LATENT PROBABILISTIC MODEL FOR CONTEXT-AWARE RECOMMENDATION

Mr. Padipat Sitkrongwong

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science and Information Technology

Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University

Academic Year 2014

ตัวแบบความน่าจะเป็นแฝงสำหรับการแนะนำที่ตระหนักถึงบริบท

นายปฏิพัทธ์ ศิษย์ครองวงษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557

Thesis Title	LATENT PROBABILISTIC MODEL FOR CONTEXT-AWARE RECOMMENDATION
By	Mr. Padipat Sitkrongwong
Field of Study	Computer Science and Information Technology
Thesis Advisor	Assistant Professor Saranya Maneeroj, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

....Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr.rer.nat)

THESIS COMMITTEE

....Chairman

(Professor Chidchanok Lursinsap, Ph.D.)

....Thesis Advisor

(Assistant Professor Saranya Maneeroj, Ph.D.)

....External Examiner

(Atchara Mahaweerawat, Ph.D.)

ปฏิพัทธ์ ศิษย์ครองวงษ์ : ตัวแบบความน่าจะเป็นแฝงสำหรับการแนะนำที่ตระหนักถึงบริบท (LATENT PROBABILISTIC MODEL FOR CONTEXT-AWARE RECOMMENDATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.ศรันญา มณีโรจน์, 61 หน้า.

ระบบแนะนำ เป็นเครื่องมือที่ถูกคิดค้นขึ้นเพื่อทำการแนะนำสินค้าหรือผลิตภัณฑ์ ที่มีความสอดคล้องหรือตรงต่อความชอบส่วนบุคคลให้แก่ผู้ใช้แต่ละคนในระบบ อย่างไรก็ตาม วิธีการส่วนใหญ่ที่ใช้ในระบบการแนะนำนั้นมักจะไม่คำถึงสภาวะบริบทเช่น สถานที่ เวลา หรือสภาพอากาศ ที่อาจมีผลกระทบต่อระดับความพึงพอใจของผู้ใช้ที่มีต่อสินค้าได้ ด้วยเหตุนี้ ระบบแนะนำที่ตระหนักถึงบริบทจึงถูกคิดค้นขึ้น เพื่อที่จะทำการแนะนำโดยใช้ข้อมูลค่าความพึงใจที่ผู้ใช้ได้ให้ไว้ในสภาวะบริบทที่แตกต่างกัน เนื่องจากการนำข้อมูลทางบริบทที่มีอยู่ทั้งหมดมาใช้อาจทำให้เกิดปัญหาข้อมูลบางเบา รวมทั้งส่งผลกระทบต่อความแม่นยำของการทำนายค่า วิธีการส่วนใหญ่จึงมุ่งเน้นไปที่การตรวจสอบและเลือกใช้เฉพาะข้อมูลทางบริบทที่เกี่ยวข้องมาใช้ในการสร้างตัวแบบ อย่างไรก็ตาม นอกเหนือจากความแม่นยำในการทำนายค่าแล้ว ความหลากหลายของการแนะนำยังมีส่วนสำคัญที่จะช่วยเพิ่มความพึงพอใจของผู้ใช้ต่อผลการแนะนำ มากไปกว่านั้น วิธีการแนะนำที่ตระหนักถึงบริบทส่วนใหญ่ไม่ได้คำนึงถึงความสัมพันธ์ระหว่างบริบท ผู้ใช้ และสินค้า ก่อนที่จะทำการทำนายค่าระดับความพึงพอใจ ในความเป็นจริงนั้น ปัจจัยทางบริบทที่แตกต่างกันอาจส่งผลกระทบต่อผู้ใช้และสินค้าแตกต่างกันไปด้วย งานวิจัยนี้ได้นำเสนอตัวแบบความน่าจะเป็นแฝงสำหรับการแนะนำที่ตระหนักถึงบริบท โดยการดัดแปลงตัวแบบผสมยืดหยุ่นให้รองรับการพิจารณาถึงข้อมูลทางบริบท ซึ่งตัวแบบนี้ทำการตรวจสอบปัจจัยทางบริบทที่เกี่ยวข้องกับประเภทของผู้ใช้และประเภทของสินค้า โดยการผสานตัวแบบผสมยืดหยุ่นของเบย์เข้ากับขั้นตอนวิธีหาค่าเหมาะสมที่สุดแบบกลุ่มอนุภาค โดยการหาค่าที่ดีที่สุดของตัวแบบนั้นแบ่งได้เป็นสองกรณี คือ กรณีที่พิจารณาถึงความแม่นยำของตัวแบบเพียงอย่างเดียว และกรณีที่พิจารณาถึงส่วนได้ส่วนเสียระหว่างความแม่นยำและความหลากหลายของการแนะนำ ซึ่งผลการทดลองได้แสดงให้เห็นว่า ตัวแบบที่เสนอนี้มีประสิทธิภาพในการแนะนำมากกว่า 1) ตัวแบบที่ไม่มีการพิจารณาถึงข้อมูลทางบริบท 2) ตัวแบบที่มีการพิจารณาถึงข้อมูลทางบริบททั้งหมดโดยไม่มีการเลือก 3) ตัวแบบที่มีการพิจารณาถึงความสัมพันธ์ของบริบทกับผู้ใช้หรือสินค้าเพียงอย่างเดียว และ 4) วิธีการของระบบแนะนำที่ตระหนักถึงบริบทในปัจจุบัน

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต	....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก	....
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ		
ปีการศึกษา	2557		

5672603223 : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

KEYWORDS: MACHINE LEARNING / CONTEXT-AWARE / RECOMMENDER SYSTEMS

PADIPAT SITKRONGWONG: LATENT PROBABILISTIC MODEL FOR CONTEXT-AWARE RECOMMENDATION. ADVISOR: ASST. PROF. SARANYA MANEEROJ, Ph.D., 61 pp.

Recommender Systems are software tools that provide personalized recommendations of relevant items to individual users. However, most of them do not take into account additional contextual information that may affect user preferences, such as place, time, or weather. Context-aware recommender systems have been proposed to solve this problem by providing the better recommendations for users based on their rating history in different situations. Since incorporating all contextual information makes the data become sparser and degrades the prediction accuracy, most context-aware methods focus on identifying and applying the relevant contextual variables into the models. However, besides the accuracy, the diversity of the recommendation is also the key to improve the users' satisfaction on the recommended results. Moreover, most context-aware techniques have not directly considered the relationships among context, users, and items before predicting the ratings. In the real world, different contextual factors tend to affect users and items differently. This work proposes a latent probabilistic model for contextual recommendation by extending the flexible mixture model to incorporate the contextual information. Combining with the binary particle swarm optimization techniques, the relevant contextual factors to the user classes and item classes are identified and incorporated into the model. The proposed model is optimized with two cases: considering only the accuracy, and considering the trade-off between accuracy and diversity. The evaluation shows that the proposed model performs better than 1) the traditional model-based techniques that do not consider contextual information. 2) the model that considers only the relations of context to users alone or items alone, 3) the model that exploits all contextual factors, and 4) the traditional context-aware recommendation method.

