelligence environment," Proceedings of the 9th IEEE International Workshop on Multimedia Information Processing and Retrieval, 2013.

- **Mohammed F. Alhamid**, Majdi Rawashdeh, Hussein Al Osman, and Abdulmotaleb El Saddik, "Leveraging biosignal and collaborative filtering for context-aware recommendation," Proceedings of the 1st ACM International Workshop on Multimedia Indexing and Information Retrieval for Healthcare, pp. 41–48, 2013.

- **Mohammed F. Alhamid**, Mohamad Eid, and Abdulmotaleb El Saddik, "A multi-modal intelligent system for biofeedback interactions," Proceedings of IEEE International Symposium on Medical Measurements and Applications, 2012.
- **Mohammed F. Alhamid**, Mohamad Eid, Abdulrhman Alshareef, and Abdulmotaleb El Saddik, "MMBIP: Biofeedback system design on cloud-oriented architecture," Proceedings of IEEE International Symposium on Robotic and Sensors Environments, pp. 79–84, 2012.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows: Chapter 2 introduces existing work and background research related to context-aware recommendations. Chapter 3 presents the detection of the user context including the detection of the user's physiological parameters. Chapter 4 describes in detail the proposed model, which includes the search for the hidden preferences and the calculation of the predicted rating values. In addition, Chapter 4 presents the optimization function. The experimental evaluation, the analysis of the computational cost, and the related results are discussed in Chapter 5. Chapter 5 also highlights certain scenarios as well as the usability of the proposed model by presenting implemented recommender applications. Finally, Chapter 6 concludes the thesis, briefly describing our future work.

Chapter 2

Background and Related Work

This chapter focuses on existing work in the area of multimedia recommendations, context detection, and context-aware computing. In addition, it highlights how this thesis relates and differs from existing works available.

2.1 Types of Recommendation Techniques

In recent years, the development of RSs has attracted considerable research interest in an attempt to discover the relation between users and the items they consume. Previous studies analyzed the hidden relationships between users and items, and proposed several methodologies that are now standard ways of establishing recommendations.

Traditional recommendation algorithms recommend items (i) to user (u) by considering different attributes and dimensions. These attributes can consist of a user's explicit or implicit ratings, or of previous interest in similar items. Such systems gather and analyze other information in order to make recommendations, such as the similarity between users and the similarity of items consumed in common. The recommendation techniques either depend on information about the items (content-based), information about the correlations between users (collaborative), or a combination of both types of information (hybrid). Incorporating contextual information has enriched and enhanced traditional recommendation approaches.

Content-Based Filtering (CBF)

The CBF approach recommends items based on the analysis of the item contents and creates a profile with assigned features such as type, category, and special attributes. The

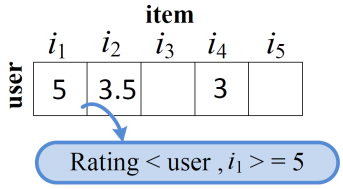


Figure 2.1: Illustration of the content-based recommendation model.

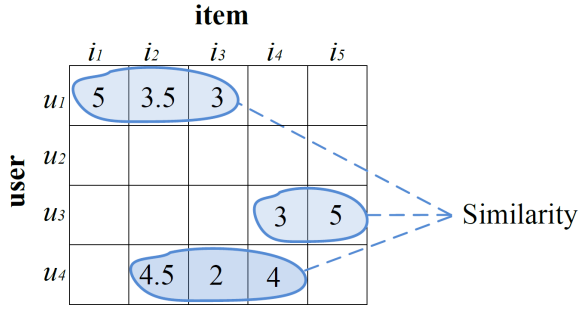


Figure 2.2: Illustration of the collaborative filtering model.

resulting profile is matched with the explicit ratings that represent the user's experience with that particular item, as illustrated in Figure 2.1. This approach is widely used to recommend music, where a user's previous rating scores and their historical selection of music help predict their degree of interest in the new items. Thus, this approach focuses on the analysis of previously consumed items, and predicts the user's interest in the new items based solely on the target profile.

Collaborative Filtering (CF)

CF recommends items based on the analysis of users rather than only items. For instance, first finding similarities between users who share the same interests, and then finding all items that have not been consumed by that particular user or group. The social aspects and the concept of sharing interests are highly considered in this approach. As opposed to CBF, CF gives more attention to social aspects and to the feedback of similar users to unknown user-item ratings, as illustrated in Figure 2.2.

Hybrid Recommendation

To overcome certain problems that exist in CBF and CF and increase the performance of the recommendation, the hybrid technique combines the approaches of the two recommendation approaches. Specifically, CF depends on the distance between users, based on the number of items that they share. This approach may consider users to be close to each other based on the calculated distance, when in reality they might have completely different tastes. Another problem is the cold-start problem [105]. This happens when a user has recently joined the system or when he/she rarely takes the time to submit feedback to the recommender system (e.g. by rating an item). The combination method has shown positive impacts on the accuracy of the recommendations [19]. A hybrid approach is presented by Lekakos and Caravelas [61], where movies are recommend by analyzing the content and the collaborative relation between users.

2.2 Context-Awareness and Recommender Systems

2.2.1 Context-Aware Computing

Nowadays, the combination of technology, which targets the interactions between users and intelligent systems, and a variety of recommendation services, is getting more attention from industry. Specifically, the focus is on the creation of a better environment for the user to obtain and consume online resources. Usually, estimating the user's interest for an online resource is a challenge. This is true since the user's interest depends on multiple selection criteria, which can be considered as their contextual dimensions. Accordingly, in order to personalize a recommendation, we need a method that can identify the user's state and collect all the related must-have contextual information about their environment. Therefore, recently, context-aware computing has been an area of high interest in the field of multimedia adaptation and recommendation.

Context-aware computing and the concept of adaptations support AmI-based systems. For instance, Ceccaroni and Verdaguer [16] designed an interactive mirror as the main interface to provide an interactive television, reminders, personal motivation content, and a personalized set of multimedia displays such as the weather forecast. Aside from the conceptual design of their model, some services were simulated rather than real implementations. Onur Asan et al. [7] proposed another interactive mirror that is to be placed in the bathroom, to allow the user to perform different activities such as checking emails, the weather forecast, or their daily schedule. Phillips Research HomeLab [57]

built an intelligent bathroom using an interactive mirror. The mirror was to be used to display visual output resulting from context-aware recognition of the bathroom environment. For instance, the mirror would have displayed the weather forecast or traffic information during regular activities like shaving. In addition, the mirror would act as a weight coach application to take the users through their latest weight loss/gain progress. Using the mirror, children could enjoy watching a cartoon while brushing their teeth. Dey et al. [25] proposed a software architecture that envisioned the use of context-aware computing.

2.2.2 Context-Aware Recommendations

Contextual information boosts the effectiveness of traditional recommender systems, which consider only the user, items, or their related information to find the next expected item to be selected. For instance, the user may like to listen to a particular type of music while running in the morning and not at other times of the day. Accordingly, if the context of the user has been well detected, the recommendation results might be completely different from the traditional ones, and better suited to the user's needs [94]. Therefore, using contextual information to enhance the quality of recommendations has shown positive results in a number of studies. Different context dimensions can be considered according to the application's scenarios, such as time, season, location and weather, as well as social and physical conditions (e.g. stress level and heart rate). Such contextual information can increase the quality of the recommendation and enhance the user's experience [2].

Various algorithms examine the use of the traditional recommendation approaches boosted with contextual information to enhance the recommendation results. One of the earliest works in adapting CF for context-based recommendations is done by Chen [17]. Chen proposed the basic idea of using context similarity through collaborative filtering. In addition, a hybrid technique proposed by Woerndl et al. [102] uses collaborative filtering for context recommendation in smartphone applications. The authors present the effectiveness of applying the hybrid approach as a successful way of accommodating a contextual type of information. L.M. de Campos et al. [22] presented the recommendation model that combines CBF and CF. Context is also considered as a content feature and it is assumed that contextual elements are independent, so they can apply the Bayesian network algorithm on them. Bogers [13] proposes a Markov random walk algorithm over a contextual-based graph on a movie dataset. The user to item ratings,

including contextual information such as associated tags, movie genre, and individual actors, are all considered contextual information in the graph. Hyung et al. [43] proposed a method that uses a Latent Semantic Analysis approach (LSA) and a Probabilistic Latent Semantic Analysis (PLSA) on textual inputs obtained from the user to find similar documents for music recommendations. Wang et al. [100] use the Naïve Bayes model on annotated music contents to recommend music based on user activity. The music annotations consist of six activity categorizations: running, working, sleeping, walking, shopping, and studying. They then relate these activities to 1200 music tracks crawled from the Internet and annotated by invited subjects. Yu et al. [110] use an Ontology model to represent context. The model consists of a number of contextual situations, where each media is evaluated and attached to a preference feature and a weight. The weight ranges between -1 to 1, representing the likelihood that a user will like that specific content. Zhang et al. [114] proposed a virtual rating method to enhance the recommendation accuracy using CF model.

Contextual information is found within a smart environment for media recommendations. For instance, in a smart car environment, Balturnas et al. [8] suggest playing different music according to the traffic conditions and the driver's mood. Using a collaborative filtering approach, the user's preferences along with the available contextual information are grouped together to select the most suitable music to play. However, the work presented in [8] focuses mainly on single user preferences. Amato et al. [6] built a smart context-aware environment for cultural heritage applications. Based on the user's location in a smart city, they can browse and explore that city from a tourist's perspective. Yang et al. [103] proposed a RS that considers the user's shopping location to recommend offers and promotions to the user. Based on their location in the shopping area, shoppers would receive promotions and special offers in the form of web pages, via their smartphone. Similarly, Noguera et al. [77] developed a 3D map for location-based recommendations. During the recommendation process, contextual information is used for website recommendations, where a list of already visited websites, web-based interactions, and social related information is exploited to predict the user's future interest [83].

Contextual information conveys the user's interests more accurately to the recommendation process. Some earlier studies proposed contextual recommendation algorithms by building a multidimensional approach. Adomavicius et al. [1] were some of the first to work on incorporating contextual information into the user-item matrix using a multidimensional model. The model inspired other researchers to enhance the recommendation

results; for instance, Weng et al. [101], focused on proposing a solution to the rating hierarchy contradictions in the multidimensional model. Another study that considered music extracted features as a context for music recommendation is proposed by Yoon el al. [106].

Recently, RSs rely on social networks in order to collect vast amounts of information about the user, their multimedia choices, and their behavior. For instance, Zangerle et al. [113] presents a facilitation approach to collect a user's music dataset from Twitter, assuming the user's music player shares such information on Twitter. More recently, matrix factorization is a technique used in different algorithms proposed to target large-scale recommendation problems. For instance, studies such as [4],[47], and [88] presented different approaches to combine contextual information using matrix factorization. The factorization approach has gained popularity and is being used to improve collaborative filtering. Shi et al. [89] propose a MAP optimization approach to the algorithm proposed by Kolda et al. in [55], aiming to build a fast learning model. Another study by Rendle et al. [84] present the experimental results of a factorization machine on a Yahoo! Webscope dataset, along with another movie dataset called Adom, where the authors wanted to reduce the complexity of building their method. Additionally, the work in [84] shows a better rating prediction than another factorization method named Multiverse, proposed in [48]. An extensive literature review of user and context related recommendation survey can be found in [87].

2.2.3 Physiological-Related Recommendations

Today's technology enables individuals to become aware of their various physiological functions and monitor them using biomedical sensors and computer systems. One of the purposes of consuming a certain type of multimedia content is maintaining an individual's well-being. For instance, music has been used in different ways to promote relaxation and control the level of stress. Krout [56] discusses the importance of a user's preferences when choosing the right type of music to relax. In another study [85], 750 people surveyed indicate that the type of activity being done affects the listener's mood, thus affecting their choice of music.

Recently, the use of biosignals in the recommendation process has attracted researchers' attention. Some researchers propose to recommend items that change the user physiological state from a condition to another by having continuous heart monitoring. For instance, [65] proposed a method that recommends a music playlist that

could bring the user's increased heart rate back to the normal range. Another group of researchers analyzed the content features in order to extract some emotion related factors. In this case, a music content analysis tool is used to find whether a particular song contains sad, happy, or other emotional features that can be extracted. Thereafter, a recommendation algorithm matches the explicit emotional inputs from the user with the available music items to generate the recommendation list, as proposed in [115, 76, 94]. For instance, Han et al. [31] propose a music recommendation model that alters the user's emotional state using the recommended music. Authors use a proposed ontology to model the relationship between the musical items and the related emotion. Cai et al. [15] extracted emotion tags available on a web document in order to match them with music lyrics.

Multimedia contents have also shown certain effects on respiration and heart rate rhythm. Bernardi et al. [10] discovered some cardiovascular and respiratory changes triggered by music. Their work shows the effects of music on the level of relaxation, where slow tempo music brings relaxation and lowers the body's attention level as a result of the heart rate and blood pressure. Their work also shows that slow or meditative music could actually bring relaxation effects, and that faster rhythmic music could heighten an individual's concentration. In their study, Liu et al. [67] addressed the effects of music playlist recommendations on the heart, while flying on an airplane. Since the airplane environment causes discomfort and stress, authors used a heart sensor embedded in the aircraft's seat to monitor the heart activities and particularly to measure the stress. This study [67] focuses on analyzing the airplane environment and the effects of music on the user's level of stress, which differs from the focus of our thesis.

Pessemier et al. [80] discussed activity detection in a mobile environment as supportive information for the recommendation process. By using an attached accelerometer on a mobile device, the authors were able to detect four basic activities: running, walking, standing, and cycling. Based on the detected activity and on other contextual information such as location and weather, the recommender system can fetch different categories of information in the surrounding area, such as train schedules and restaurants. In another study, Kim et al. [54] employed a hidden Markov model on a contextual information dataset to recommend the most appropriate menu for healthcare services.

Music is widely used as an entertainment medium for people performing physical fitness activities. Oliver and Flores-Mangas [78] proposed a system called MPTrain. which continuously tracks user movements and plays different music according to the detected activity. Acceleration and heart rate are collected using a smart phone and the

rhythm of the music is modified based on the accelerations of the movements and the range of the heart rate. The system is developed to help users achieve their exercise goals by speeding up or slowing down their activity by changing the type of music being played. In another study, Nirjon et al. [76] proposed a context-aware recommender system that uses an earphone device equipped with a heart rate detection sensor. The sensor collects continuous ECG signals and extracts the peaks in order to instantaneously calculate the average heart rate. Similarly, the system extracts offline beats from the music signal, or what is known as the tempo of the music, as well as the frequency of the sound, and the energy of the signal.

2.3 Experimental and Environmental Difficulties

One of the biggest challenges in context-based recommendations is finding a dataset that already incorporates contextual information. Due to the difficulties in collecting and acquiring contextual information, it is very hard to find such a dataset for researchers to examine and experiment with the effectiveness of any proposed context-based recommendation algorithm. Yujie and Licai [111] point out this difficulty as one of the challenges of context-aware recommender systems. Therefore, many studies build testing applications that can subjectively be evaluated by users such as [100, 95], by generating synthesized datasets for experimental analysis, like the work presented in [94]. Table 2.1 presents a summary of the context-based recommendation methods available in the related literature on evaluating the recommendation results. Table 2.2 presents other related work that tests their proposed recommendation algorithms on crawled datasets from online databases.

2.4 Comparison to Existing Studies

This section of Chapter 2 highlights the main differences between the proposed Hidden Preferences Explorer Model (HPEM) and the existing related work. This section has been divided into two different subsections.

2.4.1 Incorporating Contextual Information

Some earlier studies present the importance of identifying the user's context in order to personalize the outcomes of a RS. Such previous work usually incorporates the con-

Reference work	Experiment Setup	Number and types of content	Number of subjects
Adomavicius et al. [1]	Collected ratings using a developed movie web-based tool.	202 movies	117 students, dropped to 62 students (due to minimum number of ratings)
Park et al. [79]	Subjective evaluation.	322 songs	10 students
Kim et al. [53]	Evaluated recommendation performance using a 3-question survey.	300 pop songs	50 users
Choi et al. [19]	Recommended restaurants, news, and movies using push messages.	Recommendation messages	200 mobile users.
Lee and Lee [59]	Evaluated performance using training and test datasets.	music	659 customers of a Korean music streaming company
Lui et al. [68]	Collected ratings using a system platform.	115 songs	56 users.
Baltrunas et al. [8]	Collected ratings using a developed ratings web-based tool.	139 music tracks	59 users evaluated the relevance of contextual information.
Nirjon et al. [76]	Collected music recommendation ratings using a smartphone.	Music	37 users, 48 patients, and 17 invited users for evaluation participated in the experiment.

Table 2.1: A summary of related literature about evaluating context-based recommender systems

Reference work	Number of users	Number and types of content	Source	Types of context
Shin et al. [92]	200 users	22,214 music tracks	Last.fm[1]	Time log
Mesnage et al. [72]	68 users	252,000 music tracks	Facebook and Last.fm	Contextual tags
Balturnas et al. [8]	59 users	139 music tracks	MusicLoad	Car driver situation.
Wang et al. [100]	10 users (for evaluation)	1200 songs	Grooveshark and Youtube	Songs titles match real activities.

Table 2.2: A summary of related literature about evaluating context-based recommender systems based on crawled datasets.

textual information into the recommendation process in two ways: during the search process (pre-filtering), and by adapting the recommendation results to the user's choices (post-filtering), such as in the work published by Pombinho et al. [81]. Incorporating contextual information can be referred to adapting a recommendation algorithm to consider a contextual type of information. Adaptation can be defined as adapting the outcome of the recommendation based on the contextual information detected, such as adapting the content to fit the size of the user's browsing device, network bandwidth, location, language etc. [110]. In addition, adaptation can also be referred to as filtering the recommendation process according to the context of the consumer [81, 73].

Other studies target different issues related to context detection, context-adaptation, or context-based recommendations. For instance, Hossain et al. [40] address the problem of selecting media services for a home environment in an adaptive way. It was said that user context and satisfaction were not being properly considered in the previous work, in light of the dynamic and interactive nature of interactions. They proposed an ant-based framework that considers the quality of service and the user's preferences in the process of delivering different media in a home environment. The framework in [40] is not intended to address the recommendation problems, but rather shifts the focus to how the user's preferences are being updated and how such preferences can be logged according to the detected context.

Although certain contextual parameters have already been incorporated in the recommendation process in a number of existing works, there is a lack of a general formulation on how to employ contextual parameters as dimensions in the traditional user-item recommendation model. What type of contextual information should be considered as context is also a challenging question. Some researchers build contextual recommendation dimensions on different attributes. We mentioned earlier that time, location, companion, etc., enhance the quality of the recommendation in certain scenarios, based on some related work. Table 2.3 summarizes what type of information has been considered, in the literature, as a context for recommendations. Additionally, it presents a summarized comparison of the main features and contextual dimensions of the previously listed frameworks and recommender systems available in literature. We arranged the column entries in Table 2.3 based on the year of publication of the research work.

Our work differs from all the presented studies in this section in such a way that our proposed context-aware recommendation model considers the physiological condition of the user as a useful context parameter to enhance the recommendation results. Another difference is that our approach recommends items that match the detected context and does not try to modify the physiological condition. In addition, our model does not require the analysis of the content of the media item itself, for example the content of the image or the video. Instead, we rely on the user's available context assigned to items as tags.

Reference work	Context-Based Recommendation	Context Information Examples												Recommendation Content
		Time	Mood	Gender	Social (friends)	Weather	Single Personal preferences	Joint Preferences	History	Location	Social Tags	Ambient	Biosignal	
Oliver and Flores-Mangas [78]	◆	-	-	-	-	-	-	-	-	-	-	-	Heart rate and accelerations	Music
Elliott and Tomlinson [29]	◆	-	-	-	-	-	-	-	-	-	-	-	Accelerations	Music
Choi et al. [19]	◆	◆	-	-	-	-	◆	-	◆	◆	-	-	-	Push messages
Cai et al. [15]	◆	-	-	-	-	-	-	-	-	-	◇	-	-	Music
Yang et al. [103]	◆	-	-	-	-	-	-	-	◆	◆	-	-	-	Offers and promotions
Liu et al. [68]	◆	◆	-	◆	-	-	-	-	-	-	-	-	-	Music
Liu et al. [66]	◆	-	-	-	-	-	-	-	-	-	-	-	Heart signal	
Baltrunas and Ricci [9]	◆	-	-	◇	-	◆	◆	◆	◆	◆	-	-	-	Movies
shi et al. [90]	◆	-	◆	-	-	-	◆	◆	◆	-	◆	-	-	Movies
Su et al. [94]	◆	◆	-	-	-	-	-		-	◆	-	Light, and temperature	-	Music
Baltrunas et al. [8]	◆	-	◆	-	-	-	◆	-	-	-	◆	-	-	Music
Mesnage and Rafiq [72]	-	-	-	-	◆	-	◆	◆	◆	-	◆	-	-	Music
Hu and Ogihara [42]	◆	-	-	-	-	-	-	-	◆	-	-	-	-	Music tracks
Amato et al. [6]	◆	-	-	-	-	◆	◆	-	-	◆	-	-	-	Information and relevant sites to visit
Nirjon et al. [76]	◆	-	-	-	-	-	-	-	-	-	-	-	Heart rate and accelerations	Music
Pombinho et al. [81]	◆	-	-	-	-	-	-	-	◆	◆	-	-	-	Locations points of interest
Popescu and Pu [82]	◆	-	-	-	-	-	◆	◆	-	-	-	-	-	Music playlist

Reference work	Context-Based Recommendation	Context Information Examples													Recommendation Content
		Time	Mood	Gender	Social (friends)	Weather	Single Personal preferences	Joint Preferences	History		Location	Social Tags	Ambient	Biosignal	
Wang et al. [100]	◆	-	-	-	-	-	-	-	-	-		◇	Noise	Accelerations	Music
Yu et al. [108]	◆	◆	-	-	-	-	◆	-	-		◆	-	-	-	Activity content (multimedia, games, etc.)
Hu [41]	◆	◆	-	-	-	-	◆	-	-		-	◆	-	-	Music tracks
Hyung et al. [44]	◆	-	-	-	-	-	◆	-	◆		-	-	-	Music	
This thesis	◆	◆	◆	◇	◆	◆	◆	◆	◆		◇	◆	Light, TV, speakers	Heart signal	Multimedia

Table 2.3: A summary of related literature for context-based recommender systems.(◆: real contextual data is considered or satisfied, ◇ synthetic, semi-synthetic or partially satisfied, - : contextual information is not considered)

2.4.2 Recommendation Methodology

There are a number of differences between the proposed HPEM model and the context-aware recommendation algorithms described earlier. The proposed model gives more importance to the individual user's context, including their contextual history, whereas little attention is paid to the incorporation of collaborative contextual-based dimensions into the recommendation process, as in [13, 43]. For instance, the work presented by Oliver and Flores-Mangas [78] does not consider personal preferences, the history of the user, or whether the user is actually enjoying the music being played. Moreover, the social aspect has not been considered during the process of recommendation, for example what is the most common music played while a person is jogging, or what are the most played songs among the user's friends, for such an activity. This study also did not show the correlation between the quality of the workout and the recommended music by only considering the heart rate and the acceleration data.

The proposed HPEM model does not require the expensive process of feature extraction of available contents online in order to analyze their contextual aspect, such as in [107]. For instance, Su et al. [94] proposed a system called uMender that predicts the user's interests and recommends music by taking contextual information into consideration. uMender performs recommendations by mining available contextual information and matching already preprocessed musical features collected offline. A database of music that has already been processed for feature extraction is mandatory to create a database of patterns that can be used for online recommendations. This approach may work for music, but not for other types of multimedia where feature extraction becomes a real challenge. Moreover, such a structure requires more explicit interactions to update and add musical contents by the user in order to be considered for online recommendations. In another study, Yazdani et al. [104] confirm that extracting emotional information is a challenging task and that the results are highly dependent on different personal and contextual aspects.

The proposed HPEM model relies on the collaborative technique in situations where there is a lack of user-item-context rating information. This issue is one of the critical challenges in the domain of context-aware recommendations [109]. We therefore propose constructing three models to search for the hidden preferences in other similar users, items, and context logs. We also noticed that many existing approaches rely on private synthesized datasets that have some collected contextual information. Most of these datasets are not publicly available for experimental research. Such an issue is also noted

in similar studies [84]. In Chapter 5, we discuss in more detail the comparison of the proposed model and some of the previous related work.

2.5 Summary

With an ever-increasing accessibility to different multimedia contents in real-time, it is difficult for users to identify the proper resources from such a vast amount of choices. Users of music, movies and other media recommender systems wish to find their preferred content easily and efficiently, with minimal effort. Recently, social networks and smart devices have allowed users to easily access and consume a variety of media contents.

RSs try to increase the level of a user's satisfaction by analyzing the multimedia content or the amount of information available about the user. However, the user's preferences and choices in consuming different multimedia products should not overcome the importance of contextual information. For instance, users' choices depend on various scenarios such as time, location, mood, or their companion at that time. Such contextual information can increase the quality of the recommendation outcome and enhance the user experience [2]. The reason is that contextual information appropriately conveys the user's interests during the recommendation process. Content-based filtering and user-collaborative filtering are at the core of the fundamental recommendation strategies used to process the user's queries and find online and offline resources. Combining these two strategies helps us find hidden associations between the characteristics of content, and trace the user's tastes, based on a collaborative analysis of their consumption history. The quality of the recommendations is enhanced with the use of the user's context at the time of his/her query, as well as with all the possible contextual information about the consumed contents. The recent advancements in technology that target the methods of interaction between the user and the variety of provided recommendation services are getting more attention from industry, who wish to create a better environment for the user. We briefly bring attention to the different existing recommendation techniques, and particularly to the ones based on context-awareness. We also review existing approaches that explore context detection and context-aware computing during the recommendation process. Finally, this chapter shows how the related research work relates to this thesis, and how this thesis differs from the existing research.

Chapter 3

Context Dimension

3.1 Defining Context

Contextual information is the primary focus of this thesis, and is used for multi-media recommendations, including method of delivery. This section explains how the simple User-Item technique, which we introduced in Chapter 2, can be enhanced by using a collaborative method to improve the recommendation accuracy with additional contextual dimensions and social tagging. We use the term "Items" (abbreviated to I) to represent any media resources that a system is supposed to recommend, such as movies or music. Hence, the new dimension is incorporated into the User-Item ratings to generate the recommendation list. Figure 3.1 shows a multidimensional model that incorporates a contextual dimension into the prediction of user interest $< u, i >$.

Let us assume we have (n) dimensions, $D = \{d_1, d_2, d_3, ..., d_n\}$, which represent certain context parameters, and each dimension is given a rating value, which is basically a system prediction of suitability or interest for user (u) in item (i) for parameter (d). These attributes can be estimated to filter and find items that are probably of interest to the user. The use of an implicit prediction rather than explicit inputs from the user is an interactive way of estimating ratings based on specific attributes. For example, if the system suggests a song (s_k) to the user, and the media player registers the fact that the user switches to another song or very quickly stops playing the song altogether, it indicates that the user is not actually interested in that song (s_k). Here, the user does not have to bother telling the system what he/she feels about the song, since it might be a boring and time consuming task to do so. In addition to the rating, each context is assigned to a relationship between u_x and item i_y. We assume that a user u receives a list

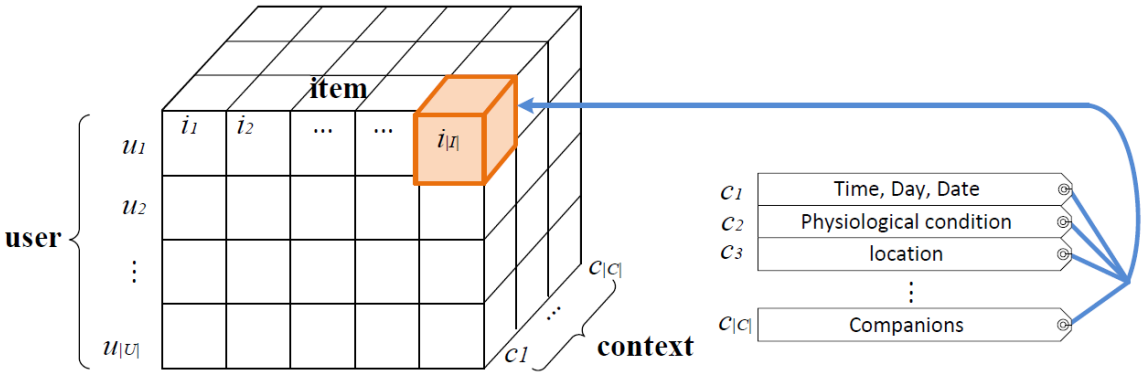


Figure 3.1: Different contextual parameters can represent the contextual dimension in the recommendation model.

of recommended items that are relevant to a given context $c \in C$. Any RS can benefit from our proposed model of recommending items based on multiple contexts, either by combining contexts into a new grouped context assignment, or by using an aggregation function to select the most influential context among them [92].

3.1.1 Types of Context

Context can be any information that helps identify the user's condition, including his/her environment, which can be represented in various attributes. For instance, context can be the location of the user, the date, the time, a companion, and a mood, as shown in Figure 3.1. Some of these parameters have already been mentioned or analyzed in different recommendation models and frameworks [94, 39, 110, 1]. Table 2.3 in Section 2.4.1 summarize the types of information most commonly used as contexts for item recommendation.