Department:	Mathematics and Computer Science	Student's Signature	....
		Advisor's Signature	....
Field of Study:	Computer Science and Information Technology		
Academic Year:	2014		

ACKNOWLEDGEMENTS

First and foremost, I have to thank my thesis advisor, Asst. Prof. Dr. Saranya Maneeroj. Without her assistance and dedicated involvement in every step throughout the process, this work would have never been accomplished.

I would also like to show gratitude to my committee, including Prof. Dr. Chidchanok Lursinsap, and Dr. Atchara Mahaweerawat, who have participated my dissertation. They raised many precious points in our discussion and I hope that I will be able to improve my works as their suggestions as much as possible.

Finally, I also want to thank my friends and my family for their sincerely and consistently supports all over the time since I started my dissertation.

CONTENTS

Page

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I
INTRODUCTION

Recommender Systems (RS) are software tools invented to reduce users' information overload by providing personalized recommendations. By using a user's past preference data (i.e. the rating) given to the specified items, the main task of RS is to provide the list of the unseen items that match to his personal interest.

Usually, most techniques in RS can be classified into two main strategies: Content-based Filtering (CBF) and Collaborative Filtering (CF). CBF makes a recommendation on the items that match to the profiles created from the user's past preference history. On the other hand, CF recommends the items for the active user using the opinions from his neighbor users who appear to hold the similar interest on items. Both of these strategies are often combined together to make more effective recommendations.

Most of the traditional RS techniques rely only on two types of data for making the recommendations: user and item. However, recent researches on Context-Aware Recommender Systems (CARS) found that the quality of recommendation can be improved by incorporating the contextual information into the model. The contextual factors such as location, companion, mood or weather might also have a strong effect on the users' decision to decide which kind of products they prefer. For example, a user might choose to watch an action movie when he watches with friends at the cinema, while the same user might choose to watch a comedy movie when he watches with family at home. By considering the contextual data in the recommendation process, the prediction results have shown to be more accurate than the standard RS methods that ignored context [2].

One of the common problems in RS is the data sparsity; most users only give a few ratings compared to all possible ratings, and the system may not have enough data to make an effective recommendation. Some methods like the CF-based approaches are not designed to deal with this problem since their predictive performance are depended directly to the number of ratings in the system.

The sparsity problem becomes even worse when it comes to CARS, which considers contextual dimension of data besides user and item. One of the crucial parts of CARS is to identify which of contextual factors should be incorporated into the model. To make an effective recommendation, only the relevant contextual factors to the recommendation should be extracted from the dataset. Using all of the contextual factors not only invoke the sparsity of the data, but also degrade the recommendation quality. This is because using the irrelevant contextual factors that are not related to the objective of the model can significantly affect the prediction accuracy [2], [18].

In order to produce the effective recommendation, the CARS methods should contain two important characteristics. First, the methods should be able to deal with the data sparsity. Also, the methods should consider only the relevant contextual factors in the recommendation process.

Many methods in RS were proposed to solve the sparsity problem, including the Matrix Factorization (MF) [15] and the Flexible Mixture Model (FMM) [23]. The MF is one of the latent-factor models; it characterizes both users and items by latent-factor vectors derived from rating patterns, and uses them for the prediction. The FMM, on the other hand, is a latent probabilistic model which introduces the latent class variables that are used to characterize users and items. The probability distributions of those latent classes are used for the prediction.

Several methods for identifying the relevant of the contextual factors were proposed in CARS. Most of them identify the set of contextual factors that have an effect on the entire rating data using various kinds of statistical testing [2], [5], [18]. For example, Adomavicius et al. proposed the technique called reduction-based Context-Aware Collaborative Filtering (CACF) [2]. This technique extracts only the rating data that match with an active user's contextual situation, and then exploits the standard collaborative filtering to make the prediction. The challenge of this approach is the selection of relevant the contextual factors that will be used to filter the rating data. The relevant contextual factors should be the ones that provide the highest accurate prediction result, while maintaining the acceptable level of coverages. The identification of the relevant contextual factors is usually done by manual, using the opinions of domain experts; or by automated, trying every possible combination of contextual factors and choosing the one with best performance. Notice that for each dataset, only one set of relevant contextual factors is discovered and applied throughout the whole recommendation process.

However, the quality of the recommendation can be improved by identifying and exploiting the set of relevant contextual factors more specifically on each component of RS. For example, [28] proposed the method that identified the set of relevant contextual factors for each part of the algorithm individually. The idea is that each component of the algorithm might have its corresponding context different from the others; by identifying the right one, the overall performance can be enhanced. On the other hand, Baltrunas et al. [4] presented another way to incorporate context by modeling the relationship between contextual features and items. They proposed the Context-Aware Matrix Factorization (CAMF) method by adding the parameters that model the relationship between each item and each contextual feature into the MF. Although CAMF provides good prediction accuracy, it considers only the relationship among context and items. In the real world, context could affect both items themselves and users' selection practices. Therefore, in order to provide better recommendation, it is important to carefully consider the relationships of context on both users and items.

Moreover, most of the methods in RS only consider the prediction accuracy in order to evaluate their performance. However, the recent research found that only good prediction accuracy might not be able to satisfy the user's interest [17]. This is because the high accurate model may recommend only the set of items that can be expected by the users. To improve the variety of the recommendation results, the proposed method is optimized by combining both accuracy and genre diversity. The genre diversity is measured on the variation of genres on the top-N recommendation list, using the proposed fitness function in form of probabilistic formula.

This work is divided into two main parts. In the first part, a novel way to incorporate the contextual features into the probabilistic model by considering their effects to both users and items is proposed [26]. The Expectation and Maximization (EM) [8] is used for the optimization on the movie-rating dataset where the contextual features are synthesis. On the other hand, in the second part, a method for identifying the relevant contextual features to the user classes and item classes by combining the feature selection technique and the probabilistic model is proposed. These factors are the ones that satisfying the objective function of the model: to provide good prediction accuracy, while maintaining the diversity in recommendations. . Such trade-off between the accuracy and diversity is modeled by the proposed fitness function. In contrast to the first part, the optimization is done using the Bayesian estimation [23] on the real context-aware dataset [16]. Since the Bayesian probabilistic model spends a lot of time for the learning procedure, using the exhaustive search to identify the relevant contextual factors from the over-millions combinations of context would not be efficient. Therefore, the Binary Particle Swarm Optimization (BPSO) technique [13] is applied in the proposed model as the feature selection technique. This technique can be helpful to reduce the number of the identifications on the relevant contextual factors within the acceptable number of iterations.

CHAPTER II

THEORETICAL BACKGROUND

2.1 Recommender Systems

The rapidly growth in information technologies and communication systems bring us to the new era, which people can access to the information they seek from internet anywhere and anytime. A lot of companies and service providers use this opportunity to extend their markets by developing the websites as the new choice for customers to access the details about their products. However, many online users are suffered from the information overload problem due to the tremendous selection of products or services provided by the certain kinds of websites. This is because most of the websites try to offer their products as variety as possible, to cover all the needs of the users, without considering the individual tastes of the users. In fact, the interests in products or services among the users are different. Offering the irrelevant products that does not match to the users' preferences might negatively affect their satisfactions on the websites. To solve this, Recommender Systems (RS) was invented. The main role of RS is to make the personalized recommendation that would match the items of interest to an active user, using several kinds of techniques.

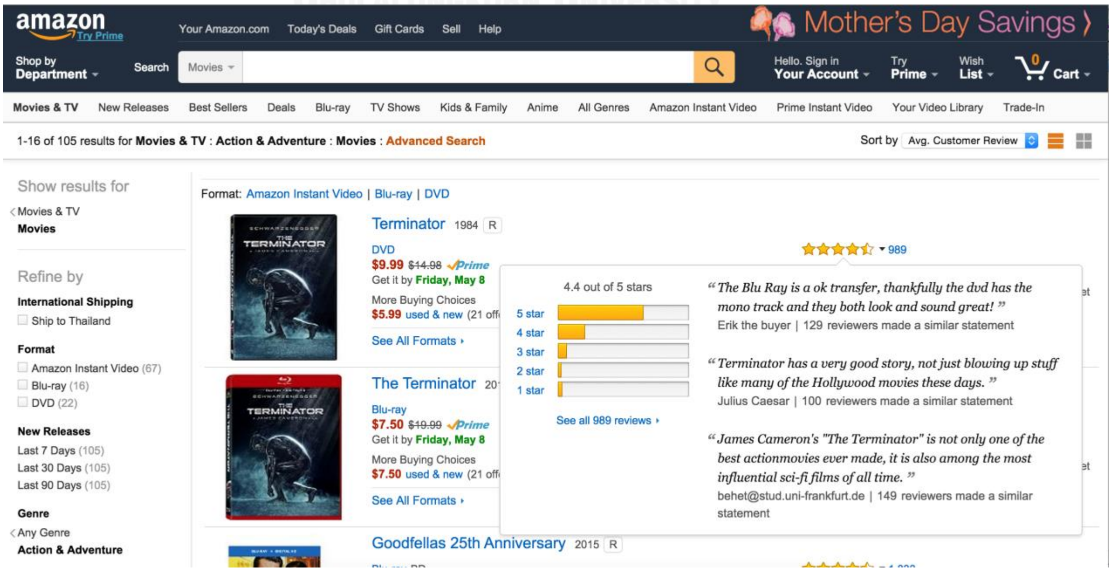


Figure 1. The Ratings Provided by the Users of Amazon.com

Some popular e-commerce websites allow their users to express their preference levels toward items via the *ratings*. Figure 1, for example, shows the ratings given to the movie Terminator, provided by the users of Amazon.com. The members of this website can write the reviews and give the ratings to the items they are interested in. These rating values usually are given in numeric scale, for example from 1 to 5. The higher the rating means the users have high preference level toward items. The main objective of most RS methods is to make use of these ratings, along with the other related features, to predict the ratings on those items that have not been rated by the users before.

In order to predict the users' ratings toward items, first, the system collects users' ratings for items from the sources of data, e.g. websites. For easily use, these ratings are usually stored in the user-item rating matrix as show by Figure 2. This matrix consists of 2 dimensions: one represents users and the other represents items. The value of each element in the matrix contains a rating that each user provides to each item, for example, User u_1 rates '4' to Item i_3. The empty elements mean that the items have not been rated by the users yet.

Item / User	i_1	i_2	i_3	i_4	i_5
u_1	5		4	4	
u_2	3	1		3	3
u_3		5	5	2	

Figure 2. The User-Item Rating Matrix

After collected, the system uses the rating data to extract the users' rating patterns and match them with the most appropriate items for recommendation. The function for predicting the level of a user's preference for an item can be represented as:

$$F_{User \times Item}: U \times I \rightarrow R$$

where

- U is information about the user,
- I is information about the item,
- R is the rating value, and
- $F_{User\times Item}$ is a two-dimensional (2D) estimation function that uses information about the user and the item to estimate the preference rating.

Up until now, the immense numbers of rating estimation functions were proposed. Most of them can be categorized into two main strategies: Content-based Filtering (CBF) and Collaborative Filtering (CF). Moreover, these two strategies are occasionally combined together to make the better recommendations, which are called Hybrid Recommendation.

2.1.1 Content-based Filtering

The content-based filtering (CBF) makes a recommendation to an active user by matching the items, which holding the certain characteristics, corresponding to his past explicit and/or implicit activities. Usually, CBF consists of these following steps:

1) The system gathers the items' characteristics from their data sources. For movie recommendations, these characteristics are, for example, the movie genres, actors, directors, budgets, as well as the keywords describing the movies.
2) The system creates the user profiles from their explicit and implicit data. The explicit data is the data the users provided to the system directly, for example, the ratings, the demographic data, and the answers from the questionnaires. On the other hand, the implicit data is the data the system acquired from the users by keep tracking their online behaviors. These behaviors are, for example, the items that the users clicked to see their details, the pages they visited, the sections/parts of web UI they moved mouse over. These profiles can represent what kinds of item characteristics each user prefers.

3) Find the relationship between the items and the users using the item characteristics and user profiles created from step 1) and 2). Then, using this relationship to recommend the most relevant set of items to each user.

The advantage of the CBF approach is that, there is no need to use the opinions of the other users to make the recommendation to an active user. Therefore, even there is only one user in the system; the recommendation can still be made. However, creating the effective item characteristics is quite challenging. This is because normally, there are lots of item features available from many sources, so a good selection technique is requires. Selecting and exploiting the irrelevant item features may result in bad quality item characteristics, leading ineffective recommendation. Moreover, since CBF tends to recommend only the items similar to the ones the users have rated in the past, it cannot suggest the unexpected items to the users. For example, if the active user only rated the action movies in his web history, the system will be able to recommend only the movies in the genre of action to this user. In the real world, some users expect the system to suggest the items that are differed from their expectation rather than the ones they can guess, which they may already know but not interested in. The serendipity and novelty play as important role to improve the users' satisfaction on the recommended items [24].

2.2.2 Collaborative Filtering

Unlike the CBF approaches that make the recommendation to an active user using only his past preference data alone, the Collaborative Filtering (CF) based approaches also exploit the preference data from the other users in order to make the recommendation. The system predicts the rating for the active user by using the ratings from the users who are similar to that active user, called neighbors. The neighbors are the users that have similar interests in items with the active user, i.e. rated the same set of items with the similar ratings. Therefore, the ratings provided by the neighbors are reasonably useful to predict the ratings on unseen items for the active user. More specifically, the basic steps of the CF-based approaches are the following:

1) The system collects the rating data from the users.
2) The system identifies the neighbors of each user by calculating the similarity. There are many similarity metrics that can be used to measure the similarities among the users; for example, cosine similarity and Pearson's correlation coefficient [21]. Only those users who are similar to an active user (have similarity values more than the defined threshold) are considered as neighbors of that active user.
3) The rating on the target item for the active user is predicted using the ratings of this item from his neighbors who have rated it before. The similarities among the users are also used as the weights for the prediction.

Since the CF-based approaches exploit the opinions of the other users in the system to make a recommendation, they are able to make a recommendation on the unexpected items for the active user—solving the serendipity problem occurred in CBF-based approaches. However, the prediction made by CF-based might not be effective if the rating data is limited (which is common problem in most of RS dataset). When the rating data is small, the system might not have enough data to make the accurate prediction. Moreover, the efficiency of CF-based is depended on the number of users in the systems. If there are few users, the system might not be able to find the users who are actually similar to an active user—for making the effective prediction.