The HPEM model is able to accommodate different contextual parameters. Dey et al. [25] identify context under three main categories as a user's locations (rooms, workplaces, etc.), things (physical components, cellphone, etc.), and people (individuals and groups). Each of these categories can be expanded with additional attributes. For instance, the three main categories can be classified into four sub-categories: identity, location, status, and time [25]. Dey et al. [26] recognize context as "the user's emotional state, focus of attention, location and orientation, date and time, objects and people". We enlarge the structure in [26] to distinguish between different contexts in the experimental part of the

Category	Context Information Examples
Location and Orientation	Rooms, workplaces, in front of the computer, in front of the TV
Things	Physical components, cellphone
People	Individuals, groups, alone, with a partner, family, friends
Time	Morning, during the day, evening, weekday, weekend
Date	Day, month, season
Emotional State	Relaxed, stressed, happy, sad, lazy
User's profile	Sex (male, female), age (child, adult, senior, young)
Weather	Clear, rain, snow, sunny, cloudy
Season	Summer, autumn, winter, spring
Temperature	Cold, warm, hot

Table 3.1: A summary of the types of contexts that can be used to personalize the recommendations.

proposed recommendation process, as summarized in Table 3.1.

Identifying the user context is only one step in the personalization of the outcome recommendations. We still need to determine how to detect the context of all the items that have previously been consumed by other users in the dataset. Obtaining such information is a long process that involves recording all possible combinations of contextual dimensions in order to build a rich contextual dataset, which is almost impossible to find online. However, by using additional information that describes the content, specifically social tags from online social networks, we can not only collect contextual annotations for items, but also analyze the relationship between items and the users who tagged them. For example, a user on last.fm tagged a music track (x) with a "relaxing" tag. By finding how many users have used such a tag to describe track (x), we can find the best list of tags used to describe that track and find all other items that they think is "relaxing" as well. To this end, in this thesis, we propose a model to bridge the gap between the user's context and the available multimedia recommendation options.

3.1.2 Context in the Recommendation Steps

The proposed model distinguishes between implicit and explicit requests from the user, for the choice of a content category. For instance, the user picks a smart remote controller to request a list of recommended movies. Such an explicit request would be dealt with

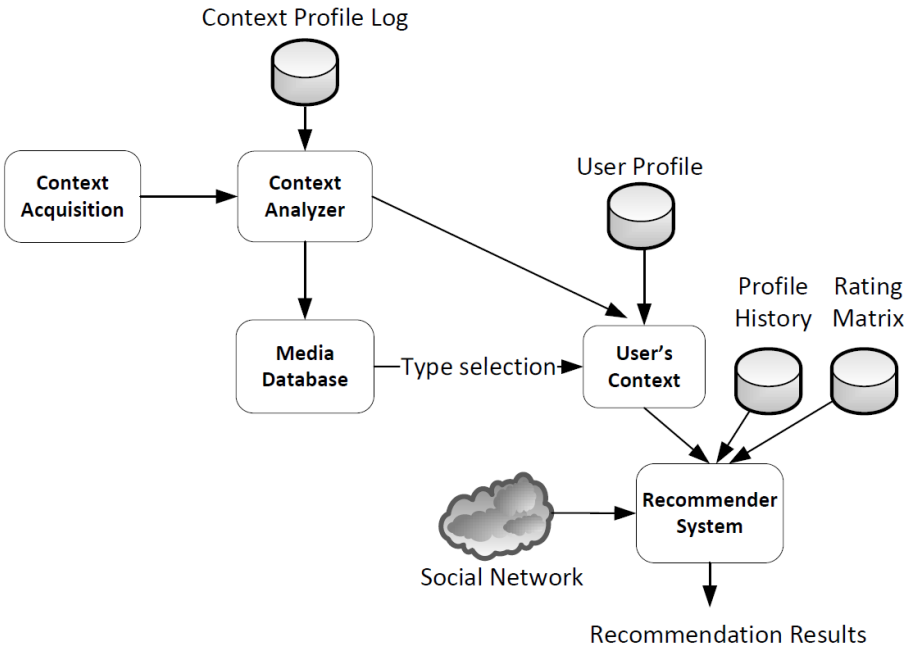


Figure 3.2: High-level context-based recommendation process.

differently than another recommendation scenario, where a system proposes a category of multimedia contents such as music. For instance, the system detects from the context log that a user regularly listens to music while working in their home office between 6:30-8:00 pm on weekdays. The system will not recommend a movie if the time and the location match a preferred category, such as music, particularly in a such context.

Context can be used to detect the suitable type of media to be presented. Each user would have a list of media categories to create a recommendation list. Figure 3.2 shows the basic schema proposed for context-based recommendations for both explicit and implicit recommendation requests. The process starts by acquiring all of the available contextual information within the user's environment. The recommender system then analyzes the collected contextual data and matches it with the history of the user's contexts. Next, if the user has not specified, as an input, the type of media contents he/she wants, the system predicts the appropriate category based on the history of the detected context. After identifying the user's context and the type of media required, the user's profile, and the profiles of their friends on social networks are passed to the recommendation engine to generate the recommendation list.

3.2 User Physiological Context

One of the main contributions of this thesis is to build a context-aware recommendation model that also considers the physiological status of the user as a useful context parameter, in order to enhance the recommendation results. Physiological context is not only useful in the recommendation model, but also to trigger an event. For example, assume collecting biomedical data such as an ECG signal triggers a high level of stress in a user, an event can be triggered to suggest a suitable multimedia content to the situation. Accordingly, the system runs the recommendation process to generate a list of suggested tracks or albums. Therefore, we are interested in recording and monitoring the user's biological signals to determine certain physiological or emotional (e.g. mental stress) user condition.

3.2.1 Detecting a User's Physiological Context

We collect biological signals, but we are not concerned with diagnosing illnesses. Our goal is to use the physiological data collected to better understand the physiological status of users. In fact, in this thesis we focus on one particular emotion: mental stress. The best tool to detect the targeted physiological conditions is to monitor the heart using an ECG signal that can measure the Heart Rate Variability (HRV) [70].

Some of the least obtrusive commercial ECG devices are in the form of a chest strap fitted with two electrodes and an electronic circuit. This requires the user to wear the ECG sensor at all times in order for the application to benefit from physiological or psychological contextual information. Since these devices are wearable, they might be associated with a certain level of discomfort over prolonged periods of use. In this thesis, we used a non-invasive device, as shown in Figure 3.3. The ECG sensor used is a product from the AliveCor Company, and is connected to a Samsung Galaxy III smartphone. By having a sensor that looks like a protection cover attached to the smart device, an installed android application can collect the ECG signal automatically, without bothering the user. We use the ECG signal to extract the HRV information; since the ECG signal represents the electrical activity of the heart, we can extract the HRV signal by calculating the series of time intervals between two consecutive heart beats or R-waves [76]. The HRV is known to shed light on mental stress and infer some conclusions about the conditions of monitored individuals [28, 33, 36, 11].

To detect mental stress, the latest measurement collected is compared to a previously recorded benchmark. The benchmarks are typically previous measurements taken

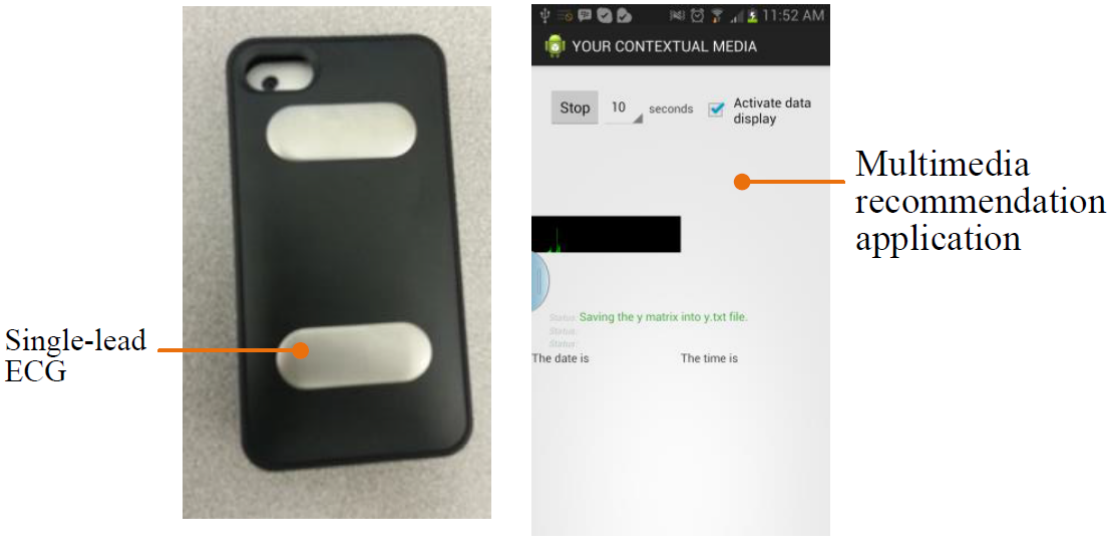


Figure 3.3: An image of the single-lead ECG sensor attached to an android device in the prototype application we developed.

during a neutral stress state. The length of the measurement is a key constraint here, since the longer the measurement is, the more reliable the conclusions are. Therefore, it is widely recommended to use ECG measurement records of at least three minutes [33]. Accordingly, the stress detection algorithm assesses the recorded ECG signal and compares the HRV calculated values to the benchmark measurement. Nonetheless, since the reason for measuring the physiological parameters in this study is to recommend suitable multimedia content, such a long measurement reduces the system's usability and applicability. In other words, asking a user to have a three minute ECG measurement before the system can recommend a song is unrealistic. In order to increase the application's practicability, we therefore decided to reduce the ECG recording time needed to evaluate the physiological parameters. The use of shorter ECG time measurements when analyzing mental stress is still a relatively new concept. Therefore, more research is needed in order to assess the accuracy of such an approach compared to longer and more accurate measurement methods [86]. We are willing to potentially sacrifice some accuracy in order to increase usability. For the physiological context detection, we simply use the short-term analysis of the heart rate variability proposed by [86] to determine if the user is in an elevated stress state, a neutral stress state, or a relaxed stress state, as in Figure 3.4.

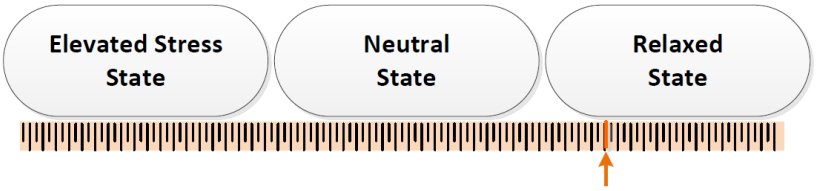


Figure 3.4: Three different physiological states are proposed as possible user physiological contexts.

After collecting the required biosignal, the physiological information extracted is sent, along with other contextual parameters, to the recommender system. Then, the recommender system recommends content such as music and movies, taking into consideration the detected physiological state of the user.

3.2.2 Experiment

We attempted to determine how the short ECG measurement is able to distinguish between the three main physiological contexts: elevated stress, neutral, and relaxed, as in Figure 3.5. We invited subjects to measure their HRV parameters while performing three types of activities. The sessions were composed of stressful, neutral, and relaxing exercises. For the stressful exercise, subjects were asked to perform a Stroop color-word test. The Stroop test has been used in different physiological and psychological studies [97, 74, 58], and involves sympatho-adrenal activation that is reflected in the subject's heart and respiration rates [97]. For the neutral exercise, subjects were asked to sit comfortably and try to read an article. We will use this exercise as a benchmark to analyze stressful and relaxing situations. For the relaxing exercise, subjects were asked to sit comfortably, close their eyes, and listen to relaxing music. Each exercise lasted three minutes and contained 3 minutes worth of ECG data. However, only the last 30 seconds were fed to the physiological detection algorithm in order to evaluate the short ECG measurement performance.

Ten adult subjects participated in the physiological context detection experiment: 6 males and 4 females. The average age was 28.7 years. The experiment was conducted in an office space in our laboratory. An office desk, a laptop, and headphones were provided for each participant. Subjects were seated in front of the laptop while the ECG sensor sent the data recorded to a Java based computer program.

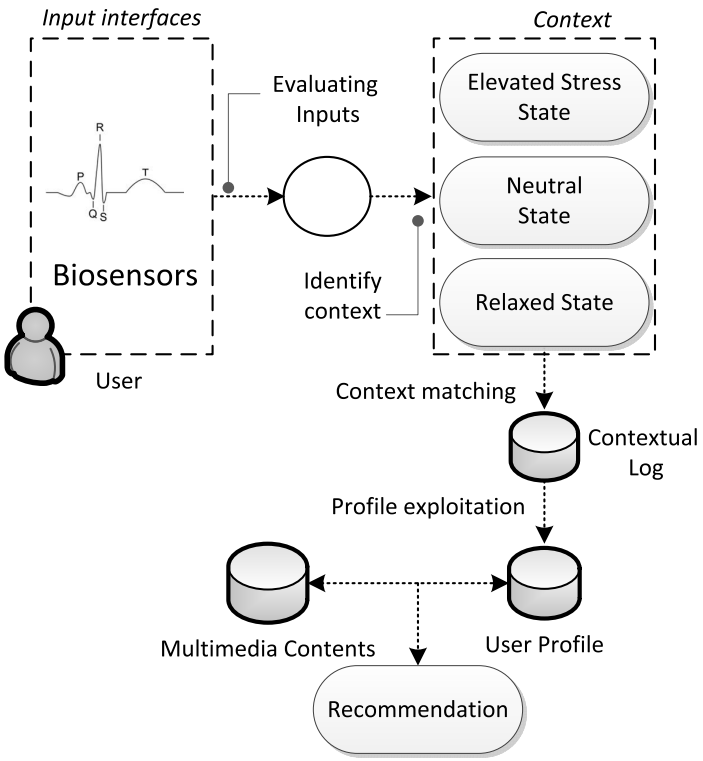


Figure 3.5: Recommendation based on the user's physiological conditions.

After collecting the ECG signal, three HRV parameters of interest were extracted to measure the level of mental stress:

- Low frequency band of the HRV (LF).
- High frequency band of the HRV (HF).
- The ratio of low frequency to high frequency HRV (LF/HF).

Numerous related studies have shown the effectiveness of these HRV parameters to measure mental stress [86]. Specifically, it was noticed that during a stressful situation, the HF component decreases while the LF to HF ratio increases. The resulting HRV parameters for the 10 subjects, during the last 30 seconds of the three experimental exercises, are presented in Table 3.2.

Subject Number	Neutral Exercise Session (**N**)			Elevated Stress Exercise Session (**S**)			Relaxed Exercise Session (**R**)		
	LF(**N**)[1]	HF(**N**)[1]	LF/HF(**N**)	LF(**S**)[1]	HF(**S**)[1]	LF/HF(**S**)	LF(**R**)[1]	HF(**R**)[1]	LF/HF(**R**)
1	2179.69	605.20	3.60	2884.29	566.49	5.09	1421.07	906.34	1.57
2	5457.56	4715.32	1.16	2871.76	2041.03	1.41	2755.02	6388.11	0.43
3	2548.07	3136.71	0.81	2467.03	1443.93	1.71	2976.48	4119.28	0.72
4	1648.41	922.55	1.79	1939.05	864.21	2.24	1399.98	563.22	2.49
5	4972.69	964.01	5.16	3898.86	863.92	4.51	1579.52	1128.10	1.40
6	1646.78	267.89	6.15	2468.13	251.07	9.83	1340.22	263.30	5.09
7	6518.59	2158.91	3.02	12489.29	1642.77	7.60	1792.80	1577.55	1.14
8	25890.21	150127.87	0.17	49184.93	108881.39	0.45	66522.72	142255.49	0.47
9	5306.72	2970.81	1.79	11629.48	4229.37	2.75	9693.36	7317.30	1.32
10	3164.43	1854.81	1.71	4436.91	1791.26	2.48	5809.44	3210.19	1.81
Average	5933.32	16772.41	2.53	9426.97	12257.54	3.81	9529.06	16772.89	1.64
STDEV	7227.36	46876.27	1.93	14487.92	33968.30	2.97	20201.45	44158.70	1.36

[1]Values are expressed in milliseconds (ms^2)

Table 3.2: The results of detecting the physiological context of the users.

Subject Number	* (0: relaxed, 10: stressed)		
	Neutral Exercise Session (**N**)	Elevated Stress Exercise Session (**S**)	Relaxed Exercise Session (**R**)
1	4.00	7	2.00
2	6.00	9	3.00
3	3.50	6.5	3.00
4	4.00	7	0.00
5	6.00	9	1.00
6	5.00	9	2.00
7	4.00	7	2.00
8	4.00	6	3.00
9	2.50	8	0.50
10	3.00	9	0.00
Average	4.20*	7.75*	1.65*

Table 3.3: The subjective results of the user experience during the experiment.

The results obtained have shown, on average, an expected increase in the LF/HF component, triggered by a stressor activity (with respect to the neutral state). Such an increase in LF/HF was also observed in earlier documented studies using a similar Stroop test [99]. In addition, for the same stressor activity, a decrease in the HF component (with respect to the neutral state) was noticed for the average experiment results, for the 10 subjects. Such a decrease has also been confirmed in [36, 20]. On the other hand, for the relaxation activity, we noticed, on average, an expected decrease in the LF/HF component with respect to the neutral state. Nonetheless, we did not find a significant change in the HF component with respect to the neutral state. Therefore, we will use the LF/HF exclusively to differentiate between the various physiological states (elevated stress, neutral, or relaxed).

After the experiments, subjects were asked to evaluate the three exercises subjectively by giving them a rating value that indicates how stressful each exercise was. A rating range from 0 to 10 was used, with 10 indicating the exercise was stressful, and 0 indicating that the exercise was relaxing. Our experimental design of three different exercises that distinguish between the required physiological statuses (elevated stress, neutral, and relaxed) has been reflected in the results reported in Table 3.3. Accordingly, the HRV analysis results verify the correlation between the physiological status of the subjects

performing the three experimental exercises and the subjective assessment observed.

3.3 Summary

With today's technology, most cellphones are equipped with sensors such as GPS and wireless connections. They often carry applications that can identify some contextual information such as the time, the date, and surrounding points of interest (e.g. movie theaters). Therefore, acquiring the contextual information surrounding the user when he/she is making their request to the system has never been easier. The AmI environment also allow us to read the condition of the user's environment using numerous sensors, make reasonable judgments about the user context, and adopt several mechanisms to modify the surrounding environment using actuation interfaces. A hand-held ECG sensor can also be used to provide the system with the ECG signal information, which gives an electrical indication of the heart activity. In this thesis, measuring the stress level is one of the objectives of recording the ECG signal. The stress level is an important pattern that can be extracted from ECG signals by measuring the HRV [46].

In this Chapter, we demonstrate how to integrate the media contents with the available user's physiological and environmental parameters to enhance the user experience and comfort level. By including a user's biological signal and leveraging collaborative filtering, we can build a context-aware model that establishes the bridge between the multimedia content and the user's context, which contains certain physiological parameters. In Chapter 4, we analyze the use of the detected user's context and explore the use of contextual tags into a recommendation model to further enhance the recommendation process.

Chapter 4

Context-Aware Recommendations

With the rapid increase of social media in real-time, context-aware recommendations offer the potential of exploiting social tags and rating information in ways that personalize the search for content in a particular detected context. They tackle the problem of identifying the proper resources from the vast number of choices available online. In this chapter, we demonstrate a new recommendation model that personalizes the recommendations and contributes in increasing the quality of the user experience by analyzing their context at the time when they wish to select a multimedia content. We explain how to leverage hidden preferences to rate items within an adapted collaborative filtering algorithm under a given context.

4.1 Preliminaries

Before describing the details of the recommendation model, we must introduce a set of definitions to formalize our recommendation problem.

4.1.1 Notation

In this chapter, we applied a common set of labeling conventions for graphical letters. The bold upper-case letters, such as $\mathbf{A}$, are used to denote matrices; whereas the corresponding lower-case italic with two subscript indices, such as $a_{x,y}$ represent the entries of the matrices. Capital italicized letters represent sets, such as U, and an upper-case italic letter with one subscript index such as U_x represents an entry element x from set U. We also formalize the matrix elements in the recommendation model to follow the

Notations	Meaning
U	Sets of users.
I	Sets of items.
C	Sets of contexts.
$\mathbf{A}_{\|C\|\times\|I\|}$	Context-item matrix.
$\mathbf{T}_{\|C\|\times\|U\|}$	Context-user matrix.
$\mathbf{B}_{\|U\|\times\|C\|}$	User-context matrix.
$\mathbf{R}_{\|U\|\times\|I\|}$	User-item matrix.
$\mathbf{W}_{\|I\|\times\|U\|}$	Item-user matrix.
$\mathbf{S}_{\|U\|\times\|U\|}$	User-user similarity matrix.
$\mathbf{E}_{\|I\|\times\|I\|}$	Item-item similarity matrix.
$\mathbf{Q}_{\|C\|\times\|C\|}$	Context-context similarity matrix.
$\mathbf{UtC}_{\|U\|\times\|C\|}$	Hidden preferences of users toward contexts.
$\mathbf{CtI}_{\|C\|\times\|I\|}$	Hidden preferences of contexts toward items.
$\mathbf{UtI}_{\|U\|\times\|I\|}$	Hidden preferences of users toward items.
$\mathbf{ItU}_{\|I\|\times\|U\|}$	Hidden preferences of items toward users.
$\mathbf{S}^k$	Contains only k most similar users.
$\mathbf{E}^k$	Contains only k most similar items.
$\bar{\mathbf{T}}$	Normalized matrix $\mathbf{T}$.
$\bar{\mathbf{E}}$	Normalized matrix $\mathbf{E}$.
x, y, z	Row and column matrix indexes.

Table 4.1: Summary of notations and their meaning.

following format: $\mathbf{Matrix} = [(element\,entry)_{SetA,SetB}]_{|A|\times|B|}$. For example, the context-item matrix in the model is represented in the following format: $\mathbf{A} = [a_{c,i}]_{|C|\times|I|}$, which denotes a matrix $\mathbf{A}$ is built on rows and columns from sets C and I respectively. The lower case letter subscripts, for example C_x, represents an entry x from the set C. The two subscripts lower-case letter $a_{c,i}$ represents a matrix entry that resides on row c and column i in matrix $\mathbf{A}$. In addition, to simplify the recommendation problem, we refer to the term "users" or (U) to represent the set of users, and "items" or (I) to denote a set of media resources that can be recommended. Table 4.1 summarizes the notations employed in the rest of this thesis.

4.1.2 Problem Definition

Suppose we have a group of users who share music content. A user's information, including their history of rating and their textual annotations for certain tracks, describing their experience as "love", "relaxing", "gym", etc., are available. We also have another group of users who have not yet entered any annotations toward any music or who have not shared their ratings publicly on the Internet. We want to personalize the recommendation for both types of users. We have first to identify different contextual dimensions and then deliver media content that best fits the user's detected context.

For a set of users, items, and contexts corresponding to our context recommendation model, the users set is denoted by $U = \{u_1, u_2, ..., u_{|U|}\}$, the items set by $I = \{i_1, i_2, ..., i_{|I|}\}$, and the contexts set by $C = \{c_1, c_2, ..., c_{|C|}\}$, a recommendation model that predicts the suitability or interest of user (u) for item (i), given a context (c) can be built. Then, the context attributes are used in the recommendation process to filter and uncover items that are probably of interest to the user in such a context, personalized according to the user's preferences. The first list U represents users who can select an item in set I under different contexts C. Given contextual tags associated with a user (u) interacting with items (i), we want to recommend new items to user based on their context. Accordingly, we can obtain the three dimensions forming our recommendation problem into a frequency tensor model with $|U| \times |I| \times |C|$ entries. Note that items or I refers to any resources that an application can recommend such as movies, music, or news. We decomposed the identified features of the tensor model into three different matrices: user-item, context-item, and context-user matrices. Additional supported matrices are also built to discover the similarities between each individual dimension.

4.1.3 Multimedia Rating

Media rating is an indication that shows the interest or preference of a certain user for a particular item. An example of such a rating is a score of five stars in a movie-rating database, where that score indicates that the user has really enjoyed watching the movie (positive feedback). Even though such information can be used to represent the user's interest or preference for certain types of applications, it may not always be accurate. A user might like a particular song and give it a very high score, but he/she may not be interested in listening to it on a daily basis. Therefore, we believe that a rating should be adapted and dynamically modified based on the user's behavior and context in general. This type of rating is called explicit rating, where a user explicitly provides the system

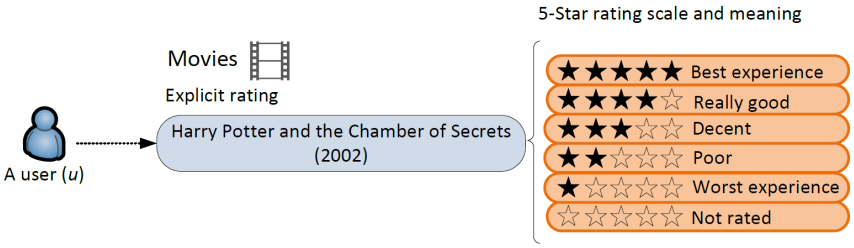


Figure 4.1: An example of an explicit rating using a 5-star scale rating system.

with input values that indicate their interest in a particular item or collection, as shown in Figure 4.1. There is another type of rating called implicit rating, where the system implicitly predicts the user's interest by monitoring the user's behavior while they are playing or consuming a multimedia content.

4.2 Exploring the Three-dimensional Space

Since the relationship between a user u_x and item i_y can exist within a context c_z, the three-dimensional space on users, items, and contexts can be reduced to a two-dimensional space represented by three two-dimensional matrices. As stated in the problem formulation section, the information we gather using the prototype application helps us shape the relationship between users and contexts, by building three main matrices to build the base for our recommendation model as follows: user-context matrix $\mathbf{B}_{|U|\times|C|}$, user-item matrix $\mathbf{R}_{|U|\times|I|}$, and context-item matrix $\mathbf{A}_{|C|\times|I|}$.

To build the required matrices needed for our proposed models, we use the example illustrated in Figure 4.2. As shown in Figure 4.2, we have a list of users $U = \{u_1, u_2, ..., u_4\}$, a list of items $I = \{i_1, i_2, ..., i_8\}$, and a list of contexts $C = \{c_1, c_2, ..., c_5\}$. Figure 4.3 shows that the three-dimensional space decoupled as three two-dimensional matrices user-context matrix $\mathbf{B}_{|U|\times|C|}$, user-item matrix $\mathbf{R}_{|U|\times|I|}$, and context-item matrix $\mathbf{A}_{|C|\times|I|}$. The user-context matrix is equal to $\mathbf{B} = [b_{u,c}]_{|U|\times|C|}$, where $b_{u,c}$ represents the number of items consumed by user u_x in context c_z. Similarly, to build the matrix $\mathbf{T}_{|C|\times|U|}$, let $t(c_x, u_y)$ be the number of times a user u_y consumed items in context c_x. The user-item matrix, $\mathbf{R} = [r_{u,i}]_{|U|\times|I|}$ represents the ratings of users u_x to item i_y. The context-item matrix is equal to $\mathbf{A} = [a_{c,i}]_{|C|\times|I|}$, where $a_{c,i}$ represents the number of users who consumed item i_y in context c_z. The matrix $\mathbf{A}_{|C|\times|I|}$ can also represent the number of times item i_y was consumed in context c_x.

In case a user has not consumed any items in a given context, or if an item has never

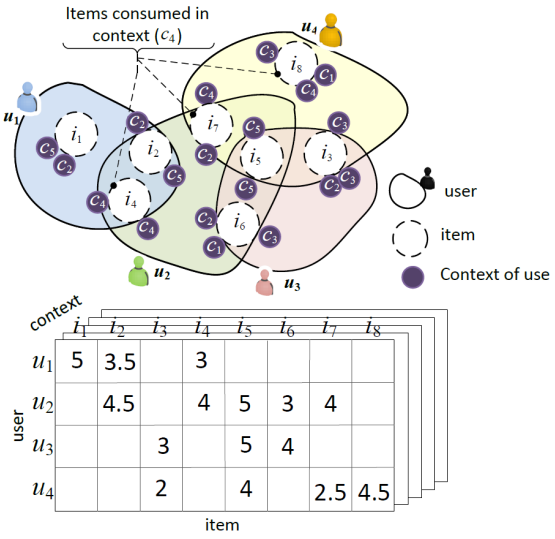


Figure 4.2: A representation of the three dimensions of the recommendation problem.