2.2.3 Hybrid Recommendation

As described that both CBF and CF have their own advantages and disadvantages, to make the better recommendation and get rid of the drawbacks, most of the system usually combine them together. This type of method is called Hybrid Recommendation [6].

2.2 Model-based Recommender Systems

The collaborative filtering approaches can also be categorized by the way the rating data are used for the prediction, as memory-based and model-based collaborative filtering. The memory-based approaches make a prediction using the rating data that is stored in the memory directly. In contrast, the model-based approaches used the rating data to create the predictive model, and use them for the prediction without storing the raw data in the memory.

In this section, the two of well-known model-based methods: the Matrix Factorization [15] and the Flexible Mixture Mode [23] are presented.

2.2.1 Matrix Factorization

The Matrix Factorization (MF) [15] is one of the latent factor models; it exploits the item characteristics and the user preferences to predict the rating. The item characteristics and user preferences are the values that each item and each user are uniquely possessed, respectively. For example, the movie Titanic might have the characteristic of drama more than action or the user Helen might have more preference on drama movies than action movies. The idea of MF is that, if the item characteristics of an item are corresponding to the user preference of a user, then that item should be recommended to the user. For example, since Titanic possessed the characteristic of drama more than action, it is relevant to the preference of Helen who likes drama more than action movies; therefore it should be recommended to Helen.

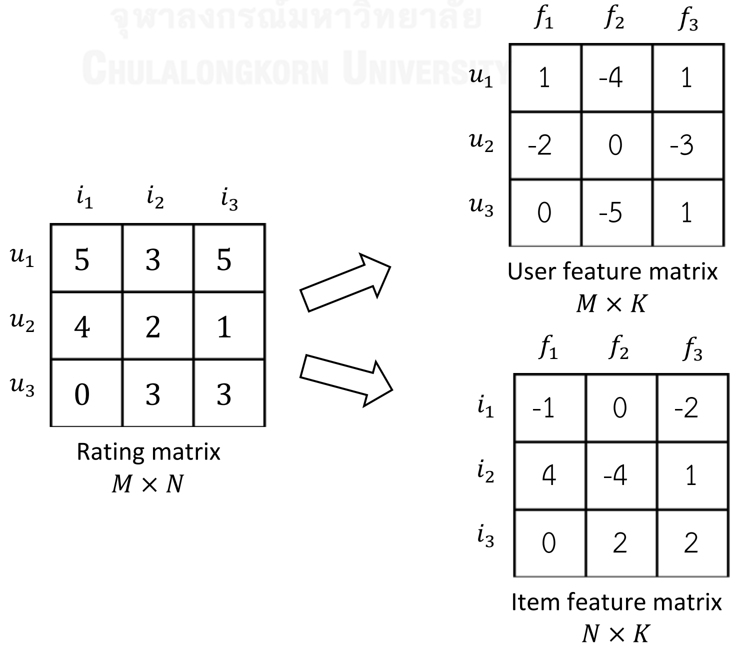


Figure 3. The Mechanism of Matrix Factorization

Figure 3 illustrates the mechanism of MF. The user-item rating matrix is factorized into the user feature matrix and item feature matrix. MF represents the item characteristics and the user preferences by the latent factor (or feature) vectors. Let p_u denotes the latent factor vector of user u, and q_i denotes the latent factor vector of item i. Then, the rating that user u will give to item i can be calculated by the dot product p_u and q_i of as in Equation (1).

$$\hat{r}_{ui} = {q_i}^T \cdot p_u \tag{1}$$

The value of the predicted rating will be high If each element of p_u and q_i is corresponding to each other. The system learns p_u and q_i by trying to minimize the regularized squared error between the actual and predicted ratings, as shown by Equation (2).

$$\min_{q^*,p^*} \sum_{(u,i)\in RT} \left(r_{u,i} - \hat{r}_{ui}\right)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \tag{2}$$

where

- $r_{u,i}$ is the actual rating,
- RT is the set of ratings in training data, and
- λ is the constant for controlling the extend of the regularization

The goal is to minimizing the square error between the actual and the predicted ratings, while avoids overfitting by considering the regularization. This minimizing can be done by using two kinds of optimization methods: the Stochastic Gradient Descent (SGD) and Alternating Least Square (ALS).

In order to improve the predictive performance, MF has been modified in various ways. For example, [15] modified MF by considering the additional biases and the temporal dynamics, as shown by Equation (3).

$$\hat{r}_{ui}(t) = \mu + b_i(t) + b_u(t) + {q_i}^T \cdot p_u(t) \tag{3}$$

where

- $\hat{r}_{ui}(t)$ is the predicted rating for user u on item i at time t

- μ is the overall average rating
- $b_u(t)$, $b_i(t)$ are the deviations from the average (biases) of user u and item i, respectively, at time t
- $p_u(t)$ is user u preference at time t

The idea behind this is that, the users' preferences on items can be changed when the time passes. The parameter $b_i(t)$ indicates the fact that the item popularity can be changed over time. For example, the popularity of the movies might depend on some events, or the time it is shown. Also, the parameter $p_u(t)$ indicates that the user can change his baseline rating over time. For example, a user might usually rate '3' on the past, but now he might like to rate '4'. Finally, the parameter $p_u(t)$ indicates that the user preference can also be changed over time. For example, he might prefer the drama movies in the past, but now he may prefer the action movies. On the other hand, the parameter q_i is not depended on time since the item characteristic is not likely changed by the time it is consumed. For example, the movie Titanic cannot change it genre from drama to horror when the time passes.

2.2.2 Flexible Mixture Model

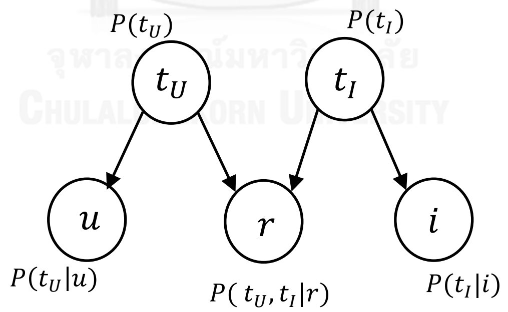


Figure 4. The Flexible Mixture Model

The Flexible Mixture Model (FMM) [25] calculates the probability that user u will give rating r to item i based on latent classes that show the characteristics of u and i. Let T_U and T_I be the sets of latent classes for users and items, respectively. The graphical model that explains a single rating prediction using the FMM is shown in Figure 4. The FMM defines the joint generation probability $P(u,i,r)$ for user u, item i, and rating r as:

$$P(u,i,r) = \sum\nolimits_{T_U,T_I} P(t_U)\,P(t_I)P(t_U|u)P(t_I|i)P(t_U,t_I|r) \quad (4)$$

where $t_U \in T_U$ and $t_I \in T_I$ and

- $P(t_U)$ and $P(t_I)$ are multinomial distributions on the user and item classes, respectively,
- $P(t_U|u)$ is a multinomial distribution on u given the specific class t_U,
- $P(t_I|i)$is a multinomial distribution on item i given the specific class t_I, and
- $P(t_U,t_I|r)$ is a multinomial distribution on rating r given the specific classes t_U and t_I.