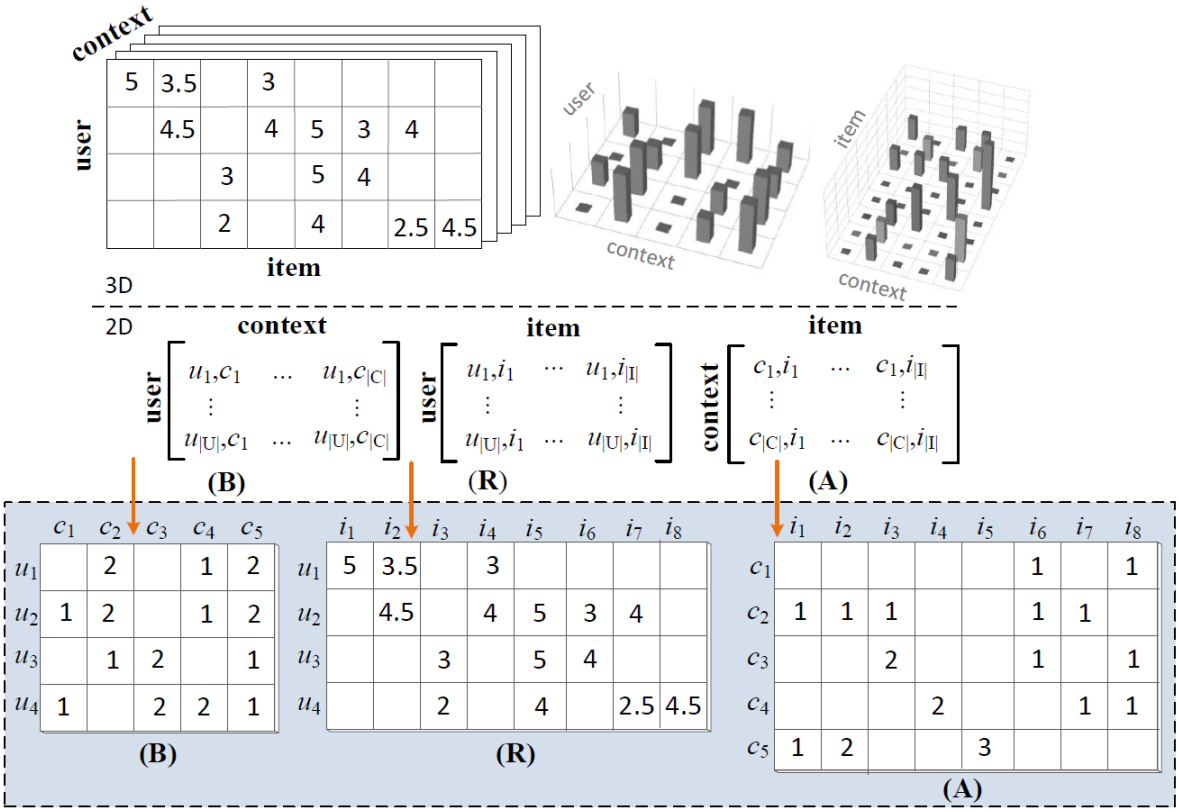


Figure 4.3: A representation of the three-dimensional space by three two-dimensional spaces.

been consumed by any user in a particular context, then the $t(c_x, u_y)$ and $a(c_x, i_y) = 0$ respectively. In addition, if we only consider the frequency of usage for a particular context within the users or items scope, then the accuracy of the recommendation results might be affected by the number of users who repeatedly use items in a large variety of contexts. Consequently, we would neglect the importance of how many users have consumed items within that context, because of a small number of users who consumed many items in a particular context. We therefore normalized the frequency values in a range between 0 and 1 by using the following formulas:

$$t(c_x, u_y) = \frac{n_{c,u}(c_x, u_y)}{N_{c_x,u}} \tag{4.1}$$

$$a(c_x, i_y) = \frac{n_{c,i}(c_x, i_y)}{N_{c_x,i}} \tag{4.2}$$

where $n_{c,u}(c_x, u_y)$ is the number of occurrences of context c_x in the list of consumed items by u_y, $n_{c,i}(c_x, i_y)$ is the number of occurrences of context c_x in the list of contexts an item has been consumed in $f_{x,y}$ as in Equations 4.1 and 4.2. $N_{(c_x,u)}$ and $N_{(c_x,i)}$ represent the number of times the context c_x is used with all items, and the number of times the context c_x is used by all users, respectively, according to Equations 4.3 and 4.4.

$$\begin{aligned} N_{c_x,u} &= \sqrt{\textstyle\sum_{y=1}^{|U|} \left(\delta_{x,y} f_{x,y}\right)^2} \\ \delta_{x,y} &= \begin{cases} 1 & c_x \, occured \, in \, u_y \\ 0 & otherwise \end{cases} \end{aligned} \tag{4.3}$$

$$\begin{aligned} N_{c_x,i} &= \sqrt{\textstyle\sum_{y=1}^{|I|} \left(\delta_{x,y} f_{x,y}\right)^2} \\ \delta_{x,y} &= \begin{cases} 1 & c_x \, occured \, in \, i_y \\ 0 & otherwise \end{cases} \end{aligned} \tag{4.4}$$

It is to be noted that, we also examined the usage of a binary version of values for **T** and **A**, but in this case, we would not be able to show how often a particular item is being used in a specific context. That is because in the binary case, we would only be able to determine whether or not an item was consumed in that specific context.

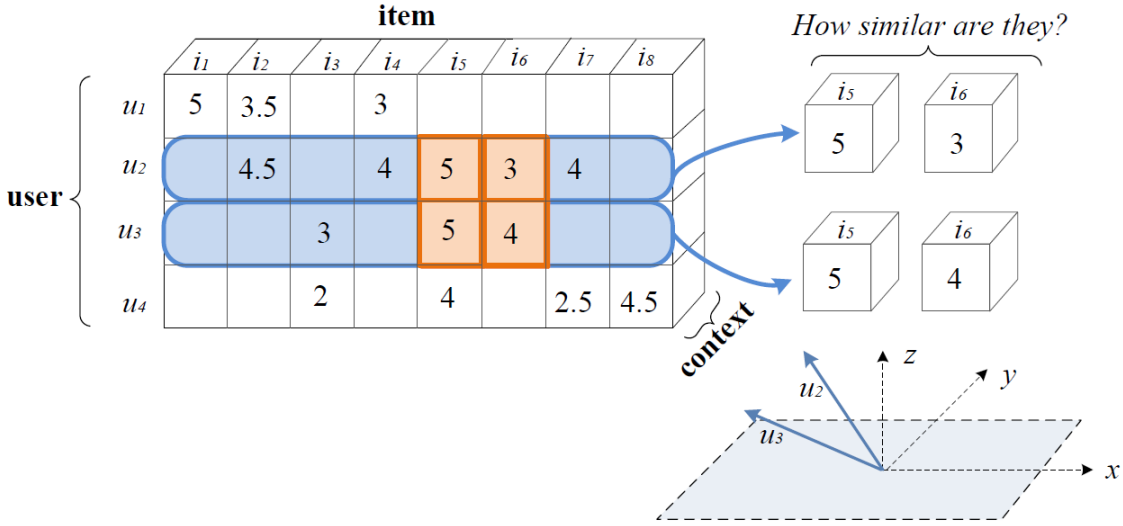


Figure 4.4: Example of computing the user-user similarity

4.3 Dimension-Based Collaborative Filtering

In order to find the hidden preferences in our model, we first calculate three types of similarities for each individual dimension: user-user, item-item, and context-context. The similarity measure can be computed from a set of decomposed matrices of the original three dimensional tensor (see Figure 4.4 and Figure 4.5). To compute the similarities between users $\mathbf{S}_{|U|\times|U|}$, we first decompose the model in Figure 4.2 to construct the user-item matrix $\mathbf{R}_{|U|\times|I|}$. Matrix $\mathbf{R}$ can be created from a boolean function, where each true value in the matrix represents the fact that a user u consumed item i regardless of their context. There is also an option of building the matrix $\mathbf{R}$ from the rating values assigned to items by users. In this thesis , we examine the use of both of these approaches. Likewise, we can compute the similarity of items $\mathbf{E}_{|I|\times|I|}$ from the same matrix $\mathbf{R}_{|U|\times|I|}$. As for finding the similarities of contexts, we create the decomposed matrix $\mathbf{A}_{|C|\times|I|}$, representing the frequency of items consumed in a particular context. The similarity of contexts shows the semantic relations of two contexts by finding how frequent an item appears in each context. Thus, the contextual description that is frequently used with an item can help depict its theme. It is the same concept of social tagging that is indicated in previous studies [12, 96, 18]. Note that the similarity of contexts is computed in terms of items rather than users, since the behavior of users toward contexts is not reliable.

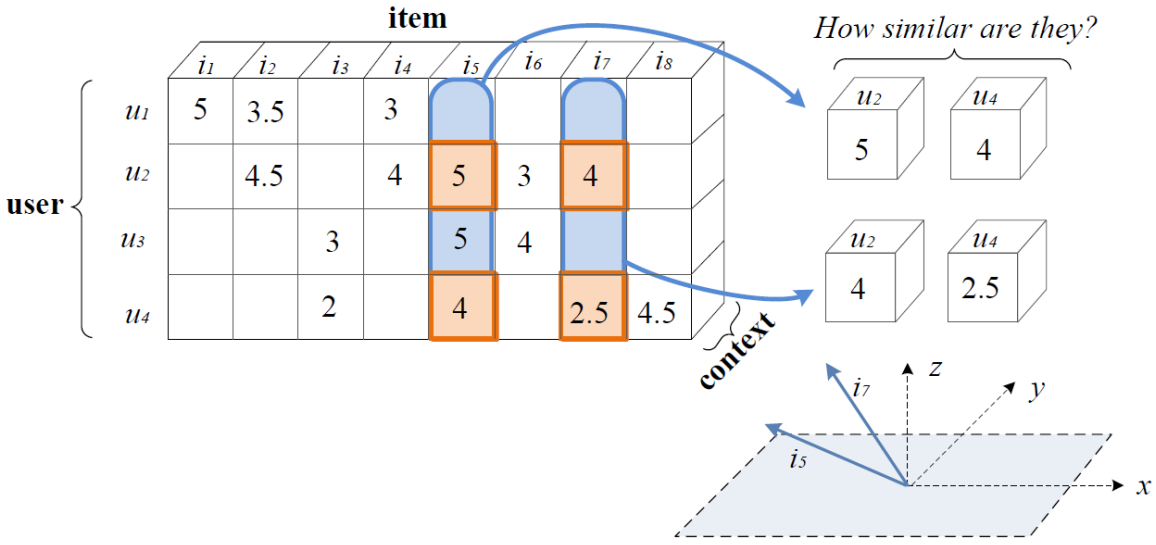


Figure 4.5: Example of computing the item-item similarity

There are various similarity measures that can be used to determine the similarity of any two vectors v_x, and v_y. For instance, cosine similarity [24], Jaccard and Overlap [71], Pearson correlation coefficient [93], and others. The cosine similarity is the approach we used in our proposed recommendation model to quantify the similarities.

4.3.1 User-Based Collaborative Filtering

Before analyzing the user's context using a collaborative filtering technique [14], the proposed recommendation model identifies the user's neighbors. The concept behind relying on the detection of similar users who share certain items is to exploit the list of items consumed by given users to find other interesting items consumed by similar users (also called nearest neighbors). To determine the similarity between two users, we used the cosine-based similarity. The cosine-based similarity takes two vectors of shared items of users u_x, and u_y, and quantifies their similarity according to their angle, as in Equation 4.5.

$$s(u_x, u_y) = cos(u_x, u_y) = \frac{u_x.u_y}{||u_x||.||u_y||} \tag{4.5}$$

To minimize the computational cost, we consider the top k nearest neighbors for each

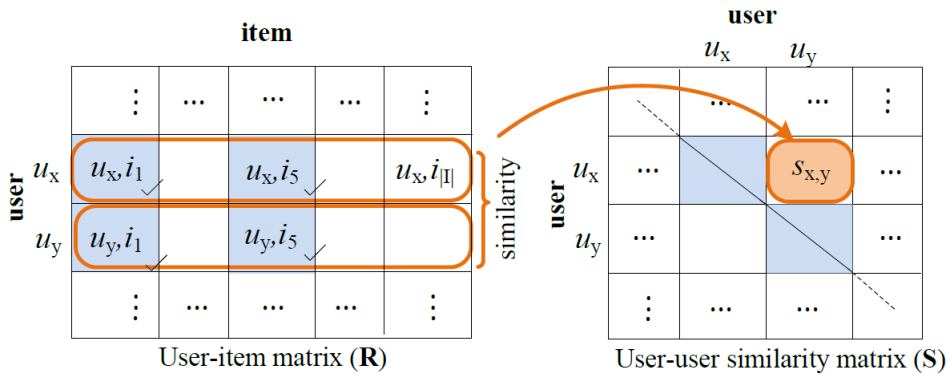


Figure 4.6: A representation of the user-user similarity matrix **S**.

user. Accordingly, we eliminate the computed similarities of those users who share few items with others, and assign a zero similarity value if the similar user is not among the top k nearest neighbors. We employ the matrix (**S**), where $\mathbf{S} = \mathbf{S}^k$, to form the user-user similarity matrix. Figure 4.6 shows an example of constructing the similarity matrix **S**.

4.3.2 Item-Based Collaborative Filtering

As we employ collaborative filtering to observe the user-user similarities, we employ the user-item matrix **R** to observe the item-item similarities. According to [24], the item similarity can be computed using collaborative filtering, which states that a user is likely to consume items that are similar to other items they have already consumed in the past. The similarity values can be obtained by measuring the cosine angle between the two column vectors in the matrix **R**, comparable to finding the user-user similarity with Equation 4.6.

$$e(i_x, i_y) = cos(i_x, i_y) = \frac{i_x . i_y}{||i_x||.||i_y||} \tag{4.6}$$

The similarities that are found allow us to form the item-item similarity matrix **E**. The similarity value of $e(i_x, i_y)$ is only considered if it is greater than the top k nearest item neighbors, otherwise the similarity value is set to zero. Figure 4.7 shows an illustrated example of computing the item-item similarity.

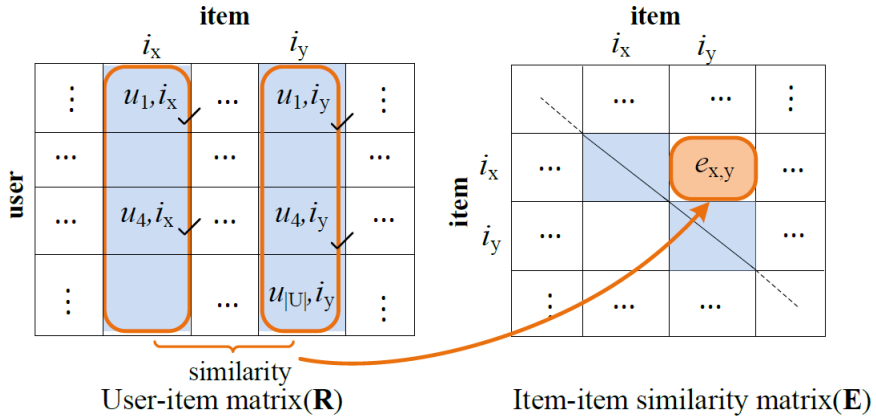


Figure 4.7: A representation of the item-item similarity matrix **E**.

4.3.3 Context-Based Collaborative Filtering

To compute the context-context similarity matrix $\mathbf{Q}_{|C|\times|C|}$, we start by utilizing the context-item matrix **A**. Using a context-based collaborative algorithm, k-nearest neighbors can be derived in any given context. Hence, the resulting matrix **Q**, obtained from the collaborative filtering, holds the similarity entries $\mathbf{Q} = (q_{c_x,c_y})$, where c_x and c_y represent the row and column of context c_x and context c_y, as shown in Figure 4.8. The similarity between each pairs of contexts is calculated using the cosine similarity, according to Equation 4.7:

$$q_{c_x,c_y} = cos(c_x, c_y) = \left(\frac{c_x.c_y}{||c_x||.||c_y||}\right) \tag{4.7}$$

The similarity is considered only if the similarity value is among the top k similarities in a given column; otherwise the similarity value is set to 0. As a result, the matrix **Q** contains $\mathbf{Q}^k$ similar contexts, where k is the number of the top context-context similarities.

4.4 Hidden Preferences Explorer Model (HPEM)

The inceptive idea of the proposed HPEM model is that users who consume certain items in particular contexts are likely to consume similar items in similar contexts. The main technical issue here is identifying the hidden features of contexts for both users and items. To solve this issue, we analyze the contextual information associated with the interactions of $< u, i >$ in the dataset, in order to identify the hidden preferences of

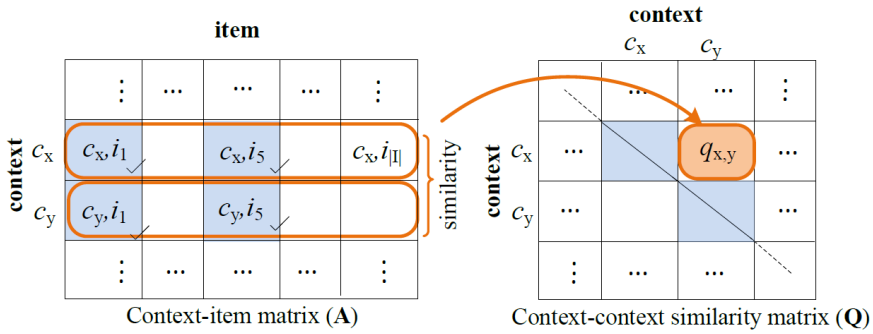


Figure 4.8: A representation of the context-context similarity matrix **Q**.

each dimension. In fact, we trace the patterns of the contextual consumption to fill the gap between users and new items, as well as between items and new contexts. There are items in I for users in U under context C, where the user's preferences are unknown. However, we can build three hidden models that represent the hidden preferences of users toward contexts $\mathbf{UtC}_{|U|\times|C|}$, the hidden preferences of contexts toward item $\mathbf{CtI}_{|C|\times|I|}$, and the users' preferences of items $\mathbf{UtI}_{|U|\times|I|}$.

4.4.1 Searching for Hidden Preferences

If we look at the relationship between users and items for different attributes of contexts, we notice that there are hidden causes for which an item is consumed in a certain context, as well as hidden reasons why users prefer to consume certain items in a given context. Finding such reasons involves an analysis of human behavior and psychological response. However, by using the similarity matrices created in Section 4.3, we can build a model that reflects the hidden preferences of a given user to a context, and of a given item to a context. Inspired by the model proposed by Kim et al. [51], the construction of the three matrices (**T**, **R**, and **A**) in Section 4.2 leads to the discovery of the hidden association of items toward a particular context, and the hidden association of users toward contexts, and accordingly, leverage relevant items for a user in a particular context.

Searching for Hidden Context-Item Preferences

Before recommending items to a user, we need to find the hidden preferences of that user toward their current context; this can be done by analyzing the hidden preferences of users toward items in a given context. By finding the hidden context-item preferences, we

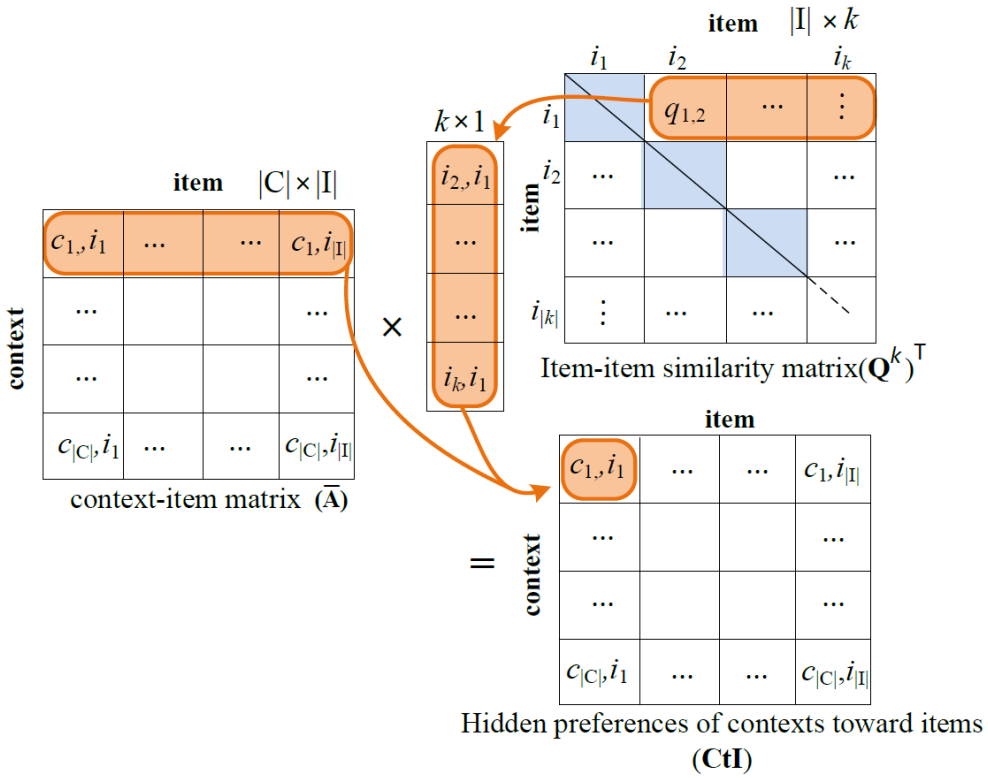


Figure 4.9: An illustration of the process of computing the hidden context-item matrix **CtI**.

capture how a particular context has occurred with the user's selection of items that are similar to a given particular item. We utilize the matrix **A** and the transpose of matrix **E**, which we constructed earlier to form the new context-item matrix **CtI**. Formally, the matrix **CtI** represents the matrix multiplication results of both **A** and **E**, as in Equation 4.8:

$$\mathbf{CtI} = \bar{\mathbf{A}}(\mathbf{E}^k)^T \tag{4.8}$$

Where the matrix $\bar{\mathbf{A}}$ denotes a normalized version of the matrix **A**, and the matrix $(\mathbf{E}^k)^T$ denotes the transpose of the top k nearest items as explained in Section 4.3.2. The multiplication of the c-th row by the i-th column implies finding the hidden preferences of context c, on item i with respect to the items' k nearest neighbor. Figure 4.9 shows the details of the construction of the new matrix **CtI**.

The reason for normalized values in matrix **A** is to reduce the effect of items that were consumed by many users (first type), than other less consumed items (second type).

Hence, the first type of items contributes more in estimating the context-based prediction value than the second type of items. Using normalization, we can minimize the effects of those items in regards to the detected contexts.

Searching for Hidden Context-User Preferences

a. Exploring the user dimension: Users in a specific context are likely to consume items that are similar to their preferences or to the preferences of similar users. In order to find the similarity between two users in the user-user matrix $\mathbf{S}$, we used the cosine vector similarity, using only the top k nearest neighbors for each user. According to Equation 4.9, we can derive the context-user matrix $\mathbf{CtU}$ from the product of the two matrices, $\bar{\mathbf{T}}$ and $\mathbf{S}$, as seen in Figure 4.10. The matrix $\mathbf{CtU}$ represents the hidden context of a given user u_x, which shows how a particular context was consumed by users similar to user u_x.

$$\mathbf{CtU} = \bar{\mathbf{T}}(\mathbf{S}^k)^T \tag{4.9}$$

In this model, we also considered the issue caused by having certain active users that consume different items in a variety of contexts. Accordingly, active users contribute more in the production of the recommendation results than other less active users. Therefore, the matrix $\bar{\mathbf{T}}$ holds normalized columns for each user to reduce the contribution effects. In addition, matrix $\mathbf{S}$ is the transpose matrix of the production of the columns that represent users in the normalized matrix $\bar{\mathbf{T}}$.

b. Exploring the context dimension: As stated in Section 4.2, we can derive a user-context matrix $\mathbf{B}_{|U|\times|C|}$ by an aggregation task over C and I. Then, by decomposing the context-item matrix $\mathbf{A}_{|C|\times|I|}$, we build the context-context matrix $\mathbf{Q}_{|C|\times|C|}$ by computing the similarity of a pair of vectors to find items that were consumed in similar contexts. The similarity value here represents how frequent such a pair of contexts is used in consuming given items. By multiplying the two matrices $\mathbf{B}_{|U|\times|C|} \times \mathbf{Q}_{|C|\times|C|}$, we can obtain the hidden matrix $\mathbf{UtC}$.

Searching for Hidden Item-User Preferences

a. Exploring the user dimension: In this step, we capture the user's hidden preferences for an item. The main idea is that users in a context consume certain items, and that when they are in the same context in the future, they will likely consume items that

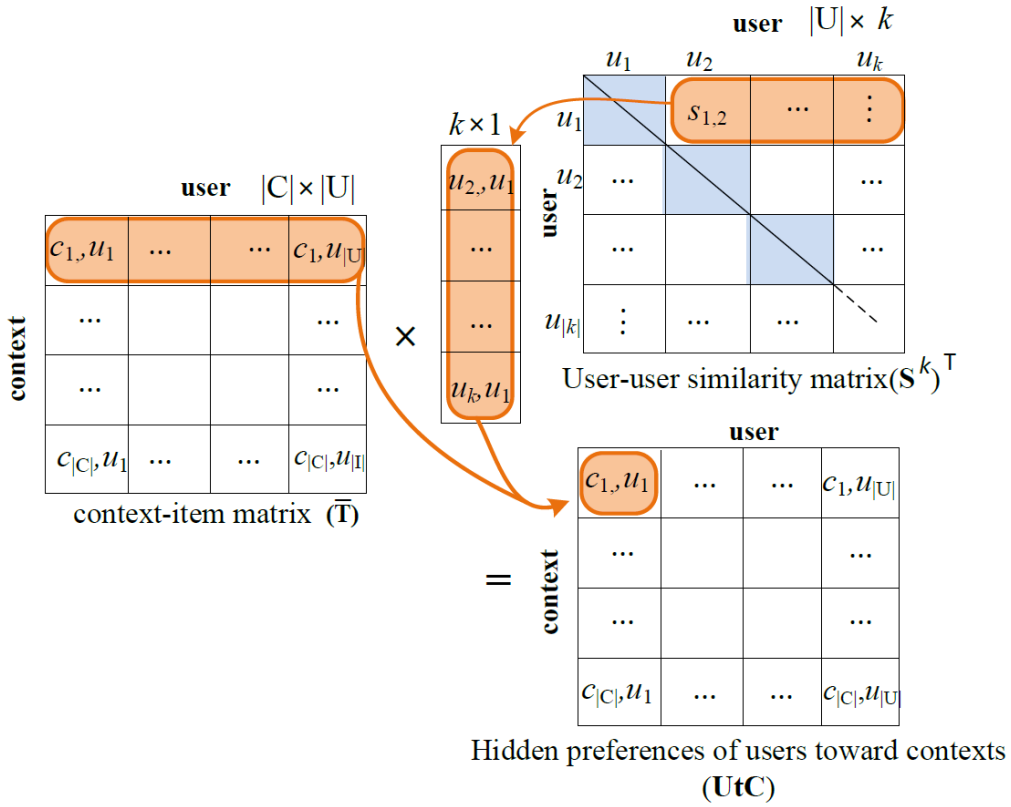


Figure 4.10: An illustration of the process of computing the hidden context-user matrix **UtC**.

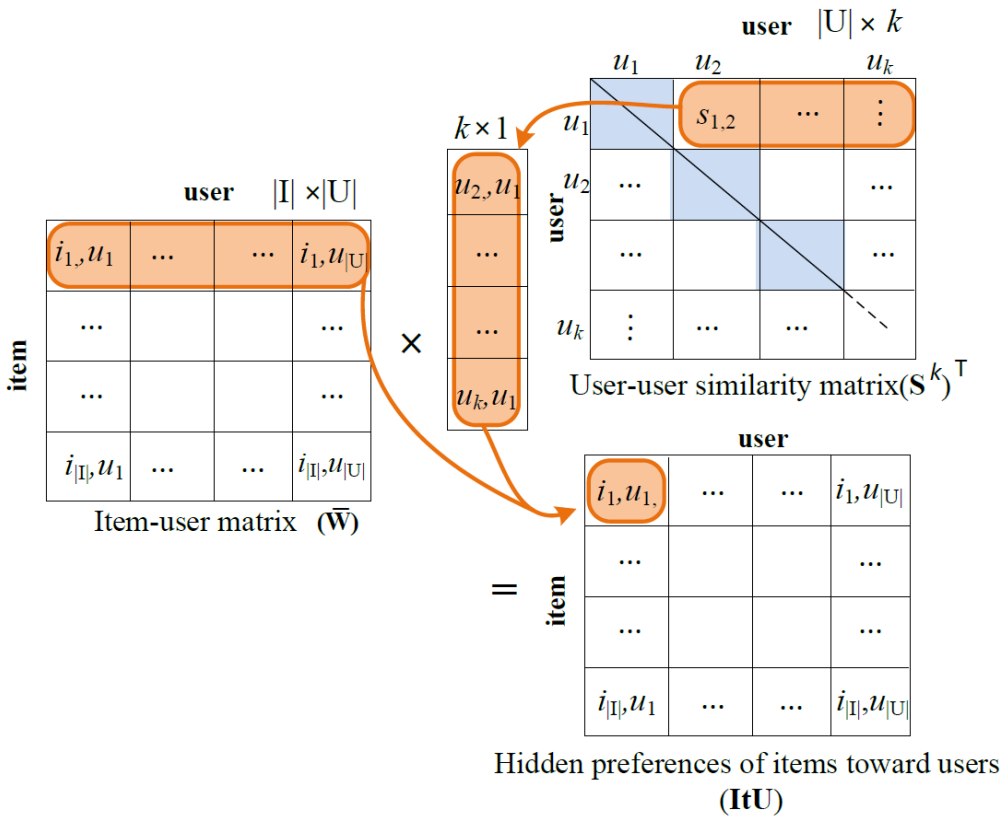


Figure 4.11: An illustration of the process of computing the item-user matrix **ItU**.

are either similar to their preferences or similar to the choice of their nearest neighbors. We denote matrix **ItU** to represent the hidden item preferences for a given user u_x, which also includes the item preferences of comparable users. We build the matrix **ItU** according to Equation 4.10 as follows:

$$\mathbf{ItU} = (\bar{\mathbf{R}})^T(\mathbf{S}^k)^T \tag{4.10}$$

Where $(\bar{\mathbf{R}})^T$ is the transpose of the original normalized rating matrix $\mathbf{R}$ (denoted as $\mathbf{W}$) and $(\mathbf{S}^k)^T$ is the top k user-user similarity matrix. The product of the two matrices $(\bar{\mathbf{R}})^T(\mathbf{S}^k)^T$ brings the user and their nearest neighbors' preferences for a given item.

At this point it is also important to consider the issue of having some users that are more active in rating and consuming different items than other users. Therefore, we normalize the values in matrix $\mathbf{W}$, before the multiplication step, in order to reduce such contributing effects. Figure 4.11 shows the details of constructing the new matrix **ItU**.

b. Exploring the item dimension: We can derive a user-item matrix $\mathbf{R}_{|U|\times|I|}$ aggregated over U and I. Likewise, we build the item-item matrix $\mathbf{E}_{|I|\times|I|}$ by computing the similarity between two items from the set I. In our experiments, we use cosine similarity, as explained in Section 4.3.2. By multiplying the two matrices $\mathbf{R}_{|U|\times|I|} \times \mathbf{E}_{|I|\times|I|}$, we can obtain the hidden matrix $\mathbf{UtI}_{|U|\times|I|}$.

4.4.2 Context-Boosted HPEM Model

a. Emphasis on the Context

We create two models to reflect the user-item relationship in a given context. The first model reproduces the answer to the following question:

- (1) How a certain user has previously consumed items in contexts that were similar to a given context?