The rating is calculated by using the sum of ratings r weighted by probability $P(u,i,r)$ as shown by Equation (5).

$$\hat{r}_{u,i} = \sum_{R} r \frac{P(u,i,r)}{\sum_R P(u,i,r)} \quad (5)$$

The model parameters of FMM are learned by using the Expectation and Maximization (EM) algorithm [8], which is one of the maximum likelihood estimation techniques. One drawback of the maximum likelihood estimation is that, the parameters are estimated using only the observed data, which may cause the overfitting problem. The Bayesian Flexible Mixture Model (BFMM) [23] was proposed to deal with this by applying the Bayesian estimation to estimate the parameters of FMM, instead of EM. The estimation on the BFMM is made by considering on both observed and unobserved (the prior parameters) data, which help avoiding the overfitting.

2.3 Context-Aware Recommender Systems

2.3.1 Definition of Context

Most of the traditional RS techniques rely only on two types of data for making the recommendations: user and item. However, recent researches on Context-Aware Recommender Systems (CARS) found that the quality of recommendation can be improved by incorporating the contextual information into the model. The contextual factors such as location, companion, mood or weather might also have a strong effect on the users' decision to decide which kind of products they prefer. For example, a user might choose to watch an action movie when he watch with friends at the cinema, while the same user might choose to watch a comedy movie when he watch with family at home. By considering the contextual data in the recommendation process, the prediction results have shown to be more accurate than the standard RS methods that ignored context [2], [26].

Anind K. and Gregory D. [9] defined the context as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." In recommender systems, context should be any situation variables that have the influences on the users' decision on choosing the items. For example, in movie recommendation, the contextual variables like location, companion, time or mood should be the ones that affect the users' preferences on movies. In this work, the term "contextual factors" is referred to the types of contextual variables, for example, location, season or weather. Also, the term "contextual conditions" is referred to the possible values of each contextual factor. For instance, the contextual conditions of the contextual factor "weather" are "sunny", "cloudy", "rainy", "snowy", "foggy" and "windy".

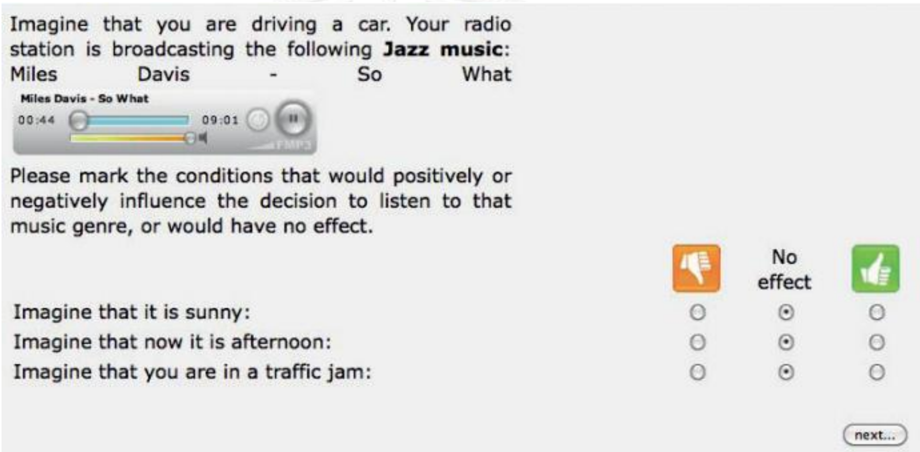
2.3.2 Acquisition of Contextual Data

There are three possible ways to obtain the contextual information: Explicitly, Implicitly and Inferring.

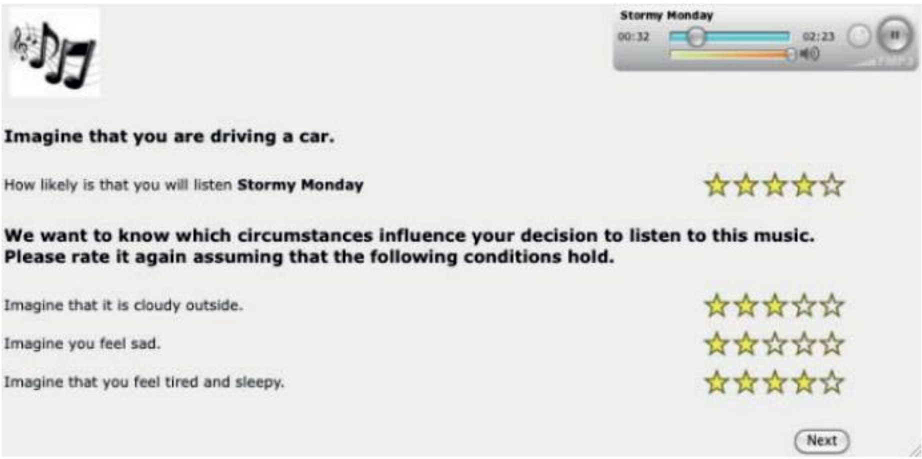
- **Explicitly**: The easiest way to obtain the contextual information is by asking the users directly. The system may request the users to fill some forms or questionnaires at the time they rate the items. Figure 5 shows the example of way to collect the contextual information explicitly by the application called InCarMusic [3]. In Figure 5(a), the users are requested to choose whether each context has effect on their choices of music (Positive Effect/ No Effect/

Negative Effect). On the other hand, the users are asked to rate the music under the different contextual situations as shown Figure 5(b).

- **Implicitly**: The other way to collect the contextual information is by indirectly tracking the user behavior and their surrounding environments. For example, the location detected by the mobile devices or the timestamps when the transactions are made. This way, the contextual information is extracting from the data itself; no forms or questionnaires are required.
- **Inferring**: The contextual information can also be obtained by inferring from the data, using the data mining or statistical methods. For example, the system can build the classifiers to decide whether the transactions are done by whom or in which situations.



(a) Effects of context



(b) User's rating under different contextual situations

Figure 5. Obtaining the Contextual Information Using InCarMusic Application

Besides the three ways to obtain the contextual information mentioned above, the contextual information could be extracted and used as latent variables; without knowing explicitly the values of each context. For example: [10] creates the latent topics by applying the Latent Dirichlet Allocaition (LDA) technique on the item features. These latent topics are then used as contextual features that capture the relationship among the item features.

2.3.3 Dealing with Contextual Information

Most of the traditional RS techniques were designed to make the recommendation considering only the user and item information. In contrast, CARS must deal with additional contextual information as well as the typical item rating information from users. Therefore, multi-dimensional (MD) ratings are required instead of the ordinary 2D ones. For example, if the task is to predict users' ratings of items at given times, the estimation function is represented as:

$$F_{User\times Item\times Time}: U \times I \times T \rightarrow R$$

where

- T is information about time acquired in the input process, and
- $F_{User\times Item\times Time}$ is a 3D estimation function that exploits information about users, items, and time to estimate the preference ratings.

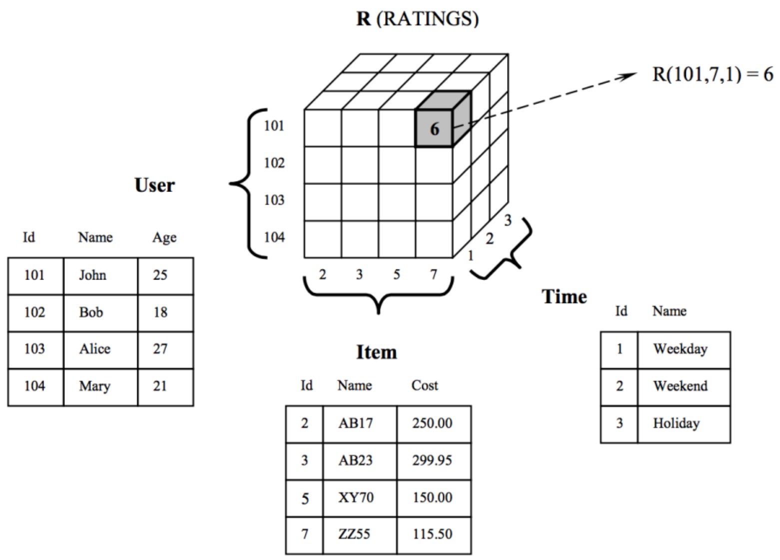


Figure 6. Multidimensional Representation of the Contextual Rating Data

As mentioned that when considering only user and item information, the rating data can be stored the 2D rating matrix (as shown in Figure 2). However, by taking the contextual information into consideration, the MD representation is required. Figure 6 shows the 3D representation of rating data; which consists of user, item, and time information in each dimension. Like the user-item rating matrix, each element of this data cube stored the rating that each user rated each item under a specific time. For example, John rated the item AB17 on weekday with the rating of 6.

With more dimensions, the classical 2D rating estimation techniques are no longer directly applicable for context-aware rating data. However, there are still the ways to apply those 2D techniques in CARS: if the contextual information is incorporated into the proper stages in the recommendation process. Figure 7 shows three possible stages of the recommendation process where the contextual information can be incorporated.

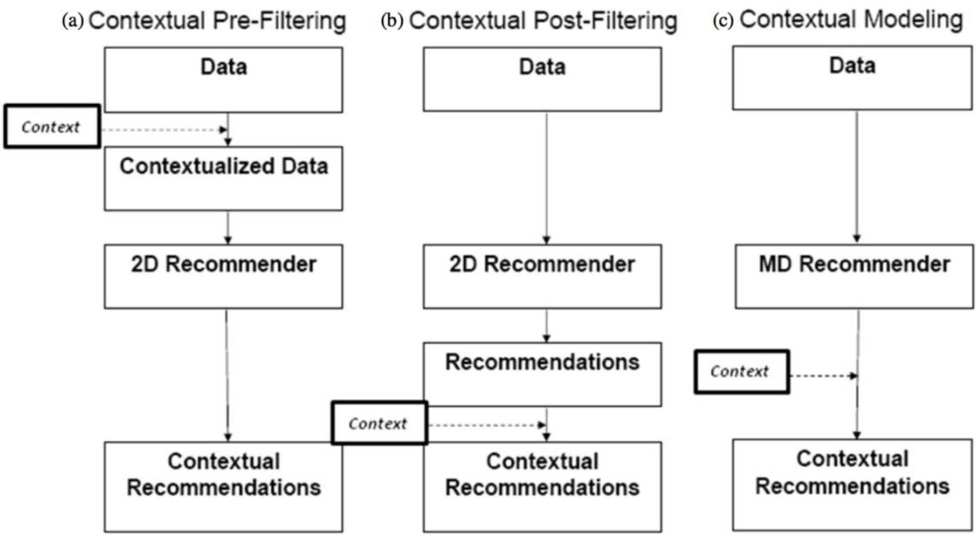


Figure 7. Three Approaches to Incorporate Context

From Figure 7, the CARS techniques based can be categorized by the stages of the recommendation process the contextual information is applied as: Contextual Pre-filtering, Contextual Post-filtering, and Contextual Modeling [2].

- **Contextual pre-filtering**: The contextual information is used as a filter for the data, i.e. only the rating data collected in the target context is incorporated into the recommendation process. For example, if an active user wants to watch a movie on weekend, only the rating data collected on weekend will be used as an input for the standard 2D recommendation techniques. The example of CARS techniques that are based on contextual pre-filtering approach are [2], [5].
- **Contextual post-filtering**: The contextual information is used to modify the recommendation results from standard 2D recommendation techniques. First, this approach extracts the item usage patterns from each user under a specific contextual situation. Then, the items that are irrelevant to an active user's contextual situation are re-ranked or filtered out from the recommendation list. For example, if the system detected that an active user only watch action or horror movie on weekend, all of the non-action or non-horror movies will be filtered out from the recommendation list of this user. The example of CARS techniques that are based on post-filtering approach are [20].
- **Contextual modeling**: The contextual information is applied in the recommendation function directly. Therefore, this approach requires the multidimensional recommendation techniques to make the rating estimation. The example of the CARS techniques that are based on contextual modeling approach are [4], [10], [11], and [26].

Although some works [7] have conducted the experiments to compare the predictive performances of these three approaches, there is still no clear conclusion which one dominates the others. Apparently, the strength of the pre-filtering and post-filtering approaches is that any existing 2D recommendation techniques can be applied to make the context-aware recommendation results. In contrast, the contextual modeling approach requires the truly MD estimation techniques. However, the similarity among these three approaches is that their performances are depended mainly on the way the relevant contextual factors are extracted from the data.

In the next section, some of the context-aware recommendation techniques that are related to this work are presented.

2.3.4 Context-Aware Recommendation Techniques

Until now, several methods in context-aware recommendation were proposed. In this section, only the methods which are related to this work is presented. These methods are the Context-Aware Collaborative Filtering [2], the Context-Aware Matrix Factorization [4], and the Differential Context Relaxation [28], [29].

2.3.4.1 Context-Aware Collaborative Filtering

One of the earliest techniques in CARS is the Context-Aware Collaborative Filtering (CACF) [2], which is based on the CF-based approach. This method extracts only the rating data that match with an active user's contextual situation, and then exploits the 2D estimation techniques like the CF-based to make the prediction.

Let $R^{D}_{User\times Item}: U \times I \rightarrow Rating$ denotes the 2D estimation function where D is the dataset containing the rating records in the format $< \mathbf{user, item, rating} >$. Also, let $R^{D}_{User\times Item\times time}: U \times I \times T \rightarrow Rating$ denotes the 3D estimation function where T is the time dimension. The rating estimation for the user u on item i with time t using 2D estimation function can be done by Equation (6).

$$R^{D}_{User\times Item\times Time}(u,i,t) = R^{D[Time=t|User,Item,Rating]}_{User\times Item}(u,i) \tag{6}$$

Where $D[Time = t|User, Item, Rating]$ means selecting only the rating records in D that are given in time t. From this, the rating data will contain only the user and item dimension, and can be used as an input for the CF-based approach.

The most important thing for the CACF approach is the selection of the contextual factors that will be used to filter the rating data. If the selected contextual factors contain too specific contextual conditions, there might not be the significant different between the ratings contained in those conditions.

For example, the ratings from the users who watched the movies on Sunday might be the same as the users who watched on Saturday. Moreover, if the selected contextual factors contain too many conditions, the sparsity problem might occur. For example, if the contextual factor "day of week" is selected, it is possible that there might be no rating given on Monday. When an active user want to watch movie on Monday, then the prediction cannot be done.