The first model represents the hidden user-context preferences denoted as $\mathbf{UtC}_{|U|\times|C|}$, which can be derived by the product of the two matrices $\mathbf{B}_{|U|\times|C|}$ and $\mathbf{Q}_{|C|\times|C|}$ as follows:

$$\mathbf{UtC}(u,c) = \sum_{m=1}^{|C|} (\bar{\mathbf{B}})_{u,m} \times (\mathbf{Q})^T_{m,c} \tag{4.11}$$

Where $\bar{\mathbf{B}}$ contains the normalized values of $\mathbf{B}$, according to Equation 4.12, and $(\mathbf{Q}^k)^T$ is a transpose matrix of $\mathbf{Q}^k$. The entries in matrix $\bar{\mathbf{B}}$ represent the frequency function of the number of items a user consumed in a context. However, this function can be replaced with a Boolean or a rating function, depending on the application. Let $n_{u,c}$ denote the number of occurrences of context c_y in the list of consumed items by u_x, and N_{u,c_y} the number of items consumed in context c_y in the list of all users, as in Equation 4.3. This computation step reflects the model's concept of the hidden influence of the most similar contexts to a given user's context, in reproducing the preference estimation value. Figure 4.12 illustrates the calculation process of building the user-context model **UtC**.

$$b(u_x, c_y) = \frac{n_{u,c}(u_x, c_y)}{N_{u,c_y}} \tag{4.12}$$

The second model uncovers items to answer the following question:

- (2) How to find items that have been consumed in a certain context and are similar to a given item?

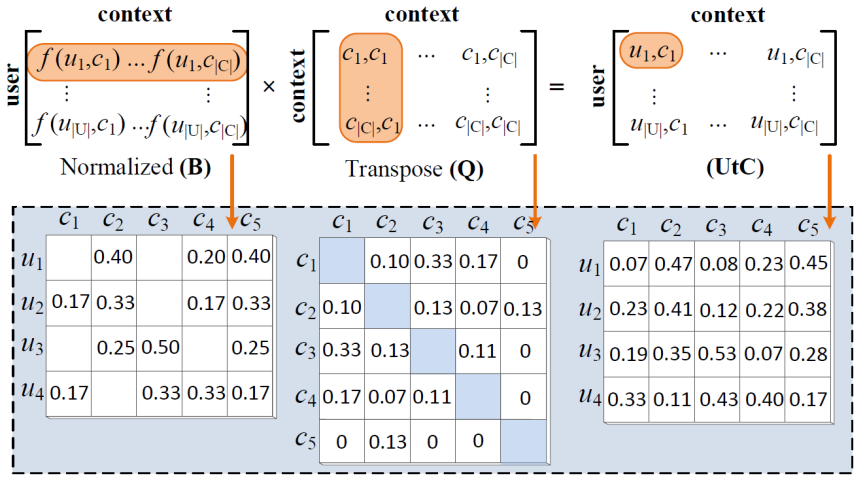


Figure 4.12: An illustration of the process of computing the user-context matrix **UtC**.

The second model can be derived by the product of the two matrices $\mathbf{A}_{|\mathbf{C}|\times|\mathbf{I}|}$ and $\mathbf{E}_{|\mathbf{I}|\times|\mathbf{I}|}$ built in Section 4.4.1, and Section 4.3.2 as follows:

$$\mathbf{CtI}(c,i) = \sum_{\mathbf{m=1}}^{|\mathbf{I}|} (\bar{\mathbf{A}})_{c,m} \times (\mathbf{E})^T_{m,i} \tag{4.13}$$

where $\bar{\mathbf{A}}$ contains the normalized values of $\mathbf{A}$, and $(\mathbf{E}^k)^T$ is a transpose matrix of $\mathbf{E}^k$. In this computation step items that are similar to a certain item contribute more to estimate the weight of that item. Figure 4.13 illustrates the calculation process of building the new context-item weight model **CtI**.

According to our proposed model, named the Context-boosted HPEM model, the final estimation of the user-item matrix is computed by the product of the two models $\mathbf{UtC}_{|U|\times|C|}$ and $\mathbf{CtI}_{|C|\times|I|}$. For a given set of contexts $C_m = \{c_1, c_2, ..., c_m\}, m \leq |C|$, the relevance value of item i_y for user u_x can be computed as:

$$Rate_{u,c}(i) = \sum_{c \in m} \mathbf{UtC}_{u,c} \times \mathbf{CtI}_{c,i} \tag{4.14}$$

where $\mathbf{UtC}_{u,c}$ is the entry value of $\mathbf{UtC}_{u,c}$ in the matrix **UtC**, and $\mathbf{CtI}_{c,i}$ is the entry value of $\mathbf{CtI}_{u,c}$ in the matrix **CtI**. By utilizing both models, **UtC** and **CtI**, items that are likely to fit a user's needs rate higher in the recommended list. Figure 4.14 illustrates the process of computing the context-boosted HPEM value for a given set of contexts.

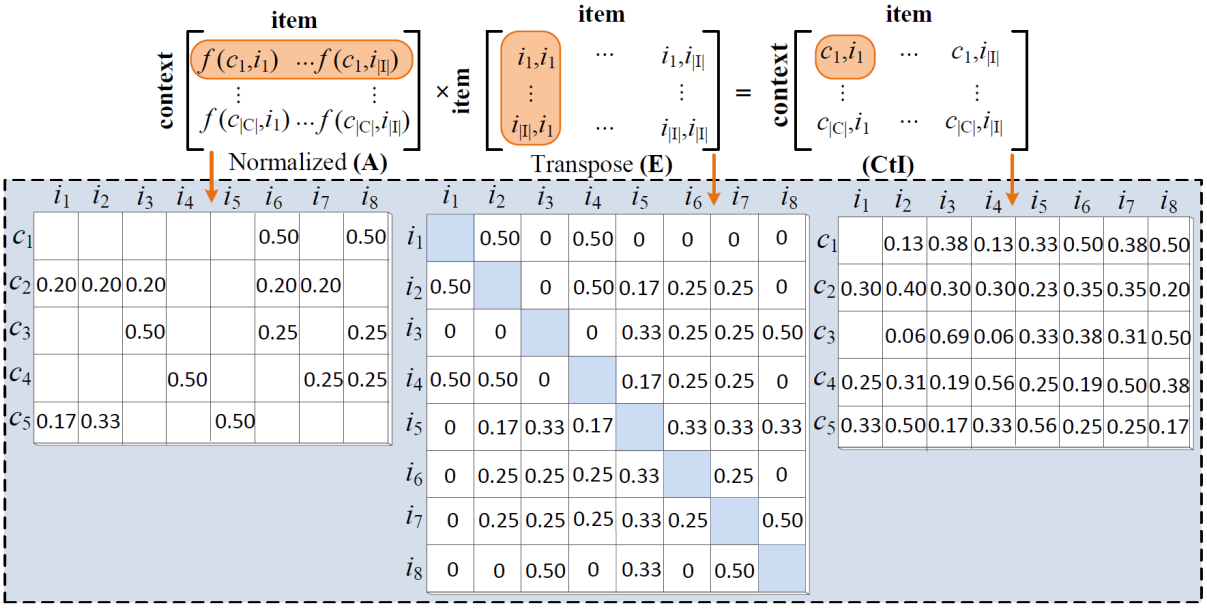


Figure 4.13: An illustration of the process of computing the context-item hidden matrix **CtI**.

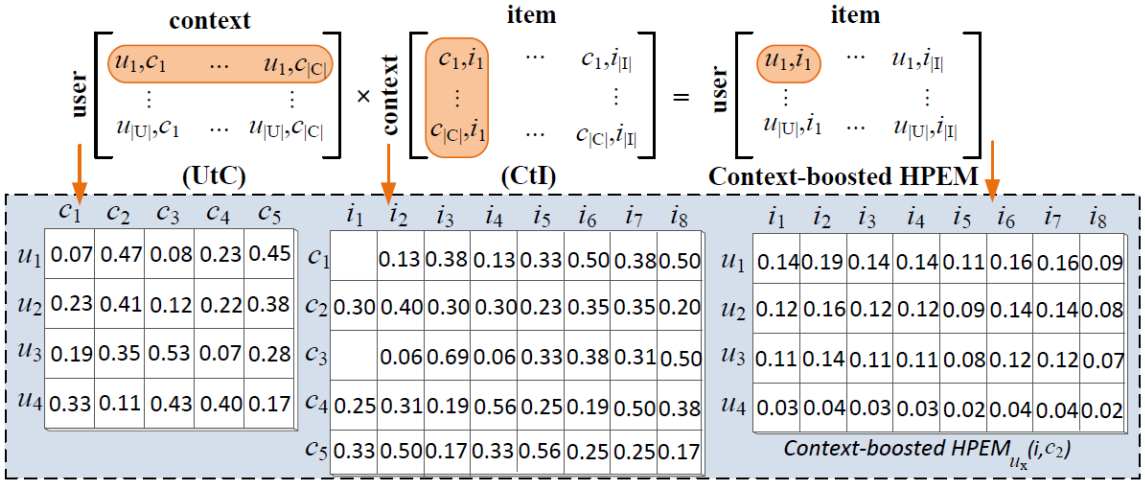


Figure 4.14: An illustration of the process of computing the user-item matrix in a Context-boosted HPEM model.

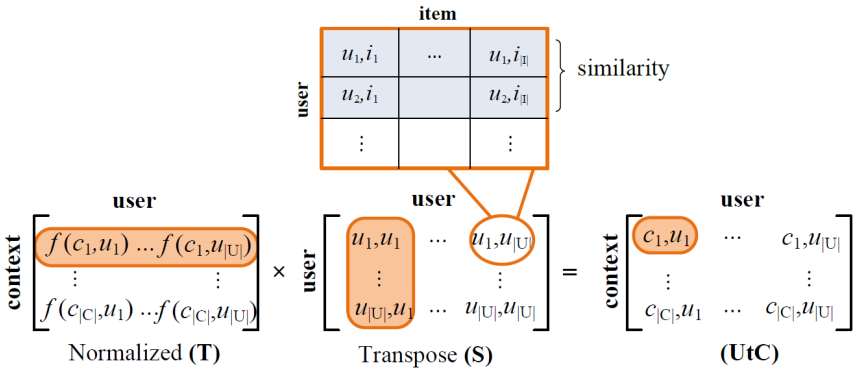


Figure 4.15: The computation step of the **UtC** matrix in the Context-boosted HPEM model (Emphasis on the user).

b. Emphasis on the User

The Context-boosted HPEM model uses the two previously created hidden preferences matrices to associate the user-item relationship to a context. This proposed method produces item recommendations relevant to a given context while also considering user preferences. The estimation of rating values for user-item can be computed using Equation 4.15.

$$Rate_{u,c}(i) = \mathbf{CtU}_{c,u} \times \mathbf{CtI}_{c,i} \tag{4.15}$$

Where $\mathbf{CtU}_{(c,u)}$ is the entry value of the c-th row and the u-th column in the **CtU** matrix, and $\mathbf{CtI}_{(c,i)}$ is the entry value of the c-th row and the i-th column in the **CtI** matrix. Equation 4.15 enables us to extract the hidden preferences of user u according to a detected context c, taking into consideration the user's previous items and the similarity of those items to the detected context. Items with a higher rating value would be recommended to the user. Recommended items obtained a higher ranting value due to the reflection of users who used such items in that particular context. Figure 4.15 shows an illustration of this, building the Context-boosted HPEM model with more emphasis on the users.

4.4.3 Context-User-Boosted HPEM Model

In order to trace the relationship between contexts, and analyze those that have a greater influence on the user's choices, we compute the similarity between each pair of contexts.

By decomposing the context-item matrix $\mathbf{A}_{|C|\times|I|}$, we build the context-context matrix $\mathbf{Q}_{|C|\times|C|}$ by computing the similarity of a pair of vectors, in order to find items that were consumed in each given context. The similarity value is calculated either by counting the frequency of consumption of an item for each pair of contexts, or by using a binary function. We then measure the cosine angle between the two computed values for all items consumed in one context compared to another. Likewise, we also want to trace the relationship between different items consumed by the users. Accordingly, we build the item-item similarity matrix $\mathbf{E}_{|I|\times|I|}$ by computing the similarity between two items from the matrix $\mathbf{R}_{|U|\times|I|}$. We can identify items that have higher rating values after computing all the rating values by multiplying the entries of **UtC** and **UtI**, based on the given context, as in Equation 4.16.

$$Rate_{u,c}(i) = \mathbf{UtC}_{u,c} \times \mathbf{UtI}_{u,i} \tag{4.16}$$

4.4.4 Context-Item-Boosted HPEM Model

In this model, we take into consideration the user's previous items, their nearest neighbors, as well as the similarity of the items that have previously been consumed in the detected context. Items with a high rating value will be recommended to the user; the recommended items reflect the user's current context. In order to compute the items' final rating, we use the two previously described models: the hidden context-item model (**CtI**), and the hidden item-user model (**ItU**). The association of these two models builds the required contextual bridge between users and items. Specifically, the proposed recommendation produces item recommendations relevant to a given context by extracting hidden preferences. The calculation of the user-item rating value is computed by:

$$Rate_{u,c}(i) = \sum_{c \in m} \alpha \mathbf{CtI}_{c,i} \times \mathbf{ItU}_{i,u} \tag{4.17}$$

Where $\mathbf{CtI}_{(c,i)}$ is the matrix entry of the $c-$th context row and the $i-$th item column of the **CtI** matrix. The $\mathbf{ItU}_{(i,u)}$ is the entry value of the i-the row and the u-th column in the **ItU** matrix. The parameter α is an attenuation factor, where $\alpha \in (0, ..., 1)$ to reduce the weight factor of a less sensitive context. The tuning of the α value is set after some experimental results. Details of the parameter α are presented in Section 5.4.1. Items with higher rating values are recommended to the user. Figure 4.16 presents an illustration of the user-item rating value computation step. In Equation 4.17, we show that the rating value is not limited to a single context; if multiple contexts are in the

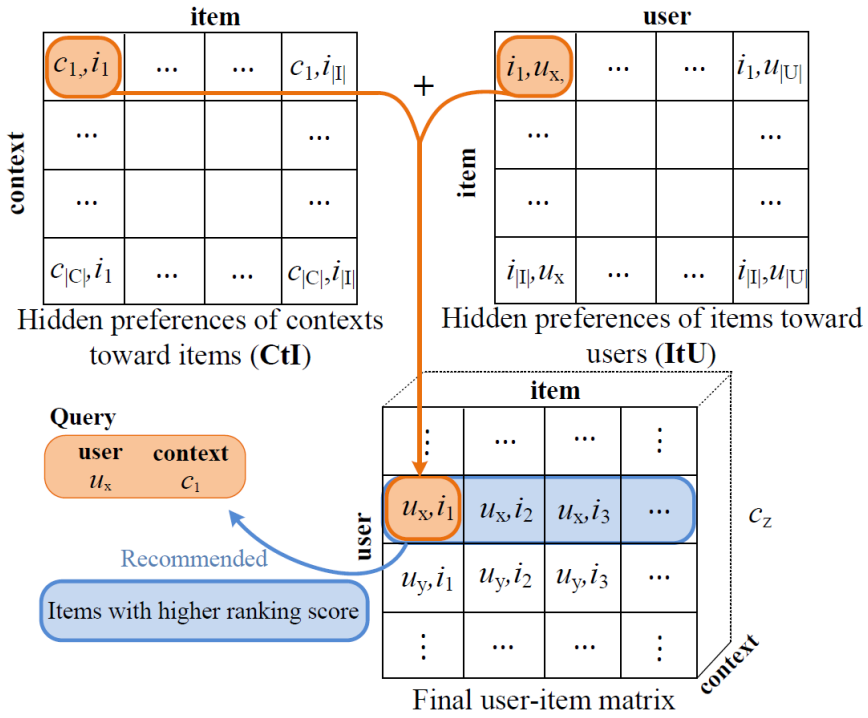


Figure 4.16: The final user-item rating value in a Context-item-boosted HPEM model.

user's query, then the summation of the multiplication will represent the rating value for that item. With regard to the example illustrated in Figure 4.2, the final matrix does not have to be physically stored in the database; its rating value can be computed by executing a query on the two matrices **CtI** and **ItU**, to reduce the model's complexity.

4.5 Optimizing the HPEM Model

The three previous versions of the HPEM model were used to develop contextual data and search for the hidden relationships between the context of a consumed item and the preferences of users in a specific context. The identification of these two types of targeted associations are used to build a recommendation model for the unknown rating of items in different possible contexts, as described in the three models seen in Sections 4.4.2, 4.4.3, and 4.4.4. Table 4.2 summarizes the different models of finding the hidden preferences. In this section, we propose to benefit from the three models as well as from the use of an optimization function to maximize the MAP measure of the resulted recommendation.

HPEM Model	Emphasis On	Similarity Function	Exploring On
Context-boosted	Contexts User	Boolean on (contexts, items) Boolean on (users, items)	$\mathbf{UtC}_{\|U\|\times\|C\|}$, $\mathbf{CtI}_{\|C\|\times\|I\|}$
Context-user-boosted	User	Frequency on (contexts, items)	$\mathbf{UtC}_{\|U\|\times\|C\|}$, $\mathbf{UtI}_{\|U\|\times\|I\|}$
Context-item-boosted	Item	Frequency on (users, items)	$\mathbf{CtI}_{\|C\|\times\|I\|}$, $\mathbf{ItU}_{\|I\|\times\|U\|}$
Optimized HPEM	Users, contexts, items	Frequency on (users, contexts, items)	$\mathbf{CtI}_{\|C\|\times\|I\|}$, $\mathbf{UtI}_{\|U\|\times\|I\|}$ $\mathbf{UtC}_{\|U\|\times\|C\|}$

Table 4.2: A summary of the different methods of building the proposed HPEM model.

4.5.1 Optimizing the Search for Hidden Preferences

We can predict the hidden preferences of a user for their current context by capturing how he/she behaves in a particular context, in terms of contextual item selection. The matrix $\mathbf{UtC}_{|U|\times|C|}$ is built to capture the preferences of a user u to contexts that are similar to a particular context, as illustrated in Section 4.4.2. The hidden user-context preferences are the result of the product of the normalized frequency matrix of ($\bar{\mathbf{B}}$), and of the transpose of the similarity matrix $(\mathbf{Q})^T$.

Similarly, we can predict the hidden preferences of items toward their detected contexts by capturing how a particular context is behaving with the user's selection of items, in terms of items rather than users. Matrix $\mathbf{CtI}_{|C|\times|I|}$ is built for this purpose. The **CtI** represents the results of the product of the normalized frequency matrix of ($\bar{\mathbf{A}}$) and the transpose of the similarity matrix $(\mathbf{E})^T$, as in Equation 4.13. We use the same normalization approach used for matrix $\bar{\mathbf{B}}$ when constructing $\bar{\mathbf{A}}$.

The final search step in finding hidden preferences is finding them within the users and items represented by the matrix $\mathbf{UtI}_{|U|\times|I|}$. To capture the user's hidden preferences toward an item, we search for items that are either similar to their preferences or similar to the choices of their nearest neighbors (similar users). We build the matrix **UtI** according to Equation 4.18.

$$\mathbf{UtI}(u,i) = \sum_{m=1}^{|U|} (\bar{\mathbf{R}})^T_{i,m} \times (\mathbf{S})^T_{m,u} \tag{4.18}$$

Where $(\bar{\mathbf{R}})^T_{i,m}$ is the transpose of the original normalized rating matrix $\mathbf{R}$, and $\mathbf{S}$ contains

	c_1	c_2	c_3	c_4	c_5
u_1		2		1	2
u_2	1	2		1	2
u_3		1	2		1
u_4	1		2	2	1

Table 4.3: An illustrative example of building the user-context matrix $\mathbf{B}$.

the top k user-user similarity matrix. The product of the two matrices brings the user and their nearest neighbors' preferences to a given item.

4.5.2 Example of Searching for Hidden Preferences

This subsection provides an illustrative example of the process of building the hidden preferences models, considering the graph given as a reference in Figure 4.2. Let us start from the table at the bottom of Figure 4.2. The table shows the ratings collected for users to items in different contexts. We assume that the rating values do not change according to the context; the ratings given by users to items are independent of the user's detected context. Only item selections are affected by their contexts. If the context, according to a given application, does affect the ratings given to items, then the overall rating values can be obtained by using an aggregate function like the average. Hence, we can obtain the matrix $\mathbf{R}_{|U|\times|I|}$. Next, we can obtain the user-context matrix $\mathbf{B}_{|U|\times|C|}$ when we aggregate contexts over their associated items for each user entry, as shown in Table 4.3. We can normalize the matrix $\mathbf{B}$ into a range between 0 and 1. For instance, $b(u_2, c_2) = \left(2/\sqrt{(2)^2 + (2)^2 + (1)^2 + (0)^2}\right) = 0.67$. Similarly, we can obtain $b(u_2, c_1) = 0.71$ and notice that because the number of items consumed in c_1 is less than half the number of items consumed in c_2, the difference between the two context weights for u_1 increases significantly as many users may consume items in a common context. We follow the same procedure to construct the context-item matrix $\mathbf{A}_{|C|\times|I|}$ as shown in Table 4.4.

The next step is to compute the similarity between entries belonging to a single dimension, as explained in Section 4.3. We begin by finding the similarity of items $\mathbf{E}_{|I|\times|I|}$ from the matrix $\mathbf{R}_{|U|\times|I|}$. Here, we consider top-$k$ similar items, where $k = 3$ in this example, to reduce the cost of computing, as described later in Section 5.6.1. The similarities of items are shown in Table 4.5. When we calculate the similarities

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
c_1						1		1
c_2	1	1	1			1	1	
c_3			2			1		1
c_4				2			1	1
c_5	1	2			3			

Table 4.4: An illustrative example of building the context-item matrix **A**.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
i_1	1	0.50		0.50				
...								
i_3			1		0.33			0.50
...								

Table 4.5: Finding item similarities between i_1 and the top-3 similar items.

	c_1	c_2	c_3	c_4	c_5
c_1	1	0.32	0.58		
...					
c_3	0.58	0.55	1		
...					

Table 4.6: Finding context similarities between c_1 and the top-3 similar contexts.

	u_1	u_2	u_3	u_4
u_1	1	0.44		
...				
u_3		0.56	1	0.54
...				

Table 4.7: Finding user similarities between u_1 and the top-3 similar users.

	c_1	c_2	c_3	c_4	c_5
u_1	0.211	0.893	0.365	0.530	0.872
u_2	0.918	0.893	0.773	0.734	0.872
u_3	0.514	0.834	0.890	0.061	0.436
u_4	1.115	0.501	1.115	1.021	0.316

Table 4.8: Predicting the hidden preferences of users toward new and previously consumed contexts.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
c_1	0	0	0.289	0	0.192	0.577	0.289	0.577
c_2	0.931	0.801	0.447	0.577	0.428	0.689	0.707	0.577
c_3	0	0	1.183	0	0.192	0.577	0.289	1.025
c_4	0.500	0.500	0.289	1.000	0.236	0.236	0.996	0.931
c_5	1.154	1.248	0.333	0.801	1.000	0.557	0.333	0

Table 4.9: Predicting the hidden preferences of contexts toward items.

between contexts $\mathbf{Q}_{|C|\times|C|}$, we consider the context-item matrix $\mathbf{A}_{|C|\times|I|}$. Table 4.6 shows an example of the calculated context similarities. As for constructing the user-user similarity matrix $\mathbf{S}_{|U|\times|U|}$, the rating matrix $\mathbf{R}_{|U|\times|I|}$ is used to find the similarity of users in terms of item consumption. Table 4.7 shows an example of creating the user-user similarity matrix.

The last step is creating the matrices that represent the hidden preferences (**UtC**, **CtI**, and **UtI**). We utilize the normalized frequency matrix of ($\bar{\mathbf{B}}$) and the context similarity matrix (**Q**), which can be created according to Equation 4.11, to construct the **CtU**. For example, the hidden preferences of u_1 toward context c_1 is calculated as follows: $(0 \times 1) + (0.67 \times 0.32) + (0 \times 0.58) = 0.21$. The hidden preferences of all users toward contexts are presented in Table 4.8.

The results in Table 4.8 represent the first prediction step for user-item recommendations. For each user, the computed values are assigned to new, never consumed before contexts such as (u_1, c_1), as well as to contexts in which users have previously consumed items, such as (u_1, c_2). The exact matrix multiplication concept is applied to find the hidden preferences of items to contexts (**CtI**). Using Equation 4.13 and the context-item matrix $\mathbf{A}_{|C|\times|I|}$, obtained in Table 4.4, we can compute the hidden preferences of an item to a given context. For instance, the prediction of the hidden preferences of context

c_2 to item i_1 is computed as follows: $(0.707 \times 1) + (0.447 \times 0.5) + (0 \times 0.5) = 0.931$, as demonstrated in Table 4.9. By creating this model, we use the item similarities to estimate the weight of each context to an item (i) by retrieving the items consumed, for each context, which are similar to the given item (i). The remaining model ($\mathbf{UtI}_{|U|\times|I|}$) is built from the rating matrix $\bar{\mathbf{R}}_{|U|\times|I|}$ and the user-user similarity matrix as in Table 4.10.

4.5.3 Context-Aware Rating

The three model versions (**UtC**, **CtI**, and **UtI**) are used to compute the expected rating value of item i for user u by using a given list of contexts m as a query. Hence, we propose adding the context of the users with their quest to consume a new item. Accordingly, the detected context helps boost the search for relevant items. However, relying only on the relation between contexts and items (as in **CtI**) might not significantly improve the search for items relevant to the user's needs. Therefore, for a given user query (m), we propose a balance algorithm that considers the hidden preferences of user to context, the hidden preferences of context to items, and the rating of users to items, according to Equations 4.19 and 4.20:

$$Relevance_{u,c}(i) = \alpha CtI_{c,i} \times \beta UtI_{u,i} \tag{4.19}$$

$$Rate_{u,c}(i) = \sum_{c \in m} Relevance_{u,c}(i) \times UtC_{u,c} \tag{4.20}$$

where m contains some contextual parameters submitted with a given query as $m = \{c_1, c_2, ..., c_m\}$ and $m \leq |C|$. $\mathbf{CtI}_{c,i}$ is the matrix entry of the $c-$th row and the $i-$th column of the **CtI** matrix. The $\mathbf{UtI}_{u,i}$ is the entry value of the $u-$the row and the $i-$th column in the **UtI** matrix. The parameters α and β are two attenuation factors, where $\alpha, \beta \in (0...1)$, that are used to reduce the weight factor of a less sensitive context. The attenuation of α and β values are set after some experimental results. Details of the attenuation of these two parameters are explained in the section Optimizing the MAP (Section 4.5.5). Items with higher rating values ($Rate_{u,m}$) are recommended to the user.

4.5.4 Example of Computing the Context-Aware Rating

Following the example presented in Section 4.5.2, we continue showing how the proposed recommendation model recommends items to users for a given set of detected contextual

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	1.000	0.961	0	0.951	0.270	0.264	0.373	0
u_2	0	0.789	0.732	0.800	1.195	1.051	1.099	0.474
u_3	0	0.445	1.131	0.451	1.228	1.138	0.764	0.539
u_4	0	0.374	1.003	0.379	1.116	0.716	0.932	1.000

Table 4.10: Predicting the hidden preferences of users toward items.

parameters. Let us assume we want to recommend items to user u_1 and his/her current context is c_3. Using Equation 4.19, we can compute the score of items that can be recommended in this situation. For instance the value of $Relevance_{u_1,c_3}(i_5)$ without using any attenuation parameters (that is $\alpha, \beta = 1$) is computed as follows: $CtI_{c_3,i_5} = 0.192$ in Table 4.9 and $UtI_{u_1,i_5} = 0.270$ in Table 4.10, then $Value_{u_1,c_3}(i_5) = 0.192 \times 0.270 = 0.052$. For another query, suppose we want to calculate the value of i_6 for user u_1 in a given context c_2. The $Relevance_{u_1,c_2}(i_6)$ can be calculated as 0.182. We notice from these two examples, that even though u_1 consumed few items in very few contexts, the model is still able to predict their expected preferences. In fact, the contexts detected, the steps needed to find the items, and the users' hidden preferences toward these items help us bridge the gap between users and new items.

To cover the scenario where a user's context can be presented as a set of contextual facts (e.g. time: evening, social: with a partner), we run a set of contexts in our query to calculate item values. For example, we want to run the query for user u_1, given two contexts (c_2 and c_3) at once for i_7. The $Relevance_{u_1,(c_2,c_3)}(i_7)$ can be computed as: $(0.707 \times 0.372) + (0.289 \times 0.373) = 0.371$. Table 4.11 shows examples of calculating item values for u_1 for different queries, as well as the results obtained for different users.

The item values in Table 4.11 show how the values change for each user and for each context. Hence, the final rating value can now be calculated by considering the user's hidden preferences toward the context given in the query. For instance, the rating of items for u_5 given contexts c_3 and c_5 using Equation 4.20 is shown in Table 4.12. For example, the $Rate_{u_3,(c_3,c_4)}(i_5) = (0.236 \times 0.890) + (0.289 \times 0.061) = 0.228$, where 0.89 and 0.061 represent the u_3 hidden preferences toward c_3 and c_4 respectively, as in Table 4.8. Accordingly, the top-2 new items that can be recommended to u_3 in such given contexts are i_8 and i_7.

User	Query	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
u_1	c_2	0.931	0.769		0.549	0.116	0.182	0.263	
u_1	c_4	0.500	0.480		0.951	0.064		0.371	
u_1	c_2, c_3	0.931	0.769		0.549	0.168	0.334	0.371	
u_2	c_3, c_4		0.395	1.077	0.800	0.512	0.607	1.412	0.926
u_3	c_3, c_4		0.222	1.665	0.451	0.526	0.657	0.981	1.054
u_4	c_3, c_4		0.187	1.477	0.379	0.478	0.413	1.197	1.955

Table 4.11: the top part of the table shows examples of item values obtained for u_1 in different context(s) represented in queries. The bottom part shows the results obtained for each user, given contexts c_5 and c_6.