These problems can be solve by do not use too specific or use more general contextual conditions. For example, the "day of week" context; which contains 7 conditions (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday), can be generalized into 2 conditions: weekday (Monday-Friday) and weekend (Saturday and Sunday). Using these generalized conditions can provide the system with more ratings to make the prediction. For example, all the ratings given on weekday can be used to predict the rating for an active user who wants to watch a movie on Monday.

Adomavicius et al. [2] proposed a reduction-based approach with a technique called "contextual generalization", which can be used to generalized the overly-specific contextual conditions to become more general. Instead of using all the ratings in $[Time = t]$, this method selects the ratings from $[Time \in S_t]$ to make the recommendation. S_t is a superset of t called contextual segment. For example, if the task is to predict the rating that John will give to Titanic on Monday, instead of selecting the ratings given in $[Time = 'Monday']$, the system will select the ratings given in $[Time \in S_{Monday} = 'Weekday']$ to make the prediction. Therefore, the estimation function using S_t can be re-written as Equation (7).

$$R^{D}_{User\times Item\times Time}(u,i,t) = R^{D[Time\in S_t|User,Item,AGGR(Rating)]}_{User\times Item}(u,i) \qquad (7)$$

The reason of using $AGGR(Rating)$ instead of $Rating$ is that some users may rate the same item more than one time in different contextual conditions. For example, John may watch Titanic in Monday and Tuesday, and gave the ratings to this movie twice. When Monday and Tuesday are merged into weekday by the generalization, some aggregation functions (e.g. average) are needed to be applied to these two ratings.

The performance of the reduction-based approach is depended on the way the contextual segments are selected. If the contextual segments contain enough ratings for making an effective prediction, this method might outperforms the 2D recommendation techniques. However, if there are small numbers of ratings in those contextual segments, the 2D techniques may perform better. Although the 2D recommendation techniques have more rating data for making the prediction, some of the ratings may not be related to an active user's contextual situation, and act as the noises that degrade the prediction accuracy. On the other hand, the reduction-based approach can use more related rating data from the contextual segment to make prediction, but also has less data to learn. Therefore, the main challenge of this approach is to determine the appropriate level of the generalization of contextual information to use as the data filtering. This generalization is usually done by manual, using the opinions of domain experts; or by automated, trying every generalization and choosing the one with best performance.

2.3.4.2 Context Aware Matrix Factorization

The Context-Aware Matrix Factorization technique (CAMF) [4] extended the idea of MF with temporal dynamics, as proposed in [14], [15]. This method predicts the rating that user u will give to item i under the contextual conditions $c_1, c_2, \dots, c_k$ of k contextual factors as follows.

$$\hat{r}_{uic_1 \dots c_k} = \vec{v}_u \cdot \vec{q}_i + a_i + b_u + \sum_{j=1}^{k} B_{ijc_j} \tag{8}$$

The first term comes from standard MF, where $\bar{v}_u$ and $\bar{q}_i$ are latent-factor vectors of u and i, respectively. The average rating a_i of item i and a baseline parameter b_u for user u are added to improve the accuracy. Finally, the additional parameters $\sum_{j=1}^{k} B_{ijc_j}$ are included to model the relationship between the item i and the contextual conditions $c_1, \dots, c_k$.

Although CAMF seems to be an acceptable technique that yields good accuracy for less cost, it may not be suitable for real-world situations. This technique considers the relations between different contextual factors and items but ignores the relations between contextual factors and users. In the real world, context might also affect user preference patterns, and this could also influence the accuracy of ratings.

2.3.4.3 Differential Context Relaxation

Another interesting work on CARS is the method called Differential Context Relaxation (DCR) [29]. Usually, most of the CARS techniques try to identify only one set of the relevant contextual factors which has an effect on the ratings, and apply it to the whole recommendation algorithm. The DCR, in contrast, introduces the idea of identifying and applying the suitable set of relevant contextual factors to each component of the algorithm differently. The author modified the Resnick's prediction algorithm (Equation (9)) to incorporate contextual features into consideration as shown by Equation (10).

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in N}(r_{u,i} - \bar{r}_u) \times sim(a,u)}{\sum_{u \in N} sim(a,u)} \tag{9}$$

$$P_{a,i,C} = \bar{r}_{a,C_3} + \frac{\sum_{u \in N_{C_1}}(r_{u,i,C_2} - \bar{r}_{u,C_2}) \times sim(a,u)}{\sum_{u \in N_{C_1}} sim(a,u)} \tag{10}$$

where

- a is a user, i is an item, C is a given contextual situation,
- $\bar{r}_a$ is the average ratings from user a,
- $sim(a,u)$ is the similarity between user a and his neighbor, user u,
- N is the neighbors of user a, and
- $C_1, C_2, C_3 \subseteq C$ are the set of contextual factors

The set of contextual factors C_1, C_2, C_3 are worked as the constraints for filtering the rating data available in each component of the algorithm. From the Equation (10), $u \in N_{C_1}$ means that only the neighbor users who provided the ratings under the context C_1 are considered. Similarly, $(r_{u,i,C_2} - \bar{r}_{u,C_2})$ means to consider only the ratings from the neighbors which are given under the context C_2. Finally,

$\bar{r}_{a,C_3}$ is the average of the ratings given under the context C_3 by the user a.

The main goal of the DCR is to identify the optimal set of C_1, C_2, C_3 for each component of the algorithm. This was done by applying the exhaustive search and evaluating the predictive performance on each of their combination to retrieve the best one. The authors also extended their work by applying the Binary Particle Swarm Optimization (BPSO) [28] as the feature selection technique on the larger set of contextual factors. The experiment shows that applying the suitable relevant contextual factors to each component of the algorithm resulting on more accurate prediction and providing better coverage than the methods which exploit the same set of relevant contextual factors to the whole algorithm.

2.3.5 Challenge in Context-Aware Recommendation

One of the common problems in RS is the data sparsity: most users only give a few ratings compared to all possible ratings. For example, in the system containing 50 users and 100 items, there will be a total of 5000 possible ratings if each user rated every single item. However, in the real world, users only rated a few available items, and left most of them remain unrated. For instance, if each of these users only gave the average of 5 ratings, there will be a total of 4750 empty ratings (95% sparse). Because of this, the system may not have enough data to make an effective recommendation. When it comes to CARS, which considers contextual dimensions of data besides the original two dimensions user and item, this problem becomes even worse. Consider the previous example which taking into account only the user and item dimensions, there are total of 5000 possible ratings. Suppose the contextual factors: Daytype with 3 conditions, Location with 3 conditions, and Emotion with 7 conditions are incorporated, the total of possible ratings will be increased to $5000 \times 3 \times 3 \times 7 = 31500$ ratings. If each user gave the average of 5 ratings, there will be a total of 31250 empty ratings (99% sparse). Although it seems that the prediction accuracy can be improved by adding more contextual factors, it is important to consider the tradeoff with the sparsity of data.