Query	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
c_3			1.338		0.236	0.657	0.220	0.552
c_4		0.222	0.327	0.451	0.289		0.760	0.502
c_3, c_4		0.014	1.211*	0.027	0.228*	0.585*	0.242	0.522

Table 4.12: The rating values calculated for items for user u_3 given c_3 and c_4. (* Items that have previously been consumed by the user in such contexts).

4.5.5 Optimizing the MAP

With the contextual information associated to the user-item relationship model for recommendations, we want to optimize the item selection for users in a given context. Based on the rating value calculated for items, we can generate the top k recommended items to the user, and calculate the precision and recall to measure how relevant the items are to the user's quest. The MAP function is used to measure the performance of the proposed model, according to Equation 4.21, where t_u denotes the number of test cases for user u, and P_n is the precision at top n.

$$\begin{aligned} MAP &= \tfrac{1}{|U|}\textstyle\sum_{u=1}^{|U|}\sum_{c=1}^{|C|}\tfrac{1}{|t_u|}\sum_{n=1}^{|t_u|}(Pn \times Rn) \\ Rn &= \begin{cases} 1 & item\,is\,relevant\,at\,rank\,n \\ 0 & otherwise \end{cases} \end{aligned} \tag{4.21}$$

From Equations 4.19 and 4.20, we notice that the rating value changes depending on the context and the user in the query. Therefore, the precision and the MAP is changing in a non-smoothed way. Accordingly, before we can use any optimization function, it is required to first apply a smoothed precision function with respect to U, I and C. We estimate the number of relevant items that represent the precision at top k, according to Equation 4.22. We can then modify the earlier function of calculating the MAP to a smoothed one, as in Equation 4.23.

$$Rn(r_i \leq r_j) \approx s(Rn) = 0.5 + 0.5tanh(\frac{Rn-1}{1}) \tag{4.22}$$

$$MAP = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{c=1}^{|C|}\frac{1}{|t_u|}\sum_{n=1}^{|t_u|}(Pn \times s(Rn)) \tag{4.23}$$

After smoothing our MAP function, we can optimize it by using a standard gradient ascent method according to Equation 4.24. Appendix A contains the algorithm for optimizing the HPEM model.

$$L(U,I,C) \;=\; \sum_{u=1}^{|U|}\sum_{c=1}^{|C|}\frac{1}{|t_u|}\sum_{n=1}^{|t_u|}(Pn \times s(Rn)) - \frac{\lambda}{2}\left(|U|^2 + |I|^2 + |C|^2\right) \tag{4.24}$$

4.6 Exploring Hidden Contexts

Resources, referred to as items, are assigned to different contextual annotations. However, these annotations are given by users subjectively, and can therefore have different meanings associated to their textual descriptions. Accordingly, we need an automatic algorithm that analyzes the relationship between the contextual annotations and the annotated items to uncover the hidden associations between the two. This step is the first in the construction of the user-context matrix $\mathbf{T}_{|C|\times|U|}$ and $\mathbf{B}_{|U|\times|C|}$, and the item-context matrix $\mathbf{A}_{|I|\times|C|}$, respectively. These two matrices are then used in the recommendation model to compute the expected rating value of item (i) for user (u) by using a given list of contexts (c), as a query.

We propose using the PLSA model to obtain a quantitative representation of the textual annotations associated to the items and users [38]. The PLSA model examines a set of occurrences of pairs of factors or parameters $Z = \{z_1, z_2, ..., z_{|Z|}\}$. First, the PLSA associates unobserved or hidden classes with each observation of user and context (u,c) and item and context (i,c). The hidden class is also referred to as hidden topics in a document originally proposed in [37]. The associated Z variables represent the hidden causes for which an item is consumed in a context c, as well as the reasons why users prefer to consume certain items in specific contexts. The number of z variables demonstrates the number of possible conditions or states that bundle users and items into a number of latent causes $(|Z|)$ [38]. The value of $|Z|$ should be less than the number of users and items to enable the model to find some common reasons by which to categorize users and items independently. If the value of $|Z|$=1, then the recommendation model ignores the effects of contexts since the selection of items by users would not depend on the detected context. For example, we can find the probability that a word belongs to a document by using the following equation:

$$P(word|document) = \sum_{z\in Z} P(word|z)P(z|document) \tag{4.25}$$

where z represents the hidden variables among the list Z. In this thesis, this model is used to maximize the performance of the HPEM model.

4.6.1 Co-Occurrence Pairs of Context

The hidden variable z is associated with every pair of user and context (u,c) observed, as well as every pair of item and context (i,c), as shown in Figure 4.17. The identity of the

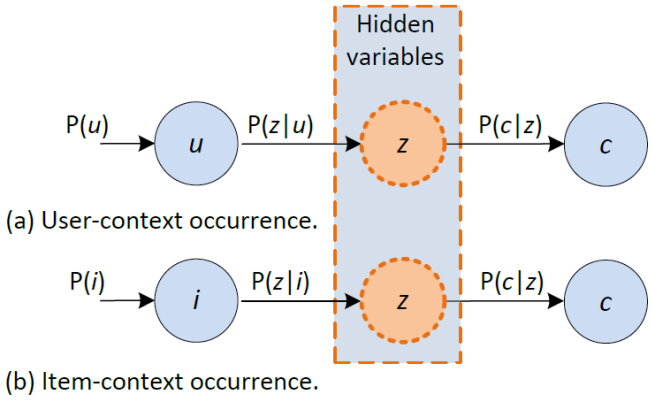


Figure 4.17: Graphical representation of the PLSA model.

user and context in our case is assumed to be conditionally independent on the latent variable z, allowing us to obtain the model in Equation 4.26. Similarly, the independent assumptions are reflected in the associations of items given a particular context, as in Equation 4.27.

$$P(c|u) = \sum_{z} P(c|z)P(z|u) \tag{4.26}$$

$$P(c|i) = \sum_{z} P(c|z)P(z|i) \tag{4.27}$$

After training the model parameters, we use the probability distribution to estimate the preference of a user toward a context, and the probability that an item is suitable for a context. Note that this step is a preliminary step in the estimation of the final preference of a user towards an item in a given context. Using the maximum likelihood statistical approach, we can compute the $P(u, c)$ and $P(i, c)$ according to the following log-likelihood functions:

$$L_u = \sum_{u}\sum_{c} log(P(u, c)) = \sum_{u}\sum_{c} log(\sum_{z} P(c|z)\, P(z|u)) \tag{4.28}$$

$$L_i = \sum_{i}\sum_{c} log(P(i, c)) = \sum_{i}\sum_{c} log(\sum_{z} P(c|z)\, P(z|i)) \tag{4.29}$$

We then follow the Expectation Maximization (EM) algorithm [23] to approximate the log-likelihood with the PLSA model introduced. The EM algorithm has two main steps:

E-step and M-step. The E-step computes the posterior probability of hidden variables z with respect to each observation in the model (Equations 4.30 and 4.31).

$$P(z|u,c) = \frac{P(c|z)P(z|u)}{\sum_{z'} P(c|z')P(z'|u)} \tag{4.30}$$

$$P(z|i,c) = \frac{P(c|z)P(z|i)}{\sum_{z'} P(c|z')P(z'|i)} \tag{4.31}$$

The M-step represents the computation required for the conditional distributions as shown in Equation 4.32 and 4.33. Appendix B contains only the description of computing the user and context $P(c|u)$ using PLSA algorithm.

$$P(c|z) = \frac{\sum_u P(z|c,u)}{\sum_{c',u} P(z|c',u)}, P(z|u) = \frac{\sum_c P(z|c,u)}{\sum_{z,c} P(z|c',u)} \tag{4.32}$$

$$P(c|z) = \frac{\sum_i P(z|c,i)}{\sum_{c',i} P(z|c',i)}, P(z|i) = \frac{\sum_c P(z|c,i)}{\sum_{z,c} P(z|c',i)} \tag{4.33}$$

4.7 Summary

Most recommender systems personalize multimedia content to the users by analyzing two main dimensions of input: content (item), and user (consumer). In this chapter, we propose a new recommendation model to improve the recommendation and the quality of the user experience by analyzing the context of users at the time when they wish to consume multimedia content. We present the algorithm used to construct the HPEM model. The inceptive idea is that users sharing an item such as music are likely to also share some hidden contextual information. Such contextual information is able to effectively describe the user's preferences toward their selected items. The analysis of the available contextual information associated with consumed items enables the analysis of items consumed in similar contexts. In general, users who consume certain items in a given list of contexts are more likely to form a contextual pattern to bridge the information gaps between users and new items. To deal with such collected contextual parameters, a new recommendation model is proposed to provide multimedia recommendations that are more relevant to the collected contextual information.

Our methods profited from the collaborative filtering and social tags to explore context-based item recommendation that does not rely on the need to analyze the item contents in order to detect the context. This is due to the vast amount of online users

who consume and prefer different types of music, and in doing so contribute to the classification and annotation of different media more effectively than by just relying on feature extractions. By leveraging the social tagging, our proposed model computes the hidden preference of users on contexts from other similar contexts, as well as hidden assignment of contexts for items from other similar items. Additionally, the proposed model finds the hidden preferences of users in a given context from other similar users. By finding the similarities between the user's contexts and between the contexts and items, we can determine the desirable items given a particular context. We then map the context on the items depending on that particular user, in order to recommend the most relevant items suitable for his/her needs.

Chapter 5

Evaluations

This chapter explains how to evaluate the performance of the recommendations. The research question is, first, whether the utilization of the user's context to recommend a different number of items contributes to the improvement of the recommendation performance; second, how the model is performing compared to the baseline methods. We also present the evaluation of the proposed model's computational complexity. We changed the number of items retrieved each time to measure their relevance to the user's request. We also indicate the method used to analyze the proposed recommendation algorithms, and the methodology used to collect the related datasets. The evaluation is divided into different steps, starting with experimental hypothesis and methodology, then the experimental results, and finally the computational analysis.

5.1 Evaluation Methodology

The evaluation procedure is divided into two parts: the first part measures the accuracy of the context-based recommendation prediction, using offline experiments on different datasets crawled from online multimedia databases; the second part evaluates the user's satisfaction with the context-based recommendations, after using the proposed prototype applications.

5.1.1 Offline Experiment

The evaluation process that measures the accuracy of the context-based recommendations is done using offline experiments that follows the procedure described in [14]. We

randomly divided the experimented datasets into two groups: a training set that represents 80% of the original dataset, and the remaining 20% used as a test set. In order to build the test set, we randomly withheld one item, including the assigned contextual tags within the user profile; we later used them as test-queries for each user. For instance, if we take one item from u_1 in our previous example in Figure 4.2, we can randomly select i_2 and remove it from u_1's profile, which includes removing its contexts c_2 and c_5. When we want to test the model's performance, we run queries based on the test set, which contains u_1's removed item(s), by providing the removed contexts, which in this case are c_2 and c_5. Based on the results of the proposed HPEM's rating function ($Rate_{u_1,(c_2,c_5)}(i)$), we can determine if item (i_2) is among the top-k resulted items. The division of our dataset into training and test sets might be sensitive to the randomly selected items/contexts. Therefore, to ensure that our evaluation is not vulnerable to the randomness of the division step, we ran the model 5 times, each time with a different partitioning. The performance results presented in the rest of this thesis represent the average of the 5 different runs, with the standard deviation.

We also evaluated the model's performance with different groups of users, to determine if the number of ratings could affect the quality of the recommendations. The results did in fact show that the recommendations for individual users could be affected by the number of items consumed by each user. In reality, some users are active and rate movies and music more frequently than other users who are passive and hardly express their opinions about the contents they consume. Passive users do not have large amounts of historical data for item and context discovery, which why we divided users into three groups: Passive Users (PU), Normal Users (NU), and Active Users (AU). A user is assigned to the PU group if the number of items rated is less than four within a particular context. A user is assigned to the NU group if the number of items he/she rated is larger or equal to four items but less than ten items. Users who rated more than ten items in a particular context are assigned to the AU group. We then calculated the MAP. The MAP values indicate if the proposed model is sensitive to the number of items consumed by a user. We also tested to see if the proposed model could still achieve a better performance, even if the user had very few items assigned to contexts. This problem is known as a cold-start problem, where there is not enough information about the user and their contexts to predict relevant items. The cold-start problem is an important issue to consider when implementing a recommender system.

5.1.2 Subjective Evaluation

In addition to the offline dataset experiments, we also conducted a subjective user evaluation using the android prototype application introduced in Section 5.1.4. Providing users with some contextual knowledge, we computed the suitable items to be recommended, according to the given context, and displayed them to the user. In order to measure the effectiveness of our model, the recommendation algorithm presents the recommended items retrieved using another algorithm that is not personalized to the user's context. We used the collaborative filtering algorithm presented in [92] as the second recommendation technique used in the application. The goal behind this experiment is to determine whether or not our context-aware prediction method can provide recommendations that are more relevant than those made by the non-context-aware technique.

Each user in the online experiment is asked to evaluate the ten pieces of music recommended in a specific context. Subjects go through different context scenarios, thus they evaluate the recommendations made for these different conditions. Subjects browse the recommended music and provide feedback as to whether they like or dislike the music in such a context.

5.1.3 Dataset

To find a publicly available dataset that carries some contextual information is a crucial challenge. Such lack of availability challenges the design of any context-aware recommendation algorithm [111]. As a solution, we crawled our dataset from an online social music database: last.fm. Specifically, music information and annotation data are extracted from the last.fm website. Last.fm is an online social music and online radio resource that enables their users to subscribe, listen to and tag their favorite albums, tracks, and artists. We obtained a number of tags from each user profile, and in our model, each one represents an element of context. Users of last.fm annotate different albums and tracks with textual tags. Due to the fact that users can give any textual description to their favorite tracks and albums, different words can be used to describe the same meaning. For instance, four different users may tag item i_1 with different words that have the same meaning: "relaxing music", "for relax", "relax", and "relaxation". These tags can be grouped together under one annotation: "relaxing". Accordingly, we ended up with semi-synthetic context data. In addition to the collected tags, we also collected the user's profile information.

Initially, the crawled dataset was too sparse and we had to clean it before starting

the experiments. The cleaning process involved the removal of items that had only been selected by a very small number of users, as well as by users who consumed very few items. In addition, we cleaned items that had been tagged by all users with less than five contexts. Similarly, we removed tags that had been annotated by less than five users.

We also tested the performance of our proposed system on a bigger dataset. We used the publicly available dataset from MovieLens[1] (www.movielens.org), which was used in [34]. The MovieLens dataset consists of 943 users, 1682 movies, and 100,000 user-item ratings. This dataset does not have explicit contextual data, but contains some information that we consider a context in our experiment, as a proof of concept: genre (romance, action, adventure etc.), and user information (age, gender, occupation and zip code).

5.1.4 Applications

In order to improve the user satisfaction and experience, the method of interaction should be convenient for users, to ensure that they have access to different services. Accordingly, we have implemented two prototype applications that support context-aware recommendations. The prototypes are designed to collect the contextual information needed to enhance the recommendation results. In addition, both prototypes facilitate the collection of the user's physiological and environmental parameters and to allow the modification of the user's surrounding environment, using AmI interfaces. The two prototype applications are: an interactive application mounted on a mirror or a TV named RecMirror, and a smart biosensor recommendation application on android named Biommender.

RecMirror Application

RecMirror is an interactive application that can be mounted on a regular mirror or connected to a TV. RecMirror collects, in real-time, a number of contextual parameters, and shares them with a recommendation engine. In this application, we use Microsoft's Kinect Xbox 360 camera for user detection and interactions. The camera enables the application to detect the skeleton of the body and the number of users sitting in front of the camera, as shown in Figure 5.1. Using an android application that acts as a smart remote controller, once the user presses the movie button, the camera can detect who is sitting in front of the TV and adapt the recommendations according to the detected profiles and context. The Kinect camera currently used only allows the detection of two

[1]The MovieLens dataset can be downloaded from: http://www.grouplens.org/node/73.

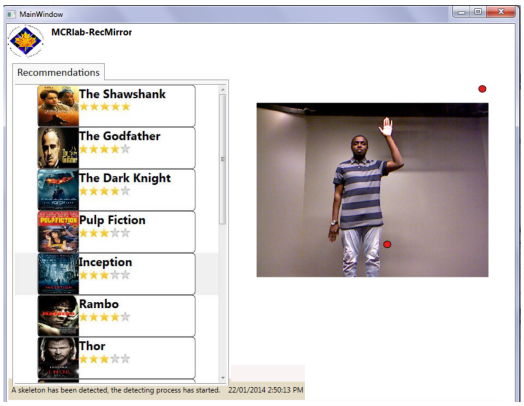


Figure 5.1: Snapshots of the RecMirror application.

Figure 5.2: Illustration of the proposed recommendation application, a part of the home entertainment system.

skeletons at a time, per camera. Such a limitation therefore exists in this prototype as well, but could be enhanced by using multiple cameras with an advanced detection algorithm. As proof of concept, in this prototype we are only interested in detecting two users at a time, in order to identify their preferences and adapt the recommendation results accordingly.

As an illustrative scenario, a 35-year-old man uses the smart mirror system at home right after he returns from work; the system finds his stress level to be beyond what is normal; after analyzing the collected information using the recommendation algorithm, the system suggests a list of his favourite music, in order to reduce his stress.

RecMirror does not necessarily have to be attached to a mirror in a home environment; it can also be installed and connected to a smart environment within the home entertainment system, for example a TV (see Figure 5.2).

Figure 5.3: Illustration of the Biommender application prototype used with a heart monitor sensor for the context-aware recommendations.

Biommender Application

A smartphone can enable many features of the recommender system, especially in detecting the user's context, for example location, activity, date, time, and access to personalized multimedia collections. It is also an important portable device to act as a bridge between different sensor networks and the recommendation processes. Biommender is an android-based application that provides a stress detection interface to detect the physiological context of the user.

The application is designed to response to the subject's stress level by adapting the light intensity of the room, playing a list of recommended music, and/or changing the room temperature, as in Figure 5.3 and Figure 5.4. We use a heart monitor sensor, a product from AliveCor company, connected to a smartphone device, the Samsung Galaxy III. The application collects the heart signal using the single-Lead ECG sensor for stress detection, and the android built-in GPS sensor for location identification. In this case, date, time, song being played, and song play count, as well as other parameters, can easily be collected. The functionality of the application is to react and deliver multimedia recommendations to the user, as shown in Figure 5.4. It is also used as a remote controller to the recommender system in the TV-based application, as described in the previous section.

The architecture of the introduced recommender system consists of four main layers: the input/output interface layer, the context management layer, the client-local resources layer, and the server-cloud resource layer, as shown in Figure 5.5. Figure C.1 in Appendix

Figure 5.4: A screenshot of the Biommender application prototype interface.

C shows the sequence of main functionality interactions for a user requesting a song. As a proof of concept, the prototype application synchronizes any media content stored within the user's Dropbox account. The input/output interface layer handles the collection of the required contextual data and interacts with the user, which includes delivering the recommendation results. The context-management layer identifies the user's context by analyzing the retrieved sensory data. The local resources layer stores the user behavior, and selects and evaluates the different recommendation parameters needed for the recommendation algorithm to function. The Entity Relationship Diagram (ERD) of the client-local resources layer is demonstrated in Appendix C (Figure C.2). The resource contents and the available social profiles are stored in a cloud-based repository.

5.2 Evaluation Metrics

To measure the HPEM model's retrieval accuracy, we adopted precision and recall, which are widely used evaluation parameters, in order to measure the effectiveness of our offline experiment recommendations. Precision can be calculated by finding the ratio of the recommended items to the items already identified as relevant to the user, as in Equation 5.1 [98]. Recall can be calculated by finding the amount of relevant contents among all recommended contents, as in Equation 5.2 [13].

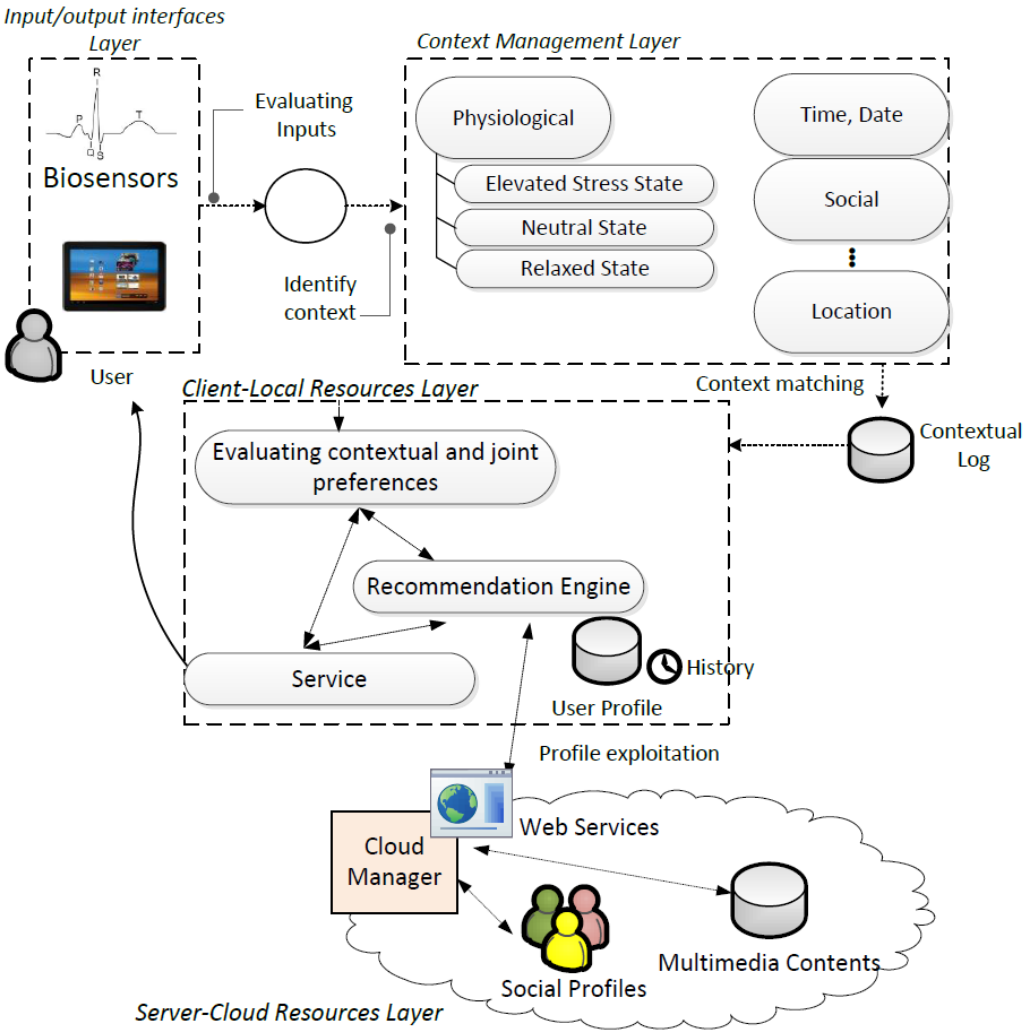


Figure 5.5: The architecture of the Biommender application prototype

$$P_{top\,k} = \frac{1}{|U|} \times \sum_{u=1}^{|U|} \frac{I_u^k}{k} \tag{5.1}$$

$$R_{top\,k} = \frac{1}{|U|} \times \sum_{u=1}^{|U|} I_u^k$$
$$I_u^k = \begin{cases} 1 & item\,appears\,in\,the\,returned\,results\,at\,top-k \\ 0 & otherwise \end{cases} \tag{5.2}$$

For the comparison of our method to the other benchmark algorithms, we report the $F1$ measure by using Equation 5.3, and MAP by using Equation 5.4, in addition to precision and recall.

$$F1 = \frac{2 \times P \times R}{P + R} \tag{5.3}$$

$$MAP = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{t_u} \sum_{n=1}^{t_u} P_n \times R_n \tag{5.4}$$

Where t_u is the number of test cases for user u, and P_n is the precision at top n and R_n is a binary variable that equals to 1 if the item is relevant at rank n [35]. The MAP reports the average precision at each top k result [108]. $F1$ measures integrate the precision and recall performance into one comparable value [35]. Note that we varied the number of items retrieved (top k values) to measure the ranking positions of each recommended item. For instance, the precision values are reported for each top k ($k = 1$, $k = 5$, and $k = 10$), which show the number of relevant items at top 1, top 5, and top 10.

5.3 Evaluating the Context-Boosted HPEM

5.3.1 Emphasis on Context Offline Experiment

Comparison with Other Methods

We evaluate the Context-boosted HPEM rating method using the precision and recall, obtained compared to four other alternative approaches. The four approaches are:

Number of users $\lvert U\rvert$	Number of items $\lvert I\rvert$	Item-context assignments $\lvert I\rvert \times \lvert C\rvert$
2747	7805	1036803

Table 5.1: Statistics of the last.fm dataset used to evaluate the Context-boosted HPEM model.

- **Popular Item (PI)**: The PI algorithm recommends the popular items (i) based on the item count within a specific context (c) to user (u) similar to the method described in [45].
- **Collaborative Filtering (CF)**: The CF algorithm recommends items (i) to user (u) according to the similarity value between user (u_x) and user (u_y), who have similar preferences. This algorithm is used in the experimental evaluations of the work presented in [92].
- **Probabilistic Model (PM)**: The PM approach uses a probabilistic model to estimate the music relevance for different daily activities [100].
- **Graph-Based Flexible Recommendation (GFREC)**: The GFREC algorithm [60] is a graph-based multidimensional recommendation model that uses the technique of the Personalized PageRank algorithm [32].

We varied the number of returned items k from 1 to 10 in order to examine how accurately each algorithm positions a relevant item at a higher rank for a given user's query. The dataset we used to evaluate the performance in this experiment is summarized in Table 5.1.

Sensitivity to Parameters

We investigated the sensitivity of the performance metrics by varying the value of the top items retrieved, denoted as (k), prior to running the experiments. The value of k influences the number of similar records retrieved after performing the similarity functions between user-user, item-item and context-context, as described in Sections 4.3.1, 4.3.2, and 4.3.3. Specifically, the Context-boosted HPEM model retrieval accuracy, which results from the product of model $\mathbf{UtC}_{|U|\times|C|}$ and $\mathbf{CtI}_{|C|\times|I|}$, depends on the value of k. Accordingly, we measured the Mean Reciprocal Rank (MRR) and coverage when k=5,

10, 20, 50, 100, 150, 200, 300, 400, and all. The MRR measure is computed using Equation 5.5, and the coverage is computed using Equation 5.6.

$$MRR_{(k=n)} = \frac{1}{|U|} \sum_{u=1}^{|U|} \left(\sum_{i \in (t_u \cap R_u^n)} \frac{1}{r(i)} \right) \tag{5.5}$$

$$coverage = \frac{1}{|U|} \sum_{u=1}^{|U|} A_u^{|I|}$$

$$A_u^{|I|} = \begin{cases} 1 & item\, appears\, in\, the\, returned\, results \\ 0 & otherwise \end{cases} \tag{5.6}$$

Where t_u represents the test cases for user u, R_u^n is the top n returned records. The value r(i) ranges between: $1 \leq r(i) \leq n$. Referring to the different values of k and having a confidence interval of 95%, we computed the MRR retrieval accuracy, as shown in Table 5.2. The first column represents the MRR and coverage for the similarity results obtained by selecting the top 5 most similar items from the item-item matrix ($\mathbf{E}_{|I|\times|I|}$), and the top 5 most similar contexts from the context-context matrix ($\mathbf{Q}_{|C|\times|C|}$).

k	5	10	20	50	100	150	200	300	400	all
MRR@10	0.215	**0.265**	0.288	0.301	0.309	0.314	**0.320**	0.326	0.328	0.304
	±0.005	**±0.005**	±0.004	±0.004	±0.004	±0.004	**±0.004**	±0.004	±0.004	±0.004
Coverage	88.6%	**88.7%**	91.0%	93.8%	98.1%	83.9%	**98.1%**	96.5%	88.8%	88.7%

Table 5.2: MRR and coverage values @ top 10, with a 95% confidence interval for different values of k.

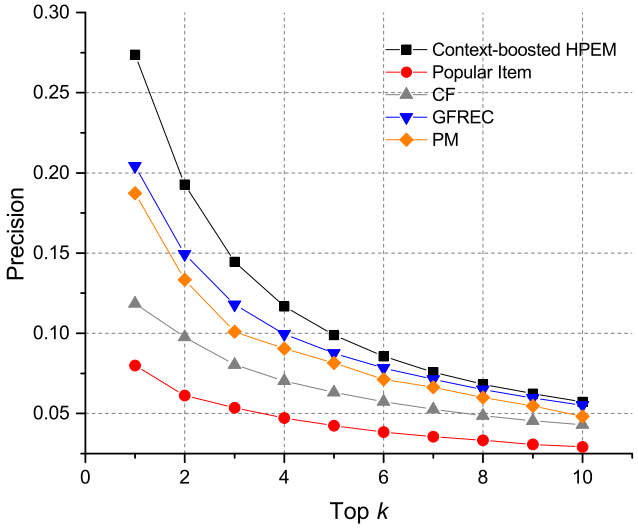


Figure 5.6: Precision at top k for each recommendation method.

The experiment results obtained in Table 5.2 show that the performance of the MRR improved until the values of k reached 200. At such a point when the value of k increases, the rating accuracy increases with little variation compared to the lower values of k. Consequently, when k=200, the number of items and contexts were large enough that the rating accuracy would barely change if more items were retrieved. When k=all, which means retrieving all users, all items, and all contexts, the return accuracy was not much better compared to the smaller values of k. In addition, the coverage obtained when varying the value of k increases until k=100. Accordingly, by reducing the value of k while still obtaining good coverage, we can significantly reduce the computation cost without affecting the accuracy. According to the results of this experiment, we set k=10 for the remaining model experiments.

Experimental Results

As mentioned earlier, we evaluated the performance of the proposed Context-boosted HPEM model by comparing it to the other four alternative methods. Figure 5.6 shows the results of the precision performance with respect to different values of k (top item recommended), according to Equation 5.1. The results show that the PI and CF approaches have the worst performance, and that our approach outperforms all other methods.

We continue to examine the recall and F1 measures of each algorithm, as shown in

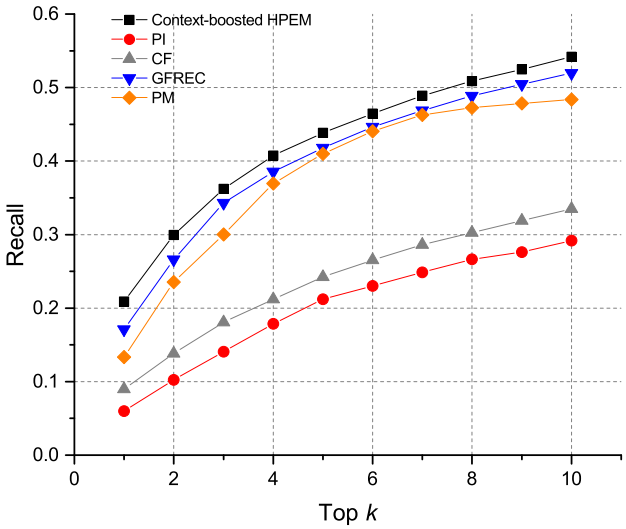


Figure 5.7: Recall at top k for each recommendation method.

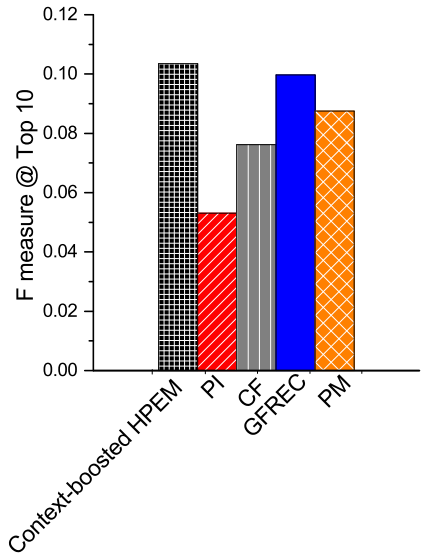


Figure 5.8: The F-measure at top k=10 for each recommendation method.

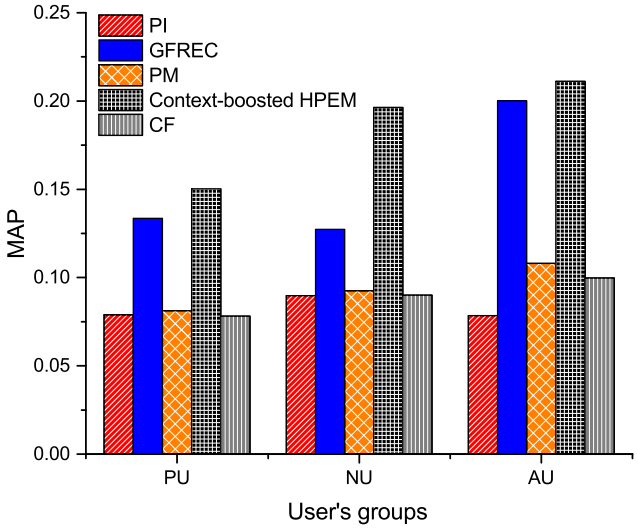


Figure 5.9: MAP at top k recommendation results for three types of users (Passive, Neutral, and Active).

Figure 5.7 and Figure 5.8, using Equation 5.2 and Equation 5.3. The proposed Context-boosted HPEM model obtained approximately 14.89%, 20.89%, 3.8%, and 7.54% improvement on recall (at top 1) and 16.83%, 13.46%, 5.08%, and 8.11% improvement on the F1 measure (at top 1), compared to PI, CF, GFREC, and PM respectively. We conclude that items consumed by other users in different contexts lead to better recommendations than a user's popular items within one context.

We continue computing the MAP according to different groups of users, as explained earlier in Section 5.1.1 and shown in Figure 5.9. The MAP values reported for each algorithm indicate that all algorithms are sensitive to the number of items consumed by a user. We conclude that the more items a user consumed, the better the recommendation they receive. The proposed Context-boosted HPEM model outperforms the other recommendation methods for the three types of users. The results also indicate that our proposed model can achieve better results even if the user does not have enough items assigned to contexts.

Number of users $\lvert U\rvert$	Number of items $\lvert I\rvert$	Item-context assignments $\lvert I\rvert \times \lvert C\rvert$
192	2509	11632

Table 5.3: Statistics of the last.fm dataset used to evaluate the Context-boosted HPEM model with emphasis on the user.

5.3.2 Emphasis on User Offline Experiment

Comparison with Other Methods

In our evaluation, we compared the proposed Context-boosted HPEM model with three baseline methods. In addition to the PI and CF algorithms described in Section 5.3.1, the other method used in the comparisons is:

- **ItemRank**: ItemRank is a random-walk scoring algorithm proposed by [30]. Using a correlation graph, the user preferences are estimated based on the probability that user (u) visits item (i) in a random walk, in a user-item relationship graph.

Table 5.3 describes the size of the crawled dataset from last.fm.

Experimental Results

We evaluated the performance of our model by comparing it with three other baseline methods, with respect to precision, recall, F1 measure, and MAP. We varied the top k values to measure how each algorithm recommends relevant items to users by sorting them by their higher ranting values. The precision and recall values achieved by each recommendation algorithm are presented in Table 5.4 and Table 5.5. The precision values of the columns $k = 1$, $k = 5$, and $k = 10$ show the number of relevant items retrieved at top 1, top 5, and top 10. The precision performance gain of the PI approach is the worst, when compared to the other three methods. Clearly, the Context-boosted HPEM model outperforms the baseline methods.

As shown in Table 5.5, our proposed model achieves approximately 6.73%, 2.77%, 3.62% improvements on recall at top 1, compared to PI, CF, and ItemRank respectively. The results imply that the Context-boosted HPEM model suggests more relevant items in any given context. We also found that when many users consume items in different contexts, the recommendations are more accurate than when only consider popular items

Recommendation Methods	Precision		
	$k = 1$	$k = 5$	$k = 10$
PI	0.237844	0.170085	0.121247
CF	0.298626	0.194292	0.13277
ItemRank	0.335624	0.20666	0.132611
Context-boosted HPEM	0.354123	0.217548	0.14445

Table 5.4: Precision at top k for each recommendation methods.

Recommendation Methods	Recall		
	$k = 1$	$k = 5$	$k = 10$
PI	0.0996812	0.314535	0.433091
CF	0.139306	0.393401	0.482963
ItemRank	0.13076	0.385579	0.520077
Context-boosted HPEM	0.167058	0.435717	0.547276

Table 5.5: Recall at top k for each recommendation method.

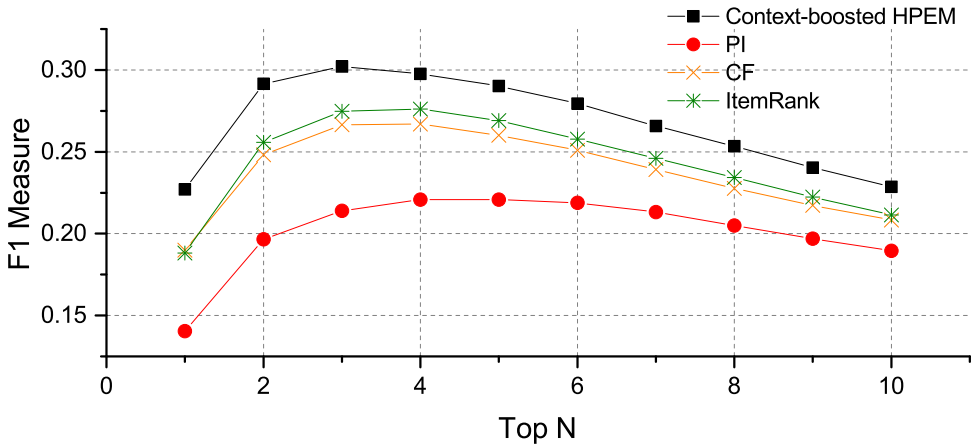


Figure 5.10: F1-measure at top k for each recommendation method.

associated to a particular context. In addition, the proposed model is able to find hidden items that are relevant to given contexts by ranking those items in higher positions in the list of recommended items. We continued to evaluate the algorithms' performances with respect to the F1 and MAP measures. In Figure 5.10, we showed only the F1 values up to $k = 10$. The MAP results are shown in Figure 5.11. The MAP values represent the average precision at top k results [108]. The two measurements clearly show that the Context-boosted HPEM model has the potential to provide considerably accurate context-boosted recommendations, compared to the baseline methods.

After conducting the experiments, we noticed that in our dataset, there are users who are active and rate different items in a variety of contexts, and other users who are less active and rate very few items. As explained in Section 5.1.1, we divided the users into three groups for testing: active users (Active), normal users (Normal), and passive users (Passive). Using MAP, we measure the performance of the three methods using different types of users, as shown in Table 5.6. The results show that all the recommendation methods are sensitive to the number of items consumed by a user, as the recommendation accuracy increases when increasing the number of items consumed by a user. However, our model achieved better MAP values overall, when compared to the baseline methods. It also shows that even though the user might not have enough history that can be used in the recommendation process, our model could indeed a achieve better performance.

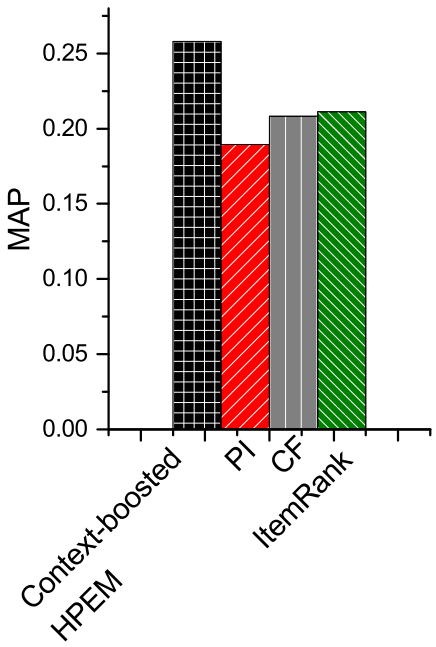


Figure 5.11: MAP at top k recommendation results.

	Condition	$cn < 3$	$cn >= 3$ and $cn < 7$	$cn >= 7$
	Group	Passive	Normal	Active
Recommendation Methods	PI	0.1459	0.1657	0.1781
	CF	0.1757	0.1817	0.1889
	ItemRANK	0.1657	0.2315	0.2999
	Context-boosted HPEM	0.2006	0.2653	0.2903

Table 5.6: MAP at top k according to a variation of the number of items consumed by a user in a context. .

Number of users $\lvert U\rvert$	Number of items $\lvert I\rvert$	Item-context assignments $\lvert I\rvert \times \lvert C\rvert$
2100	5310	996815

Table 5.7: Statistics of the last.fm dataset used to evaluate the individual vs. group recommendations.

5.3.3 Evaluating Individual vs. Group Recommendations Using HPEM with PLSA

To evaluate the recommendation performance of the proposed model given a set of values, we conducted an offline experiment on online crawled datasets from last.fm and MovieLens. We measured the sensitivity of our proposed HPEM model to the user's context, and the accuracy of the resulted recommendations by comparing them to a scenario where the user's context was neglected. We tested the accuracy of the recommendations after running the PLSA to construct the matrices $\mathbf{B}_{|U|\times|C|}$ and $\mathbf{A}_{|C|\times|I|}$, as illustrated in Section 4.6.

We first calculated precision by changing the number of recommended items k from 1 to 10. Table 5.7 shows a description of the last.fm dataset used, in addition to the MovieLens dataset.

Individual Recommendations

For each user in the test set, we validated the top-k recommended items and used precision and recall to examine the ability of the proposed model (as described in Section 4.4.3) to suggest a relevant item according to the rating values. Figures 5.12 and 5.13 depict the precision and recall curves, showing how our context-based method outperformed the method without context, at the selected top-k positions. Note that the algorithm described in Section 4.4.3 utilizes the two similarity matrices $\mathbf{Q}_{|C|\times|C|}$ and $\mathbf{E}_{|I|\times|I|}$. These matrices are built using two approaches: a frequency function, and a binary function as described in Section 4.2. Accordingly, we performed the experiments twice, once with each approach, for comparisons purposes. In Figures 5.12-5.19, the number of recommended items is plotted on the graph curves using data points; the first point of each curve refers to the case of the top-1, whereas the last point is the case of the top-10 recommendations. The reason for using different values of k is that the recommendation

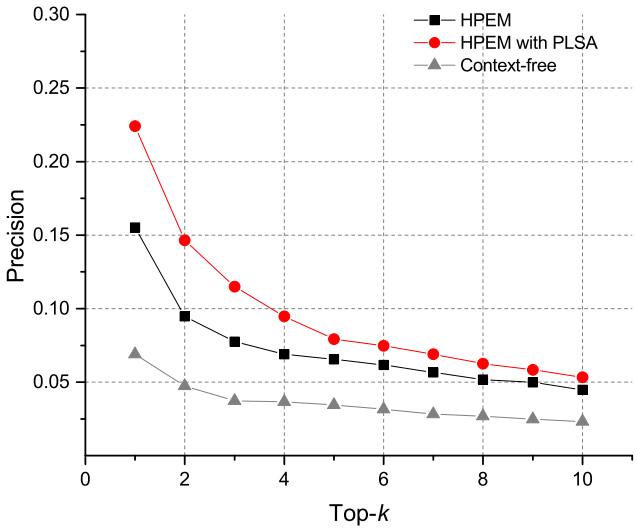


Figure 5.12: Precision results obtained by the proposed HPEM model using the frequency function on the last.fm dataset.

accuracy is affected by the number of items retrieved, as noted in a number of related studies. The results in Figures 5.12 and 5.13 show that the use of the user's context contributed to the improvement of the precision rate by 8.62%, and of recall by 1.73% at top-1, compared to when context was not considered. Based on the results obtained from the experiment on the last.fm dataset, we found that considering the user's context increases the number of relevant items that are placed at the top of a recommendation list, and thus can help tailor the list based on a user's personal preferences. As shown in Figures 5.14 and 5.15, there is a significant improvement in the performance of the algorithm when the frequency function is used to calculate the similarities, compared to when the binary function is used.

We continued our evaluation of the HPEM model on the MovieLens dataset and analyzed its performance when not considering context. We also presented the performance of the HPEM model after applying the PLSA algorithm, as shown in Figures 5.16-5.19.

Group Recommendations

The HPEM model needs to personalize the recommendations for a group of users and not just for an individual. For instance, the RecMirror application is used to recommend a movie to a number of family members who are sitting together in front of the TV

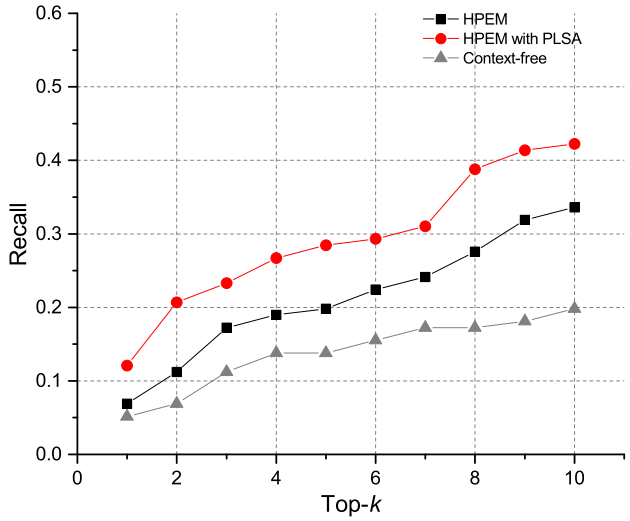


Figure 5.13: Recall results obtained by the proposed HPEM model using the frequency function on the last.fm dataset.

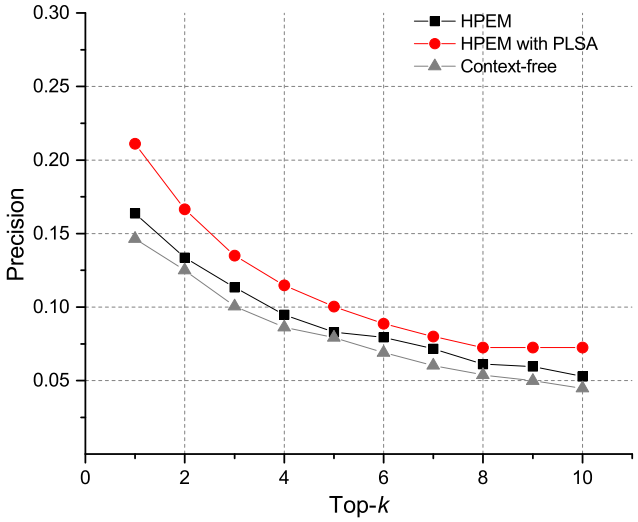


Figure 5.14: Precision results obtained by the proposed HPEM model using the binary function on the last.fm dataset.

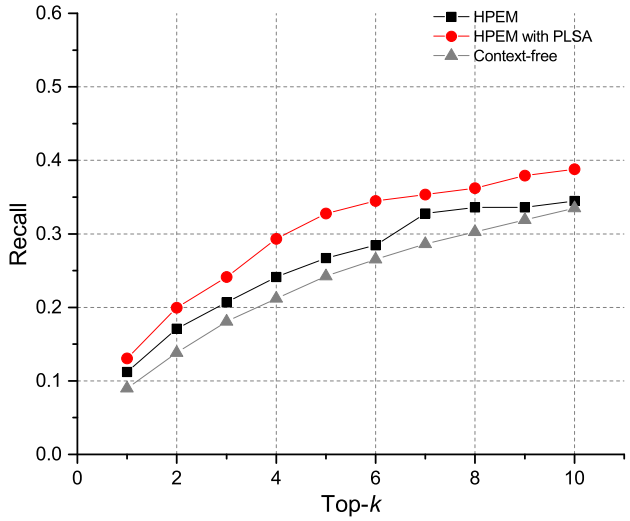


Figure 5.15: Recall results obtained by the proposed HPEM model using the binary function on the last.fm dataset.

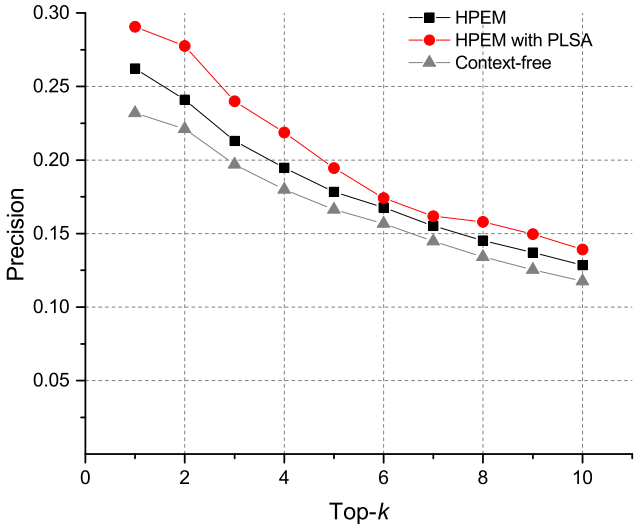


Figure 5.16: Precision results obtained by the proposed HPEM model using the frequency function on the MovieLens dataset.

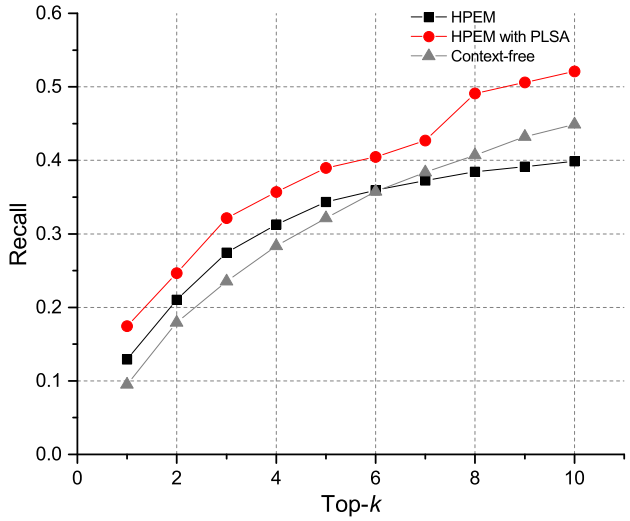


Figure 5.17: Recall results obtained by the proposed HPEM model using the frequency function on the MovieLens dataset.

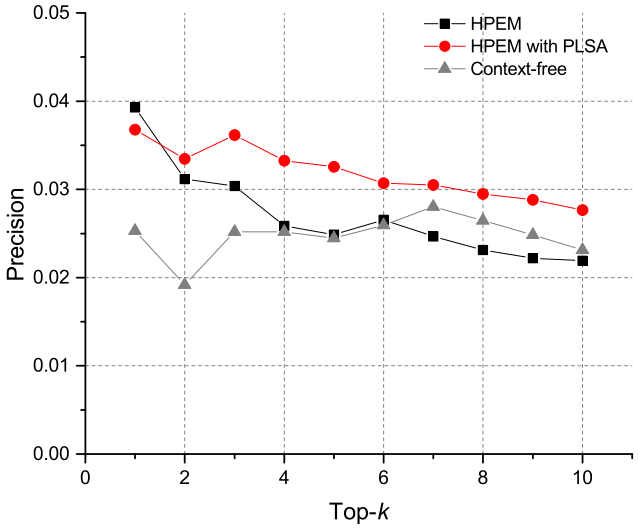


Figure 5.18: Precision results obtained by the proposed HPEM model using the binary function on the MovieLens dataset.

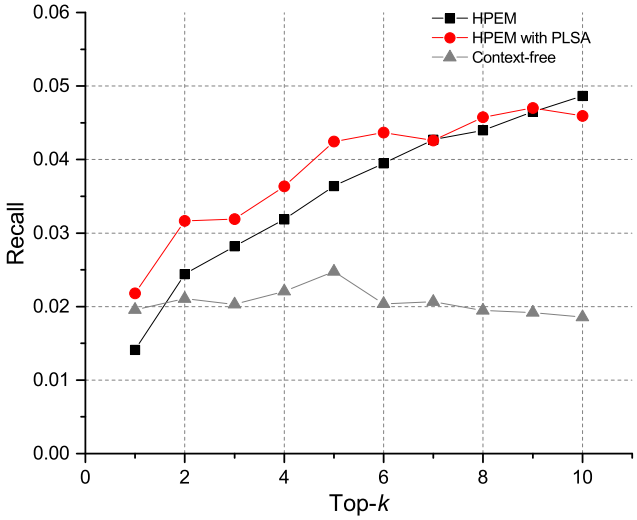


Figure 5.19: Recall results obtained by the proposed HPEM model using the binary function on the MovieLens dataset.

[5]. Hence, we measure the model's capability of generating relevant items to a group of users. In order to divide the dataset in our experiment to form different groups, we follow a similar procedure as the one proposed in [50]. We group the users in each dataset into similar, dissimilar and random groups. Note that each group type contains a different number of users, but that each group within that group type is composed of four users. We set the size of each group to four members, assuming that we need to recommend a movie or music to four users at a time. The distribution of users into groups is based on the similarity values they obtain. In fact, the cosine similarity computed in the early steps of the model is used for each pair of users in the dataset. In order to define the similarity that categorizes the users, we computed the average of the computed similarities in the user-user similarity matrix. The average of the similarity obtained was 0.168 on last.fm, and 0.171 on MovieLens. Users who have a similarity value greater than 0.28 form the similar group. Users whose similarity values are less than 0.28 form the dissimilar group. Users in the random group are free from the similarity restriction and are selected randomly. We evaluated the performance of group recommendations using the MAP obtained with the proposed HPEM model.

As with the procedure used in the individual recommendations, we withheld one item that the four users in a single group shared and added it to the test set. Accordingly, the

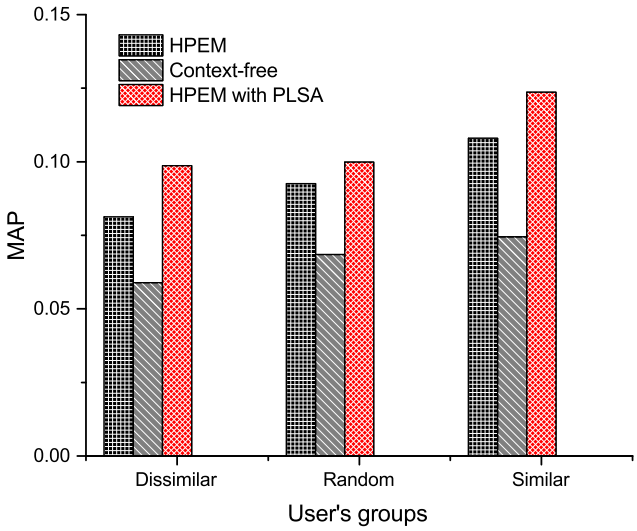


Figure 5.20: MAP obtained by the proposed HPEM model for group-based recommendations on the last.fm dataset.

mean average precision gives the average of how often the withheld item in the dataset appears on the recommended list. As shown in Figures 5.20 and 5.21, the PLSA algorithm has positive impacts on HPEM's performance in its ability to recommend more relevant items for a group, regardless of their similarity, differences, and group type. When we consider the accuracy of the recommendations for the dissimilar users, we notice that the lower the similarity values between members of one group are, the worse the accuracy of the resulted recommendations become. In fact, this was initially expected, since low similarity values indicate how difficult it will be to find items that satisfy all users in the group. Additionally, on both datasets, the recommendations retrieved are affected by the types of users who form the group. We also noticed during the experiment that the type of user had an impact on the resulting MAP values. Specifically, the user might belong to a category of users known as cold-start users if they do not have enough ratings that can be used to help the recommendation algorithm build the necessary associations between them, the items, and their contexts. Therefore, all users employed in this experiment (group recommendation) must have rated at least twenty items. This way, we ensure that the population of items in each user's profile is sufficient for the system to focus on the algorithm performance for the groups rather than for the types of users.

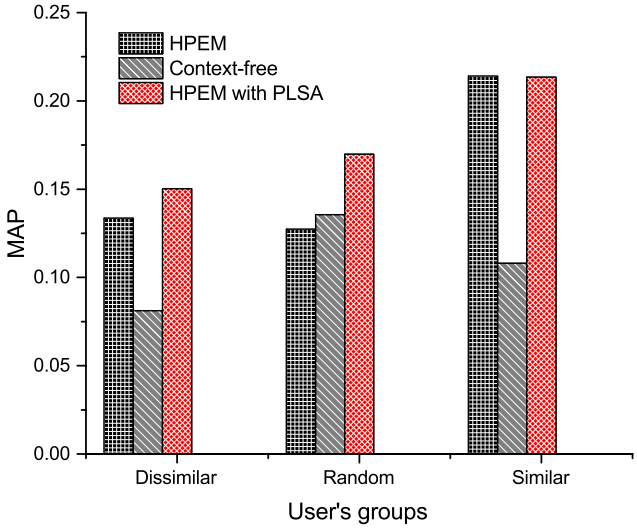


Figure 5.21: MAP obtained by the proposed HPEM model for group-based recommendations on the Movielens dataset.

5.4 Evaluating the Context-Item-Boosted HPEM Model

5.4.1 Parameter Tuning

The proposed recommendation model uses α, an attenuation factor, where $\alpha \in (0, ..., 1)$ is used to reduce the weight factor of the contextual effects on the prediction values. Prior to starting the experiments, we gave α equal values in all contexts in order to run an empirical study. By tuning this parameter, we may increase or decrease the influence of the context on the final scoring value given to an item. Hence, it is critical to correctly set the value of α to optimize the recommendation performance.

We first measure the MAP and the MRR using different values of α on the Last.fm dataset (as described in Table 5.8) and then on the Movielens dataset, as described in Section 5.1.3. The results in Table 5.9 show the best MAP and MRR values obtained when $\alpha = 0.3$ on the Last.fm dataset and $\alpha = 0.5$ on the Movielens dataset.

5.4.2 The Effect of Normalization

Prior to running the comparison experiments, we investigated the impact of matrix normalization on the evaluation metrics. We measured MAP and MRR after running the

Number of users $\|U\|$	Number of items $\|I\|$	Contextual tags $\|C\|$	Item-context assignments $\|I\| \times \|C\|$
164	626	251	10711

Table 5.8: Statistics of the last.fm dataset used to evaluate the Context-item-boosted HPEM model.

Dataset	α	Number of Items MAP@10	MRR@20
Last.fm	0.1	0.120 ± 0.007	0.214 ± 0.008
	0.3	$\mathbf{0.225 \pm 0.016}$	$\mathbf{0.584 \pm 0.004}$
	0.5	0.153 ± 0.004	0.412 ± 0.008
Movielens	0.1	0.131 ± 0.008	0.177 ± 0.007
	0.3	0.106 ± 0.011	0.325 ± 0.016
	0.5	$\mathbf{0.152 \pm 0.011}$	$\mathbf{0.369 \pm 0.09}$

Table 5.9: The different weights assigned to α to measure its sensitivity in each dataset.

model on normalized and non-normalized matrices. Specifically, we employed two different versions of the context-user matrix $\mathbf{T}_{|C|\times|U|}$ and context-item matrix $\mathbf{A}_{|C|\times|I|}$. Tables 5.10 and 5.11 report the improvement on MAP and MRR using normalized matrices over non-normalized matrices of $\mathbf{T}_{|C|\times|U|}$ and $\mathbf{A}_{|C|\times|I|}$, for each dataset. In addition, we conducted a statistical analysis to measure the significance of the improvement of the normalized approach over the non-normalized one. Specifically, we conducted the two-tailed paired t-tests under the same conditions of $\mathbf{T}_{|C|\times|U|}$ and $\mathbf{A}_{|C|\times|I|}$ and for the same dataset. The results confirmed that the normalized approach had a positive impact on both MAP and MRR values.

5.4.3 Offline Experiment

Comparison with Other Methods

In our evaluation, we present the detailed experimental results of the proposed Context-item-boosted HPEM rating method in comparison with other benchmark methods. In addition to the PI and ItemRank described in Section 5.3.1 and Section 5.3.2, the other

	Non-Normalization		Normalization	
T, **A**	MAP@10	MRR@10	MAP@10	MRR@10
25	0.118 ± 0.009	0.288 ± 0.001	$0.129 \pm 0.007^{**}$	$0.294 \pm 0.008^{**}$
50	0.125 ± 0.008	0.311 ± 0.010	$0.138 \pm 0.008^{**}$	$0.299 \pm 0.009^{*}$
100	0.166 ± 0.019	0.385 ± 0.007	$0.149 \pm 0.007^{*}$	$0.302 \pm 0.004^{**}$
150	0.179 ± 0.004	0.402 ± 0.008	$0.169 \pm 0.016^{**}$	$0.335 \pm 0.008^{*}$
All	0.186 ± 0.002	0.422 ± 0.001	$0.199 \pm 0.016^{**}$	$0.579 \pm 0.004^{**}$

*Significant at $p < 0.01$, **Significant at $p < 0.001$

Table 5.10: Effect of normalization on the Last.fm dataset.

	Non-Normalization		Normalization	
T, **A**	MAP@10	MRR@10	MAP@10	MRR@10
25	0.0915 ± 0.003	0.158 ± 0.002	$0.159 \pm 0.001^{*}$	$0.169 \pm 0.001^{*}$
50	0.119 ± 0.004	0.160 ± 0.009	$0.174 \pm 0.001^{**}$	$0.197 \pm 0.007^{*}$
100	0.152 ± 0.008	0.177 ± 0.007	$0.198 \pm 0.003^{*}$	$0.201 \pm 0.006^{*}$
150	0.189 ± 0.006	0.225 ± 0.0012	$0.239 \pm 0.001^{*}$	$0.267 \pm 0.003^{**}$
All	0.202 ± 0.002	0.281 ± 0.014	$0.286 \pm 0.001^{**}$	$0.318 \pm 0.014^{**}$

*Significant at $p < 0.05$, **Significant at $p < 0.01$

Table 5.11: Effect of normalization on the Movielens dataset.

methods used in the comparison are:

- **uMender**: This recommendation technique is proposed by Su et al. [94]. The algorithm first creates a sub-matrix to find users and items that are similar in the same context. Then the algorithm obtains the positive and the negative preferences, based on the available rating values. Finally, the algorithm finds frequencies in the negative and positive item sets and computes the related user-item prediction.

- **Collaborative and Content-based Technique (CCbT)**: Lops et al. [69] proposed a collaborative content-based tag recommendation algorithm. Since we are dealing with context by utilizing social tags, we considered comparing this technique with our proposed model since tag annotation is also analyzed here.

Experimental Results

In this section, we present the results of the comparison between the performances of the Context-item-boosted HPEM model and of the recommendation techniques introduced in Section 5.4.3. As explained in Section 5.1, we computed the performance of each recommendation approach in retrieving accurately relevant items, as well as their ranking positions in the recommendation list. Firstly, we evaluated the recommendation performance by calculating precision and recall, obtained by our model and by the other four approaches, on the Last.fm and Movielens datasets, as in Figures 5.22 and 5.23.

Figure 5.22 and Figure 5.23 depict the precision-recall curves, showing how the Context-item-boosted HPEM model outperforms the other baseline algorithms on both datasets. The number of retrieved items for a user's quest is plotted on data points on the graph curves; the curves start from the left, denoting the top k=1, and the last point on the right denotes the top k=10. The proposed Context-item-boosted HPEM model achieved approximately 2%, 3%, 10%, and 12% precision improvement on the Last.fm dataset, when compared to CCbT, uMender, ItemRank, and PI respectively. Our model also achieved approximately 3%, 1%, 3%, and 6% precision improvement on the Movielens dataset, when compared to CCbT, uMender, ItemRank, and PI respectively.

When comparing the results, we observed that finding the hidden assignments of context to items and hidden relations of users towards contexts reveals more relevant items than non-context-aware approaches. We also noticed that our model is somewhat affected by the number of context parameters attached to an item, which indicates that the more users consume an item in different contexts, the better the recommendation.

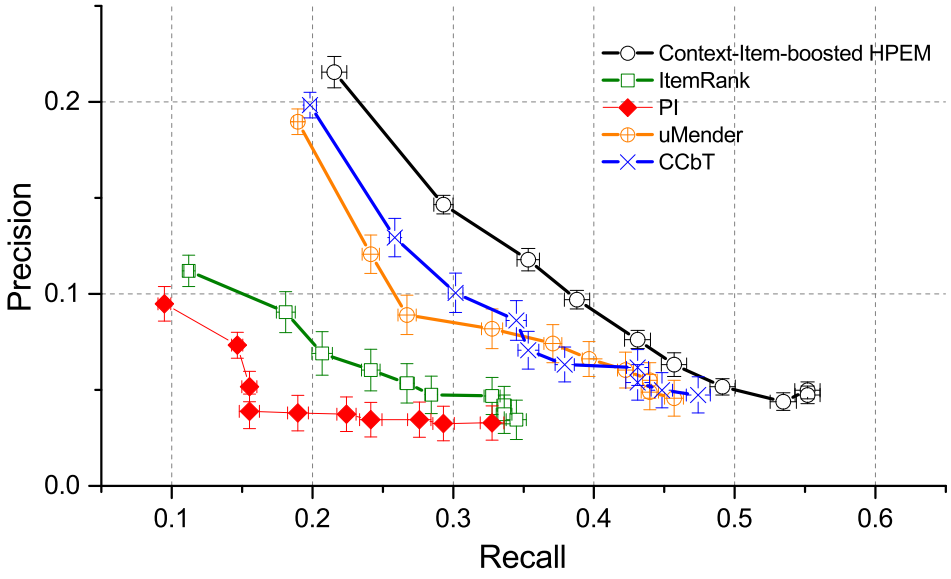


Figure 5.22: Precision and recall obtained by the 5 algorithms on the Last.fm dataset.

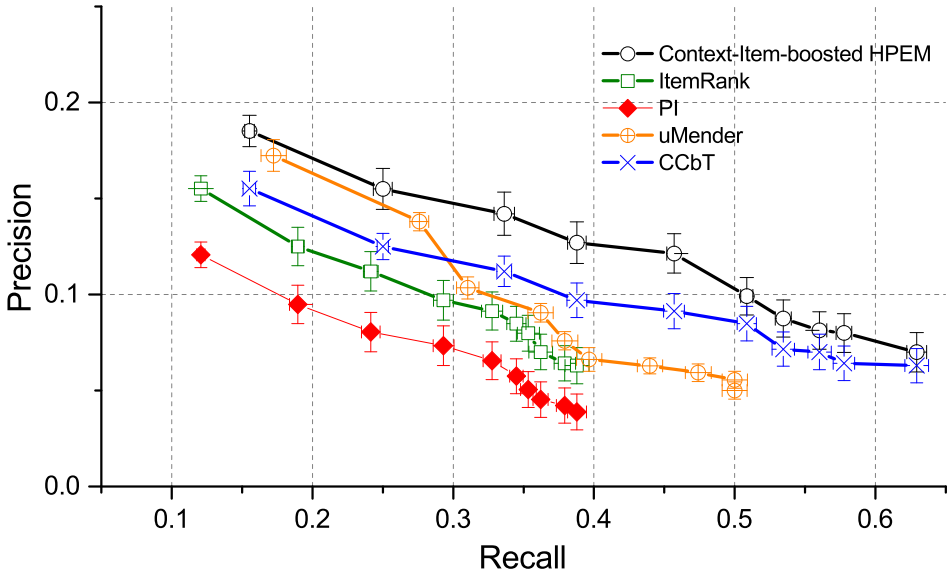


Figure 5.23: Precision and recall obtained by the 5 algorithms on the Movielens dataset.

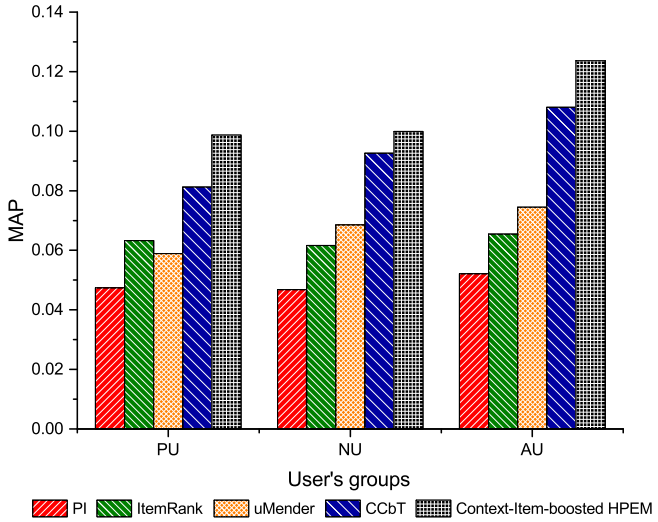


Figure 5.24: A comparison of MAP according a varying number of items rated by a user in the Last.fm dataset.

Since in both datasets there are users who only rated a few items in different contexts, as well as other users who rated many items, we investigated the size of each user's rating history, which is used for context discovery. As introduced in Section 5.1.1, we divided the users in the datasets into three different groups denoted as AU, NU, and PU. If a user rated at least 11 items or more, they are considered to be active users, if they rated from five to ten items, they are considered normal users. Passive users or (cold start users) are those who rated less than 5 items. The resulted performances for each recommendation approach in the two datasets are shown in Figure 5.24 and Figure 5.25. As we expected, all the algorithms were sensitive to the number of items rated by each user. Note also that all algorithms achieved considerably lower MAP for the cold-start problem, due to the fact that the algorithms did not have enough history to feed the recommendation process. However, detecting the user's context indeed facilitates the improvement of the recommendations in such cold start cases.

5.4.4 Subjective Evaluation

To provide insight into the performance of the Context-item-boosted HPEM model on real users, we conducted an online experiment on invited subjects. According to the experimental setup introduced in Section 5.1.2, we invited 15 subjects to participate in

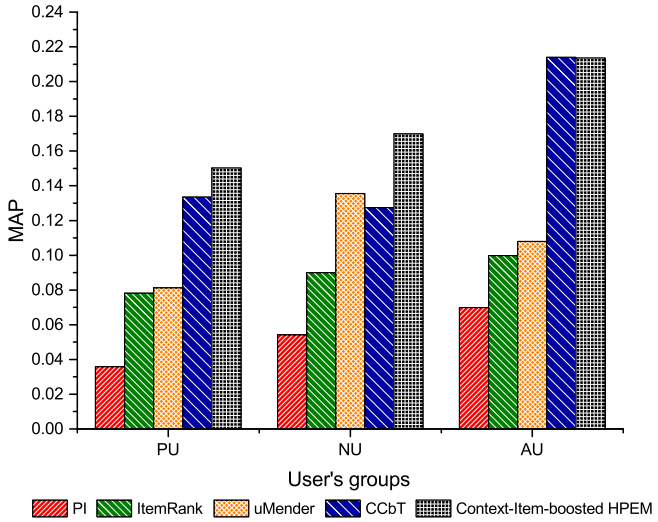


Figure 5.25: A comparison of MAP according a varying number of items rated by a user in the Movielens dataset.

evaluating our context-aware recommender prototype application. The subjects were adult, 7 males and 8 females, with an average age of 23.2 years. To eliminate the effects of other contextual parameters such as the user's age, education level, mother tongue, and culture, we carefully selected participants, for those parameters that are not included as a context in our application. The experiment was performed outside the laboratory and each user was given an android phone to use. All phones needed access to the Internet in order to save the data collected on our server cloud. We used a smaller online crawled dataset, which contained 40 popular artists and 419 musical tracks. Then, we asked the subjects to rate at least 20 tracks and tag each track with some contextual tags, according to their preferences. From these surveys, we obtained 343 item ratings and 1747 context-item associations. Each subject was asked to listen to a track for one minute and then tag it with one or words. We followed the same evaluation protocol that was used in similar studies such as the one by Cai et al. [15]. The rating and contextual data collected are used as the ground truth dataset for the proposed model to generate a recommendation list for a given subject. Afterwards, for each user, we detected six different scenarios of context and ran the algorithm to produce 10 recommendations of items suitable for that particular context. Three of the scenarios reflect the three physiological context dimensions introduced in Section 3.2. We again asked the subjects to perform three types of activities (perform a Stroop color-word test,

sit comfortably and try to read an article, and sit comfortably, close your eyes, and listen to relaxing music). After performing each activity, subjects were asked to provide their feedback about the recommendation list. As for the remaining aspects of the three other contextual scenarios, the application detects them based on the information available as well as the information provided by the user. For instance, date, time, and weather are immediately detected by the android application, while the user has to specify if he/she is alone, with a partner or with a family member. Subjects can optionally select a contextual condition such as: studying for an exam, having a romantic date, etc. Based on the information collected, the application randomly chooses three different contexts as test queries for the evaluation.

Users could listen to each recommended item and provide feedback as to whether they liked or disliked that music in such a context. As a result, we received 900 responses telling us whether a user u_x liked or disliked the retrieved item i_y in a given context c_z for each recommendation algorithm. We then computed the average precision of a user, obtained from the 6 different contextual scenarios. The results are briefly summarized in Figure 5.26. In addition, we conducted a statistical analysis to measure the significance of the improvement of our model using the two-tailed paired t-test. The proposed Context-item-boosted HPEM model achieved a statistical significant of 1%, where $p < 0.01$ over the non-context-aware approach.

We further analyzed the subjects' evaluation of each recommendation list in regard to the three physiological contextual queries. We ran a two-tailed paired t-test on the responses of subjects in relaxed, stressed and neutral situations. Tests on both relaxed and neutral conditions showed that users have certain preferences for these two conditions. However, there were no significant differences with the results obtained in the neutral condition test. Hence, we conclude that other contextual information should be included in the query when the physiological condition of the subject does not differ from their normal benchmarks.

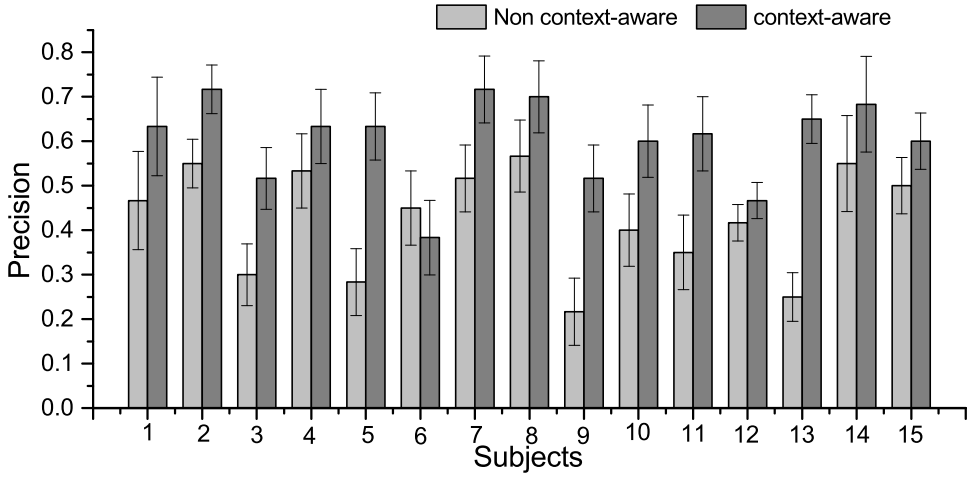


Figure 5.26: Average precision of recommendation retrieval.

5.5 Evaluating the Optimized Hidden Preferences Explorer Model

5.5.1 Comparison with Other Methods

In this section, we present the performance of the optimized version of the proposed HPEM model, as well as the results of the comparison with other recommendation techniques. In addition to uMender and CCbT described in Section 5.4.3, the other method used in the comparison is:

- **Joint Matrix Factorization with Mood Specific (JMF-MS)**: JMF-MS [91] is similar to our approach, where a matrix factorization function with optimization is proposed for mining movies for mood-specific tags. Although the intention of the authors was to focus on the mood tags to run the recommendations, we found their approach to be quite relevant for our work. Note that the dataset used in [91] is "Moviepilot mood track", which is a larger dataset than the one we used in this thesis. Unfortunately, such a dataset is not publicly available.

We also include in our comparison the basic recommendation approach of prediction weight to items with a higher count value, within a specific context (PI) described in Section 5.3.1. Additionally, we implemented the four algorithms mentioned, to the best of our knowledge, based on the algorithm descriptions in the published papers. The dataset used in this experiment is described in Table 5.12.

Number of users $\|U\|$	Number of items $\|I\|$	Contextual tags $\|C\|$	Item-context assignments $\|I\| \times \|C\|$
397	1298	746	17036

Table 5.12: Statistics of the last.fm dataset used to evaluate the Optimized HPEM model.

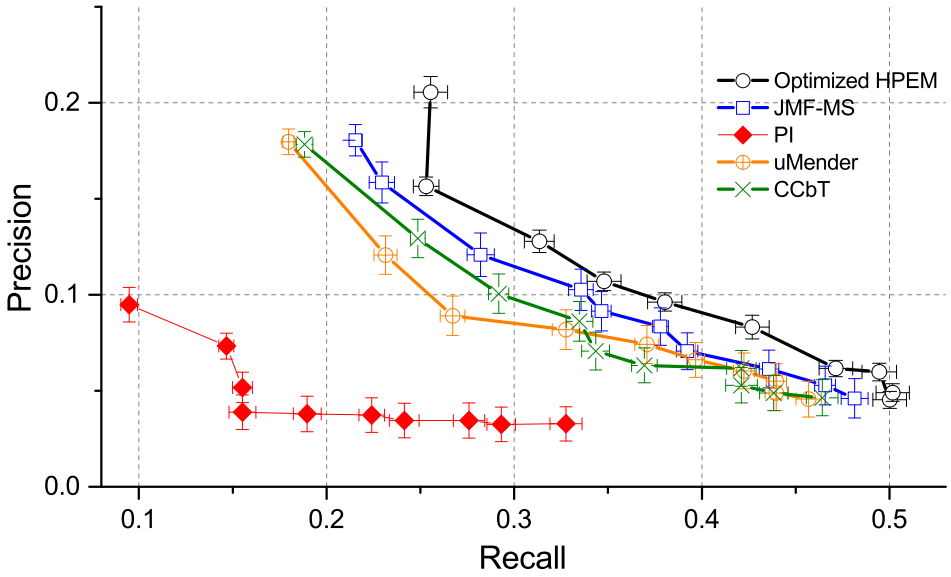


Figure 5.27: Precision and recall obtained from the five algorithms on the Last.fm dataset.

5.5.2 Experimental Results

The following experimental results show the performances obtained by each recommendation approach when retrieving accurately relevant items, as well as their rank in the recommendation list.

First, we evaluated the recommendation performance by calculating precision and recall, obtained by the optimized HPEM model and the other four alternative approaches, on Last.fm, as in Figure 5.27. The precision and recall curves shown in Figure 5.27 represent the performance of each algorithm used in the experiment, as well as the standard deviations of x and y. The points on the graph represent the precision and recall obtained by varying the number of returned items (top-k), where the most left point in each curve represents the precision and recall results at top-1, while the right most point represents the results at top-10. The graph shows that the Optimized HPEM

	CCbT	uMender	PI	JMF-MS	Optimized HPEM
$k = 1$	18.32%	17.97%	9.48%	19.65%	22.78%
$k = 10$	8.44%	8.31%	5.96%	8.40%	8.93%

Table 5.13: F1-measure at top-1 and top-10 for each recommendation method.

model that uses the hidden preferences model along with the result of the optimization function outperforms the other algorithms. The performance of the PI clearly shows that selecting the top used items according to a given context does not lead to identifying relevant items. The precision and recall of uMender, CCbT, JMF-MF, and of our model decreases when the value of k increases. On average, at top-1, our model achieved a 2% precision improvement over JMF-MS, a 3% improvement over uMender and CCbT, and a 11% improvement over PI. Additionally, at top-1, the results obtained from our model also show a recall improvement of 4% over JMF-MS, 7% over CCbT, 8% over uMender, and 16% over PI. We also measured the two-tailed paired t-test to know if there was any statistical significance in the obtained results. Our model achieved statistical significant where $p < 0.01$ over uMender, CCbt, and PI on both precision and recall. Our model was not statistically significant for the precision values obtained, with regards to the JMF-MS. However, our model did achieve a better performance, with statistical significance, on the recall. The result indicates that both our model and JMF-MS are still recommending non-relevant items for the user query, but our model brings more relevant items than JMF-MS.

We continued examining the F1 measure of the algorithm's performance, as shown in Table 5.13. When $k = 1$, the proposed algorithm improved by 3.13%, 4.46%, 4.81%, and 13.30% over JMF-MS, CCbT, uMender, and PI, respectively.

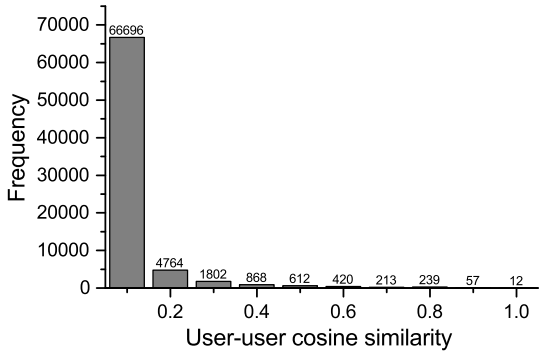


Figure 5.28: The frequency of the cosine similarity between pairs of users

5.6 Computational Analysis

5.6.1 Computing Similarity

To minimize the cost of computing the required offline similarities, we consider only the k most similar neighbors for each similarity computation. Figure 5.28 shows the frequency (number of pairs of users) of the cosine similarities obtained in the dataset described in Table 5.12. We are interested in the long tail of the higher similarities. Accordingly, we eliminate the computed similarities of the users who share few items with others, and assign a zero similarity value if the similar user is not among the top k nearest neighbors. We notice the same pattern when computing the item-item similarities, as in Figure 5.29. For instance, we employ the matrix ($\mathbf{E}$), where $\mathbf{E} = \mathbf{E^k}$ to form the item-item similarity matrix. In addition, we neglect computing the similarities of the same vector entries, i.e. for any two column vectors $sim(v_x, v_y) = 1$, if $x = y$, and they therefore represent the diagonal similarity values in any resulted similarity matrix. Then, only the non-zero entries are used in the process of building the recommendation model. The worst-case computational cost of building the similarity matrices ($\mathbf{S}_{|U|\times|U|}$, $\mathbf{E}_{|I|\times|I|}$, and $\mathbf{Q}_{|C|\times|C|}$) is $O(|U|^2 \times |I|)$, $O(|I|^2 \times |U|)$, and $O(|C|^2 \times |I|)$, respectively.

5.6.2 Finding Hidden Preferences

As when computing the similarity, finding the hidden preferences is done offline in the recommender system. In the worst-case scenario, computing the three hidden preferences models ($\mathbf{UtC}_{|U|\times|C|}$, $\mathbf{CtI}_{|C|\times|I|}$, and $\mathbf{UtI}_{|U|\times|I|}$) requires $O(k|U||C|)$, $O(k'|C||I|)$, and $O(k''|U||I|)$, respectively, where k, k', k'' represents the top similar entities from each

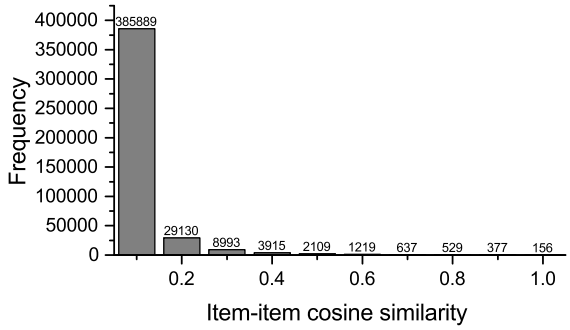


Figure 5.29: The frequency of the cosine similarity between pairs of items

dimension. However, the total cost of building each model includes the cost of this step and the cost of creating the similarity matrix in Section 5.6.1. Hence, computing each model costs $O(|U|^2|I|+k|U||C|)$, $O(|I|^2|U|+k'|C||I|)$, and $O(|C|^2|I|+k''|U||I|)$ for **UtC**, **CtI**, and **UtI** respectively.

Thus we have considered the worst-case scenario for calculating the computational cost. However, previous studies consider such evaluations as overestimating [112, 49, 51]. They experimentally discuss the sparsity between dimensions in such a dataset environment, where the number of contexts associated to an item, as well as the number of contexts in which a user consumes items, are significantly lower than $|I|^2$, $|C|^2$, $|U|^2$. Accordingly, the cost of building the similarity matrices is close to $O(|U||I|)$, $O(|I||U|)$, and $O(|C||I|)$, based on the fact that users consume a small number of items from the whole dataset and few contextual parameters are assigned to the consumption of each item. As a result, the total cost of building **UtC**, **CtI**, and **UtI** hidden preference models are: $O(|U||I| + k|U||C|) \cong O(|U||I| + |U||C|)$, $O(|I||U| + k'|C||I|) \cong O(|I||U| + |C||I|)$, and $O(|C||I| + k''|U||I|) \cong O(|C||I| + |U||I|)$. Accordingly, the total cost at this step becomes $O(|C||I| + |U||I|)$.

5.6.3 Computing the Context-Aware Rating

This part is the only online part of the proposed model. All the steps computed previously can be done offline (once a day, a week, or a month, etc.). When finding the item rating for a given user query, which contains m contextual tags, the required computational cost is $O(m)$, with an average computation time of 0.064 seconds, for a query with three given contexts.

5.6.4 Optimization Complexity

We analyze the computational complexity of the optimization function according to the complexity of the optimization learning for one iteration. Given Equation 4.24, we can obtain a linear complexity of $O(|U||C||T|)$ for one iteration, where $|T|$ is the size of the test set and $T \subset U \times I$. The size of $|T|$ is usually much smaller than the size of the training set $|T| << |U| \times |I|$ to leave the majority of the dataset for algorithm training. Then, in practice, we have a linear complexity in one iteration of $O(|U||C|)$.

The offline, online, and optimization computational analysis described above show the applicability of the proposed model to large-scale applications. In fact, the design of the hidden preferences model does not require the whole model to be completely re-built when new items, users, or contexts are added to the database. The changes in each vector only need to reflect the changes to a particular computation for the newly added information.

5.7 Summary

This chapter provided a methodology for how the proposed versions of the HPEM recommendation models are evaluated. The evaluation is divided into two main parts. The first part is evaluating the effectiveness of the proposed HPEM model, and the second part is evaluating the computational performance. We investigated whether or not the proposed recommendation model improves the accuracy of the item prediction. The goal of the experiments is to evaluate the accuracy of the proposed HPEM model: the utilization of the user's context to recommend a different number of items. The HPEM model experimentally demonstrates favorable advantages in enhancing the accuracy of the prediction and providing suitable items for the user's context. It has also shown an improvement to the cold-start problem, since it uncovers all hidden associations between the user's context and the suitable related items. Also, user satisfaction is qualitatively evaluated.

As an example of an implementation, we introduced two prototype applications: RecMirror, and Biommender. The prototype applications facilitate the personalization of the recommended contents by identifying the user's context through adaptive user interfaces in an AmI environment. Both applications use the multi-modal recommendation interfaces to provide the means of personalizing the delivery of the recommended multimedia contents.

Chapter 6

Conclusion and Future Work

With today's wide use of social media resources and services, Internet users can benefit from the rich information available online through the use of recommender systems. Additionally, sensors attached to smartphones are becoming widely used to support the interactions between the user and the different context-aware recommender systems. Therefore, as demonstrated in this thesis, we propose a multimedia recommendation model. We demonstrate the importance of using contextual information to provide enhanced recommendations and to increase the level of interactions between the user and their preferred multimedia contents. To enhance the user experience and personalize the multimedia services that a user wants to consume, our model collects physiological parameters from the body, such as the ECG signal. Moreover, the model integrates other parameters such as light intensity, temperature, and user detection. The social network and history of choices are also included in the contextual data acquisition for the recommendation.

In this thesis, we also demonstrate a new method of searching for hidden preferences, associated to multiple dimensions of user, item, and context, in a tensor model. In addition, the proposed model identifies the hidden contexts assigned to items and applies collaborative filtering to find hidden context preferences from similar users. The advantage of our proposed recommendation model is that it considers the contextual information by reflecting the social tags available online to explore the hidden contexts assigned to items, as well as by applying CF to find hidden context preferences from similar users. Accordingly, the proposed model can search and rate items without the need to analyze the item's content, such as the music lyrics or the voice signal, to predict the associated context. The experimental results demonstrate that the proposed

context-aware recommendation technique offers favorable advantages in enhancing the accuracy of the prediction and providing suitable items for the user's context. We also address the issue of providing recommendations to a group of users rather than only to individuals.

Lack of available contextual information has become a crucial challenge in the design of any context-based recommendation. Therefore, we present two prototype applications that collect contextual data explicitly and implicitly. By building such an application, we are able to collect the required contextual information and enhance the recommendation performance.

As for future work, we are working on the use of different measurements to compute the similarity within a singular dimension. Specifically, we are interested in analyzing the effects of context on the search for similarities between users and items. In our experiments, we only study the model performance using the cosine similarity, but for the extension of this work, we are examining the effect of other similarity methods such as the one proposed in [63]. Additionally, in our experiments, we optimize the proposed HPEM model by optimizing the MAP measure, but we could study the effects of optimizing other quality measures and their effects on the overall performance.

We plan to pursue our work in order to evaluate the recommendation performance using group of contexts. Specifically, we plan to recommend items for users according to a group of similar and dissimilar contexts. We are also interested in studying the use of adaptive rating, where the context of the user plays a role in defining the relation between users and items. We also like to target items in the long tail of recommendation by utilizing context. This issue exists due to the fact that there are items that have low frequency of use. The proposed recommendation model can also be extended for different types of resources such as books, or news. The identification mechanism of individual context parameters and the evaluation of such parameters influences the user's final selection would be valuable to develop further, in the extension of this thesis.

Appendix A

Optimization Algorithm

Algorithm 1 Optimizing the HPEM model.

procedure OPTIMIZATION($\mathbf{UtI}_{|U|\times|I|}$, $\mathbf{CtI}_{|C|\times|I|}$, $\mathbf{UtC}_{|U|\times|C|}$, $\mathbf{R}_{|U|\times|I|}$, Test set ($Test$))
Output: The learned α, β, MAP
    initialization rating=(α,β);
    initialization $t = 0$;
    initialization $MAP_0 = findPrecision(\alpha = 1, \beta = 1, Test)$; ▷ based on Equation 4.21
    repeat
        for `u=1:`$|U|$ **do**
            for `c=1:`$|C|$ **do**
                for `i=1:`$|I|$ **do**
                    $rating1 = findCtI(c, i)$; ▷ based on Equation 4.19
                    $rating2 = findUtI(u, i)$; ▷ based on Equation 4.19
                    relevance=$(\alpha\, rating1) \times (\beta\, rating2)$;
                end for
            end for
        end for
        $t = t + 1$;
        $MAP = findPrecision(\alpha, \beta, Test)$ ▷ based on Equation 4.23

.

```
        if MAP − MAP_0 ≤ 0 then
            break;
        end if
    until t ≥ maximumIter
end procedure
```

Appendix B

The PLSA Algorithm

Algorithm 2 PLSA based HPEM model of computing the user and context probability $P(c|u)$.

1: **procedure** DOPLSA($\mathbf{R}_{|U|\times|I|}$)
2: **Output:** The observed list of contexts for each user order by rating score $sumZ_{|U|\times|C|}$.
3: initialization $iterations$, $|Z|$, $prob_{c|C|\times|Z|}$, $prob_c out_{|C|\times|Z|}$, $prob_{u|Z|\times|U|}$, $prob_u out_{|Z|\times|U|}$, sum,$sumZ_{|U|\times|C|}$;
4: randomArrayValuesContexts(); ▷ updating the $prob_c$ matrix with random values.
5: randomArrayValuesUsers(); ▷ updating the $prob_u$ matrix with random values.
6: **for** $u = 1 : u < |U|$ **do** ▷ updating the $SumZ$ matrix
7: **for** $c = 1 : c < |C|$ **do**
8: **for** $z = 1 : |Z|$ **do**
9: $sum+ = prob_c(c, z) \times prob_u(z, u)$;
10: **end for**
11: $sumZ(u, c) = sum$;
12: **end for**
13: **end for**
14: **for** $k = 1 : iterations$ **do** ▷ starting the PLSA algorithm.

.

```
15:     for z = 1 : |Z| do                ▷ calculating p(c|z) according to Equation 4.32
16:         sum=0;
17:         for u = 1:|U| do                                   ▷ calculating p(z|c, u)
18:             for c = 1:|C| do     ▷ calculating p(z|u, c) according to Equation 4.30
19:                 Pz_uc = prob_c(c, z) × prob_u(z, u);
20:                 Pz_uc = Pz_uc/sumZ(u, c);          ▷ calculating the p(z|u, c) is now
    complete.                          ▷ We need to calculate the p(z|u) as in Equation 4.32
21:                 prob_cout(c, z)+ = Pz_uc;      ▷ summation of all p(z|c, u) for specific
    user, all contexts
22:                 prob_uout(z, u)+ = Pz_uc;        ▷ all users, all contexts for specific z
23:                 sum+ = Pz_uc;            ▷ sum all probabilities of contexts, for users
24:             end for
25:         end for
26:         updatePLSA();▷ The calculation is complete for a single z variable for all
    users, updating the prob_cout and prob_uout matrices is required now.
27:     end for
28:     for c = 1 : c < |C| do
29:         for z = 1 : z < |Z| do
30:             prob_c(c, z) = prob_cout(c, z);
31:             prob_cout(c, z) = 0;
32:         end for
33:     end for
34:     for z = 1 : z < |Z| do
35:         for u = 1 : u < |U| do
36:             prob_u(z, u) = prob_uout(z, u);
37:             prob_uout(z, u) = 0;
```

.

```
38:             end for
39:         end for
40:         sum=0;
41:         for u = 1 : |U| do      ▷ calculating the log likelihood based on Equation 4.28
42:             for c = 1 : |C| do
43:                 sum+ = log(sumZ(z, u);          ▷ Σ_u Σ_c log(Σ_z P(c|z) P(z|i)) as in
    Equation 4.28
44:             end for
45:         end for
46:     end for
47: end procedure
```

Appendix C

Biommender Application

Figure C.1: Interaction diagram of a user requesting a song using the Biommender application.

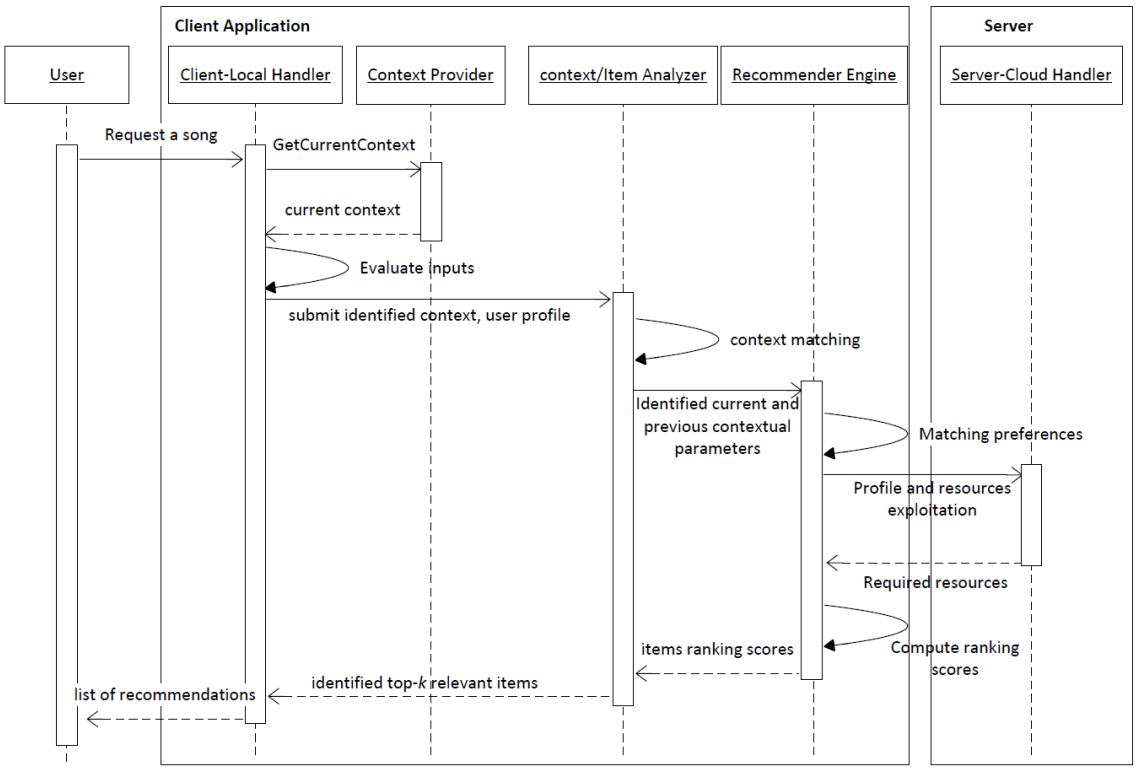
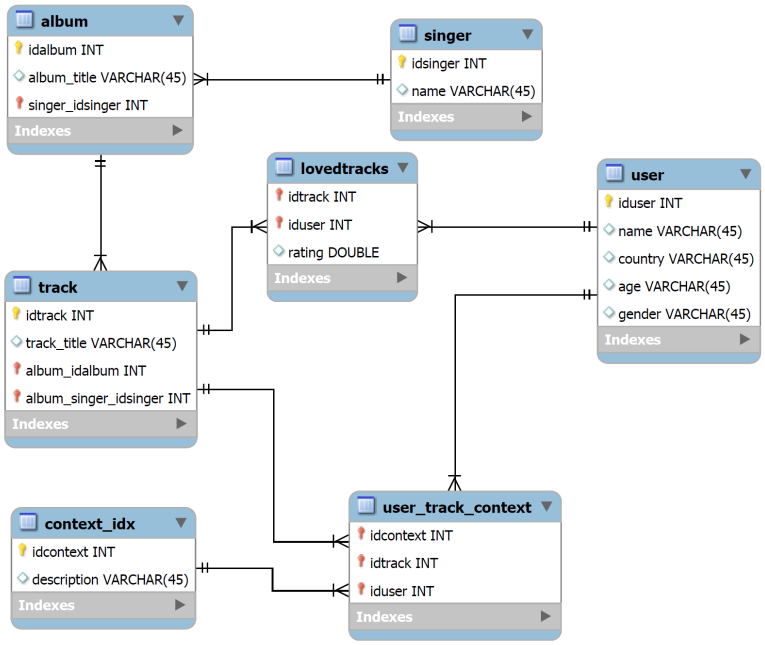


Figure C.2: Client-Local Resources Layer Database ERD.



Bibliography

[1] Gediminas Adomavicius, Ramesh Sankaranarayanan, Shahana Sen, and Alexander Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23(1):103–145, 2005.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Context-Aware Recommender Systems. In *Recommender Systems Handbook*, pages 217–253. 2011.

[4] Deepak Agarwal, Bee-Chung Chen, and Bo Long. Localized factor models for multi-context recommendation. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, c:609, 2011.

[5] Mohammed F. Alhamid, Majdi Rawashdeh, and Abdulmotaleb El Saddik. Towards context-aware recommendations of multimedia in an ambient intelligence environment. *Proceedings of the 9th workshop on Workshop on Multimedia Information Processing and Retrieval*, 2013.

[6] Flora Amato, Angelo Chianese, and Vincenzo Moscato. SNOPS: a smart environment for cultural heritage applications. *Proceedings of the 12th International Workshop on Web Information and Data Management*, pages 49–56, 2012.

[7] Onur Asan, Mark Omernick, Dain Peer, and Enid Montague. Designing a better morning: a study on large scale touch interface design. *Human-Computer Interaction. Interaction Techniques and Environments, Lecture Notes in Computer Science*, 6762:13–22, 2011.

[8] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke, and Roland Schwaiger. Incarmusic: Context-aware music recommendations in a car. In *E-Commerce and Web Technologies, Lecture Notes in Business Information Processing*, pages 89–100. 2011.

[9] Linas Baltrunas and Francesco Ricci. Context-dependent items generation in collaborative filtering. *Proceedings of the Workshop on Context-Aware Recommender Systems*, 2009.

[10] Luciano Bernardi, Cesare Porta, and Peter Sleight. Cardiovascular, cerebrovascular, and respiratory changes induced by different types of music in musicians and non-musicians: the importance of silence. *Heart (British Cardiac Society)*, 92(4):445–52, 2006.

[11] Luciano Bernardi, Joanna Wdowczyk-Szulc, Cinzia Valenti, Stefano Castoldi, Claudio Passino, Giammario Spadacini, and Pete Sleight. Effects of controlled breathing, mental activity and mental stress with or without verbalization on heart rate variability. *Journal of the American College of Cardiology*, 35(6):1462–9, 2000.

[12] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 193–202, 2008.

[13] Toine Bogers. Movie recommendation using random walks over the contextual graph. *Proceedings of the 2nd Intl. Workshop on Context-Aware Recommender Systems.*, 2010.

[14] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artifcial Intelligence*, pages 43–52, 1998.

[15] Rui Cai, Chao Zhang, Chong Wang, Lei Zhang, and Wei-ying Ma. MusicSense: Contextual Music Recommendation using Emotional Allocation Modeling. *Proceedings of the 15th International Conference on Multimedia*, pages 553–556, 2007.

[16] Luigi Ceccaroni and Xavier Verdaguer. Magical Mirror: multimedia, interactive services in home automation. *Proceedings of Workshop on Environments for Personalized Information Access*, pages 10–21, 2004.

[17] Annie Chen. Context-aware collaborative filtering system: Predicting the user's preference in the ubiquitous computing environment. *Location-and Context-Awareness*, pages 1110–1111, 2005.

[18] Jun-Ming Chen, Meng-Chang Chen, and Yeali S. Sun. A tag based learning approach to knowledge acquisition for constructing prior knowledge and enhancing student reading comprehension. *Computers & Education*, 70(0):256 – 268, 2014.

[19] Joon Yeon Choi, Hee Seok Song, and Soung Hie Kim. MCORE: a context-sensitive recommendation system for the mobile Web. *Expert Systems*, 24(1):34–36, 2007.

[20] Roberto Colombo, G. Mazzuero, F. Soffiantino, M. Ardizzoia, and G. Minuco. A comprehensive PC solution to heart rate variability analysis in mental stress. *Computers in Cardiology 1989, Proceedings*, pages 475–478, 1989.

[21] Diane J Cook, Juan C Augusto, and Vikramaditya R Jakkula. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, 2009.

[22] Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Miguel a. Rueda-Morales. Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*, 51(7):785–799, 2010.

[23] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[24] Mukund Deshpande and George Karypis. Item-based top- N recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

[25] Anind Dey, Gregory Abowd, and Daniel Salber. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, 16(2):97–166, 2001.

[26] Anind K Dey. Context-aware computing: The CyberDesk project. *Proceedings of the Spring Symposium on Intelligent Environments*, 1998.

[27] Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. Scenarios for ambient intelligence in 2010. *Tech. report, Information Society Technologies Advisory Group, Institute of Prospective Technological Studies*, 2001.

[28] Niels Egelund. Spectral analysis of heart rate variability as an indicator of driver fatigue. *Ergonomics*, 25(7):663–72, 1982.

[29] Greg T Elliott and Bill Tomlinson. PersonalSoundtrack : Context-aware playlists that adapt to user pace. *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, pages 736–741, 2006.

[30] Marco Gori, Augusto Pucci, Via Roma, and I Siena. Itemrank: A random-walk based scoring algorithm for recommender engines. *Proceedings the 20th International Joint Conference on Artificial Intelligences*, pages 2766–2771, 2007.

[31] Byeong-jun Han, Seungmin Rho, Sanghoon Jun, and Eenjun Hwang. Music emotion classification and context-based music recommendation. *Multimedia Tools and Applications*, 47(3):433–460, 2009.

[32] Taher Haveliwala, Sepandar Kamvar, and Glen Jeh. An Analytical Comparison of Approaches to Personalizing PageRank. 2003.

[33] Jennifer A Healey and Rosalind W Picard. Detecting Stress During Real-World Driving Tasks Using Physiological Sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, 2005.

[34] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999.

[35] Jonathan L. Herlocker, Joseph a. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[36] Nis Hjortskov, Dag Rissén, Anne Katrine Blangsted, Nils Fallentin, Ulf Lundberg, and Karen Sø gaard. The effect of mental stress on heart rate variability and blood pressure during computer work. *European journal of applied physiology*, 92(1-2):84–9, 2004.

[37] Thomas Hofmann. Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[38] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[39] M. Anwar Hossain, Pradeep K. Atrey, and Abdulmotaleb El Saddik. Gain-based Selection of Ambient Media Services in Pervasive Environments. *Mobile Networks and Applications*, 13(6):599–613, October 2008.

[40] M. Shamim Hossain, S. K. Alamgir Hossain, Atif Alamri, and M. Anwar Hossain. Ant-based service selection framework for a smart home monitoring environment. *Multimedia Tools and Applications*, pages 1–21, 2012.

[41] Yajie Hu. *A Music Recommendation System Based on User Behaviors and Genre Classification*. PhD thesis, University of Miami, 2012.

[42] Yajie Hu and M Ogihara. Nextone player: A music recommendation system based on user behavior. *Proceedings of 12th International Society for Music Information Retirval Conference*, pages 103–108, 2011.

[43] Ziwon Hyung, Kibeom Lee, and Kyogu Lee. Music recommendation using text analysis on song requests to radio stations. *Expert Systems with Applications*, 2013.

[44] Ziwon Hyung, MA Lee, and K Lee. Music Recommendation Based on Text Mining. *Proceedings of the 2nd International Conference on Advances in Information Mining and Management*, pages 129–134, 2012.

[45] Robert Jäschke, Leandro Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Communications*, 21(4):231–247, 2008.

[46] Emil Jovanov, Amanda O'Donnell Lords, Dejan Raskovic, Paul G Cox, Reza Adhami, and Frank Andrasik. Stress monitoring using a distributed wireless intelligent sensor system. *IEEE Engineering in Medicine and Biology Magazine : the quarterly magazine of the Engineering in Medicine & Biology Society*, 22(3):49–55, 2003.

[47] Alexandros Karatzoglou and Xavier Amatriain. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. *Proceedings of the 4th ACM conference on Recommender Systems.*, 2010.

[48] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. *Proceedings of the 4th ACM conference on Recommender systems*, pages 79–86, 2010.

[49] George Karypis. Evaluation of item-based top-n recommendation algorithms. *Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 247–254, 2001.

[50] Heung-Nam Kim, Mark Bloess, and Abdulmotaleb El Saddik. Folkommender: a group recommender system based on a graph-based ranking algorithm. *Multimedia Systems*, 19(6):509–525, 2012.

[51] Heung-Nam Kim, Majdi Rawashdeh, Abdullah Alghamdi, and Abdulmotaleb El Saddik. Folksonomy-based personalized search and ranking in social media services. *Information Systems*, 37(1):61–76, 2012.

[52] Hyun-Jun Kim and Young Sang Choi. EmoSens: Affective entity scoring, a novel service recommendation framework for mobile platform. *Proceedings of the 5th ACM Conference on Recommender Systems*, 2011.

[53] Jong-Hun Kim, Chang-Woo Song, Kee-Wook Lim, and Jung-Hyun Lee. Design of music recommendation system using context information. In *Agent Computing and Multi-Agent Systems*, pages 708–713. 2006.

[54] Jonghun Kim, Daesung Lee, and Kyung-Yong Chung. Item recommendation based on context-aware model for personalized u-healthcare service. *Multimedia Tools and Applications*, 2011.

[55] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[56] Robert E. Krout. Music listening to facilitate relaxation and promote wellness: Integrated aspects of our neurophysiological responses to music. *The Arts in Psychotherapy*, 34(2):134–141, 2007.

[57] Tatiana Lashina. Intelligent bathroom. In *European Symposium on Ambient Intelligence*, 2004.

[58] PJ Lattimore. Stress-induced eating: an alternative method for inducing ego-threatening stress. *Appetite*, 36(2):187–8, 2001.

[59] Jae Sik Lee and Jin Chun Lee. Context awareness by case-based reasoning in a music recommendation system. In *Ubiquitous Computing Systems*, pages 45–58. 2007.

[60] Sangkeun Lee, Sang-il Song, Minsuk Kahng, Dongjoo Lee, and Sang-goo Lee. Random walk based entity ranking on graph for multidimensional recommendation. *Proceedings of the 5th ACM conference on Recommender Systems*, page 93, 2011.

[61] George Lekakos and Petros Caravelas. A hybrid approach for movie recommendation. *Multimedia Tools and Applications*, 36(1-2):55–70, 2006.

[62] Greg Linden, Brent Smith, and Jeremy York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing*, 7(1):76–80, 2003.

[63] Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian, and Xuzhen Zhu. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56:156–166, 2014.

[64] Hao Liu. *Biosignal Controlled Recommendation in Entertainment Systems*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2010.

[65] Hao Liu, Jun Hu, and Matthias Rauterberg. Music playlist recommendation based on user heartbeat and music preference. *Proceedings of the International Conference on Computer Technology and Development*, pages 545–549, 2009.

[66] Hao Liu, Jun Hu, and Matthias Rauterberg. Software architecture support for biofeedback based in-flight music systems. *Proceedings of the 2nd International Conference on Computer Science and Information Technology*, pages 580–584, 2009.

[67] Hao Liu, Jun Hu, and Matthias Rauterberg. iheartrate: a heart rate controlled in-flight music recommendation system. *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*, page 26, 2010.

[68] Ning-Han Liu, Szu-Wei Lai, Chien-Yi Chen, and Shu-Ju Hsieh. Adaptive music recommendation based on user behavior in time slot. *International Journal of Computer Science and Network Security*, 9(2):219–227, 2009.

[69] Pasquale Lops, Marco de Gemmis, Giovanni Semeraro, Cataldo Musto, and Fedelucio Narducci. Content-based and collaborative techniques for tag recommendation: an empirical evaluation. *Journal of Intelligent Information Systems*, 40(1):41–61, 2012.

[70] Marek Malik, J Thomas Bigger, A John Camm, Robert E Kleiger, Alberto Malliani, Arthur J Moss, and Peter J Schwartz. Heart rate variability standards of measurement, physiological interpretation, and clinical use. *European heart journal*, 17(3):354–381, 1996.

[71] Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, and Gerd Stumme. Evaluating similarity measures for emergent semantics of social tagging. pages 641–650, 2009.

[72] Cédric S Mesnage, Asma Rafiq, Simon Dixon, and Romain P Brixtel. Music discovery with social networks. *Proceedings of the Workshop on Music Recommendation and Discovery*, 793:1–6, 2011.

[73] Moeiz Miraoui, Chakib Tadj, Jaouhar Fattahi, and Chokri Ben Amar. Dynamic context-aware and limited resources-aware service adaptation for pervasive computing. *Advances in Software Engineering*, page 7, 2011.

[74] F Narducci, W J Snape, W M Battle, R L London, and S Cohen. Increased colonic motility during exposure to a stressful situation. *Digestive Diseases and Sciences*, 30(1):40–4, 1985.

[75] Fatma Nasoz, Christine L. Lisetti, and Athanasios V. Vasilakos. Affectively intelligent and adaptive car interfaces. *Information Sciences*, 180(20):3817–3836, 2010.

[76] Shahriar Nirjon, Robert F Dickerson, Qiang Li, Philip Asare, John A Stankovic, Dezhi Hong, Ben Zhang, Xiaofan Jiang, Guobin Shen, and Feng Zhao. MusicalHeart: A Hearty Way of Listening to Music. *ACM, Sensys*, 2012.

[77] José M. Noguera, Manuel J. Barranco, Rafael J. Segura, and Luis Martínez. A Mobile 3D-GIS Hybrid Recommender System for Tourism A Mobile 3D-GIS Hybrid Recommender System for Tourism. *Information Sciences*, 215:37–52, 2012.

[78] Nuria Oliver and F Flores-Mangas. MPTrain: a mobile, music and physiology-based personal trainer. *Proceedings of the 8th Conference Human-Comput. Interact. Mobile Devices Services*, 2006.

[79] Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In *Fuzzy Systems and Knowledge Discovery*, pages 970–979. 2006.

[80] Toon Pessemier, Simon Dooms, and Luc Martens. Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications*, 2013.

[81] Paulo Pombinho, Maria Beatriz Carmo, and Ana Paula Afonso. Context aware point of interest adaptive recommendation. *Proceedings of the 2nd Workshop on Context-awareness in Retrieval and Recommendation*, pages 30–33, 2012.

[82] George Popescu and P Pu. What's the best music you have? designing music recommendation for group enjoyment in groupfun. *Proceedings of the ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, pages 1673–1678, 2012.

[83] Rani Qumsiyeh and Yiu-Kai Ng. Predicting the ratings of multimedia items for making personalized recommendations. *Proceedings of the 35th International ACM SIGIR conference on Research and Development in Information Retrieval*, page 475, 2012.

[84] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 635–644, 2011.

[85] Gordon Reynolds, Dan Barry, Ted Burke, and Eugene Coyle. Interacting with large music collections: Towards the use of environmental metadata. *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 989–992, 2008.

[86] Lizawati Salahuddin, Jaegeol Cho, Myeong Gi Jeong, and Desok Kim. Ultra short term analysis of heart rate variability for monitoring mental stress in mobile settings. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4656–9, 2007.

[87] Markus Schedl, Sebastian Stober, Emilia Gomez, Nicola Orio, and Cynthia C. S. Liem. User-Aware Music Retrieval. In *Multimodal Music Processing*, volume 3, pages 135–156. 2012.

[88] Yue Shi, Alexandros Karatzoglou, and Linas Baltrunas. TFMAP: Optimizing MAP for top-n context-aware recommendation. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012.

[89] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Alan Hanjalic, and Nuria Oliver. Tfmap: optimizing map for top-n context-aware recommendation. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 155–164, 2012.

[90] Yue Shi, Martha Larson, and Alan Hanjalic. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pages 34–40, 2010.

[91] Yue Shi, Martha Larson, and Alan Hanjalic. Mining contextual movie similarity with matrix factorization for context-aware recommendation. *ACM Transactions on Intelligent Systems and Technology*, 4(1):1–19, 2013.

[92] Dongmin Shin, Jae-won Lee, Jongheum Yeon, and Sang-goo Lee. Context-Aware Recommendation by Aggregating User Context. *Proceedings of the IEEE Conference on Commerce and Enterprise Computing*, pages 423–430, 2009.

[93] George W. Snedecor and William G. Cochran. *Statistical Methods*. Iowa State University Press, 7 edition, 1980.

[94] Ja-Hwung Su, Hsin-Ho Yeh, Philip S Yu, and Vincent S Tseng. Music recommendation using content and context information mining. *Intelligent Systems*, 25(1):16–26, 2010.

[95] Tiffany Y Tang, Pinata Winoto, and Gordon McCalla. Further thoughts on context-aware paper recommendations for education. In *Recommender Systems for Technology Enhanced Learning*, pages 159–173. Springer, 2014.

[96] Jian Tian, Kening Gao, Yin Zhang, and Bin Zhang. Improving search by extending tags according to recommendation level and combinations of types. pages 36–43, 2011.

[97] JHM Tulen, P Moleman, HG Van Steenis, and F Boomsma. Characterization of stress reactions to the Stroop Color Word Test. *Pharmacology, biochemistry, and behavior*, 32(1):9–15, 1989.

[98] Cornelis van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.

[99] Emil Šiška. The Stroop colour-word test in psychology and biomedicine. *Universitatis Palackianae Olomucensis. Gymnica*, 32(1), 2002.

[100] Xinxi Wang, David Rosenblum, and Ye Wang. Context-Aware Mobile Music Recommendation for Daily Activities. *Proceedings of ACM Multimedia*, 2012.

[101] Sung-Shun Weng, Binshan Lin, and Wen-Tien Chen. Using contextual information and multidimensional approach for recommendation. *Expert Systems with Applications*, 36(2):1268–1279, 2009.

[102] Wolfgang Woerndl, Christian Schueller, Rolf Wojtech, and Unternehmertum Gmbh. A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications. *Proceedings of the 23rd International Conference on Data Engineering Workshop*, pages 871–878, 2007.

[103] Wan-Shiou Yang, Hung-Chi Cheng, and Jia-Ben Dia. A location-aware recommender system for mobile shopping environments. *Expert Systems with Applications*, 34(1):437–445, 2008.

[104] Ashkan Yazdani, Evangelos Skodras, Nikolaos Fakotakis, and Touradj Ebrahimi. Multimedia content analysis for emotional characterization of music video clips. *EURASIP Journal on Image and Video Processing*, 2013(1):26, 2013.

[105] Kam Fung Yeung. *A context-aware framework for personalized recommendation in mobile environments*. PhD thesis, University of Portsmouth, Portsmouth, United Kngdom, 2011.

[106] Kyoungro Yoon, Jonghyung Lee, and Min-Uk Kim. Music recommendation system using emotion triggering low-level features. *IEEE Transactions on Consumer Electronics*, 58(2):612–618, 2012.

[107] Kyoungro Yoon, Jonghyung Lee, and Min-Uk Kim. Music recommendation system using emotion triggering low-level features. *IEEE Transactions on Consumer Electronics*, 58(2):612–618, 2012.

[108] Kuifei Yu, Baoxian Zhang, and Hengshu Zhu. Towards Personalized Context-Aware Recommendation by Mining Context Logs. *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, 7301:431–443, 2012.

[109] Kuifei Yu, Baoxian Zhang, Hengshu Zhu, Huanhuan Cao, and Jilei Tian. Towards personalized context-aware recommendation by mining context logs through topic models. In *Advances in Knowledge Discovery and Data Mining*, pages 431–443. 2012.

[110] Zhiwen Yu, Xingshe Zhou, Daqing Zhang, Chung-Yau Chin, Xiaohang Wang, and Ji Men. Supporting context-aware media recommendations for smart phones. *Pervasive Computing*, 5(3):68–75, 2006.

[111] Zhang Yujie and Wang Licai. Some challenges for context-aware recommender systems. *Proceedings of the 5th International Conference on Computer Science & Education*, pages 362–365, 2010.

[112] Valentina Zanardi and L Capra. Social ranking: uncovering relevant content using tag-based recommender systems. *Proceedings of the ACM Conference on Recommender Systems*, pages 51–58, 2008.

[113] Eva Zangerle, Wolfgang Gassler, and Günther Specht. Exploiting Twitter's Collective Knowledge for Music Recommendations. *Proceedings of the 2nd Workshop on Making Sense of Microposts*, pages 14–17, 2012.

[114] Weishi Zhang, Guiguang Ding, Li Chen, Chunping Li, and Chengbo Zhang. Generating virtual ratings from chinese reviews to augment online recommendations. *ACM Transactions on Intelligent Systems and Technology*, 4(1):9:1–9:17, February 2013.

[115] Lijuan Zhou, Hongfei Lin, and Cathal Gurrin. Emir: a novel music retrieval system for mobile devices incorporating analysis of user emotion. In *Advances in Multimedia Modeling*, pages 627–629. 2012.