Therefore, one of the crucial parts of CARS is to identify which of contextual factors should be incorporated into the model. To make an effective recommendation, only the relevant contextual factors to the recommendation should be extracted from the dataset. Using all of the contextual factors not only invoke the sparsity of the data, but also degrade the recommendation quality. This is because using the irrelevant contextual factors that are not related to the objective of the model can significantly affect the prediction accuracy [2], [18].

In the next section, some of the useful feature selection techniques for context-aware recommendation that are related to this work are presented.

2.4 Feature Selection Techniques for Context-Aware Recommendation

2.4.1 Exhaustive Search technique

The most simple and basic way for finding the relevant contextual factors is by applying the exhaustive search on the set of contextual factors. This method will evaluate the predictive performance of all possible combinations of the contextual factors, and select the combination that provides the best performance. For example, suppose there are 3 contextual factors: Time, Weather, and Location; all combinations and their accuracy results are shown in Table 1. Since there are 3 contextual factors, all possible combinations are 2^3 = 8 combinations. Among these 8 combinations, the combination <1,0,1> provided the most accurate result/ Therefore the context Time and Location are selected as the relevant contextual factors.

Table 1. Example of Exhaustive Search Feature Selection where '1' indicates relevant, '0' otherwise

Contextual Factors			RMSE
Time	Weather	Location	
0	0	0	1.65
1	0	0	1.45
0	1	0	1.47
0	0	1	1.54
1	1	0	1.37
1	0	1	1.24
0	1	1	1.44
1	1	1	1.29

The weak-point of the exhaustive search is that it is only applicable for the dataset with small number of contextual factors, and the methods that do not take a lot of time on making prediction. Usually, the context-aware datasets contain not only 3 or 4, but sometimes more than 10 contextual factors. For this case, the exhaustive search might not be practical or scalable.

Many methods were proposed to reduce the workload of the exhaustive search on finding the relevant contextual factors, including the Binary Particle Swarm Optimization technique.

2.4.2 Binary Particle Swarm Optimization

The Binary Particle Swarm Optimization (BPSO) [13] is the discrete binary version of the Particle Swarm Optimization [12] technique, which the values of the solutions are only {0, 1}. The idea is that the optimization is made by the number of particles that move through the search-space searching for the optimal solution. The movement of each particle is guided by its local best solution as well as the global best solution among all other particles. To determine the best solution, the fitness function for measuring the quality of the solutions is needed to be defined, which is depending on the objective of each specific model. For example, the Mean Absolute Error (MAE) or the Root Mean Square Error (RMSE) can be used as the fitness values, to measure the accuracy of the prediction of RS methods.

Given the following variables:

- S_d is the binary vector representing the solution of a problem, derived from particle d,
- x_{d_j} is the value of the j^{th} position of the binary vector S_d,
- v_{d_j} is the velocity of x_{d_j},
- $P_{lBest(d_j)}$ is the value of x_{d_j}, retrieved from the iteration which the particle d has received the best fitness value,
- $P_{gBest(j)}$ is the value of the j^{th} position of S from the particle that provide the best fitness value among all particles.

In each iteration of the optimization, each particle moves to the new position (the value to be optimized, i.e. the solution) in the search-space based on its velocity. This velocity is calculated based its velocity on the previous iteration, along with the local and global best position it has received so far.

The velocity of the j^{th} position of S_d at the iteration IT can be calculated by

$$v_{d_j}^{IT} = \omega_{IT} v_{d_j}^{IT-1} + \alpha_1 \varphi_1 \left(P_{lBest(d_j)} - x_{d_j}^{IT-1} \right) + \alpha_2 \varphi_2 \left(P_{gBest} - x_{d_j}^{IT-1} \right) \quad (11)$$

and the value of j^{th} position of S_d is updated by

$$x_{d_j}^{IT} \begin{cases} 1, \text{if } (rand_{IT} < S(v_{d_j}^{IT})) \\ 0, \text{otherwise} \end{cases} \quad (12)$$

where

ω_{IT} is inertia on iteration IT, define as $\omega_{IT} = \omega_{end} + \frac{(MAX\_IT - IT)}{MAX\_IT}(\omega_{start} - \omega_{end})$, where MAX_IT is maximum number of iterations,

φ_1, φ_2 is random values between [0, 1],

α_1, α_2 is learning factors that control the weight of local and global minimum respectively (set as 2.0 as suggested by [28]),

$rand_{IT}$ is random value between [0, 1] using uniform distribution,

$S(v_{d_j}^{IT})$ is sigmoidal function for mapping the value of $v_{d_j}^{IT}$ to the range [0, 1], defined as $S(v_{d_j}^{IT}) = \frac{1}{1+\exp(-v_{d_j}^{IT})}$.

The inertia ω_{IT} is used to manage the search of BPSO; the large inertia facilitates the global search while the small facilitates the local search. Following the study on [28], the value of ω starts with $\omega_{start} = 0.9$, then its value are linearly decreasing to $\omega_{end} = 0.4$.

The algorithm starts by initializing the value of each position of S_d (can be only 0 or 1), and the corresponding velocity of that position. The values of velocity is restricted within the range $[-v_{max}, v_{max}]$, where v_{max} is suggested to be the maximum value of each bit on S_d (i.e. 1). For each iteration, the value of velocity and position of each particle are updated using Equation (11), and Equation (12), respectively. After that, each particle evaluates its performance and updates the local best fitness value along with the corresponding solution it has received so far. When all particles are done with the evaluation, the global best fitness value and the corresponding solution are then identified.

The weak-point of the original BPSO is that, each position of the solution is depended only on the possibility of changing to 1, and ignores the possibility of changing to 0. This is the value of each position is determined by its velocity; if the velocity is high, then it has high possibility to change to 1. The improved version of BPSO by considering on both possibilities is proposed. The idea is that, if the bit of global best solution is 1, the velocity of change to 1 of that position should be increased, and the velocity of change to 0 should be decreased. Based on this idea, the velocity can be calculated using the following:

$$If\ \ P_{lBest(d_j)} = 1\ \ Then\ \ q^1_{d_j,1} = \alpha_1 r_1\ \ and\ \ q^0_{d_j,1} = -\alpha_1 r_1$$
$$If\ \ P_{lBest(d_j)} = 0\ \ Then\ \ q^0_{d_j,1} = \alpha_1 r_1\ \ and\ \ q^1_{d_j,1} = -\alpha_1 r_1$$
$$If\ \ P_{gBest(j)} = 1\ Then\ \ q^1_{d_j,2} = \alpha_2 r_2\ \ and\ \ q^0_{d_j,2} = -\alpha_2 r_2$$
$$If\ \ P_{gBest(j)} = 0\ Then\ \ q^0_{d_j,2} = \alpha_2 r_2\ \ and\ \ q^1_{d_j,2} = -\alpha_2 r_2$$

where $q^1_{d_j}$ and $q^0_{d_j}$ are two temporary values, r_1 and r_2 are two random variables in range (0,1), α_1 and α_2 are the same learning factors as the original BPSO. With this, the velocity is separately calculated with two cases: the velocity of changing to 1 ($v^1_{d_j}$) and the velocity of changing to 0 ($v^0_{d_j}$).

$$v^1_{d_j} = \omega v^1_{d_j} + q^1_{d_j,1} + q^1_{d_j,2} \tag{13}$$

$$v^0_{d_j} = \omega v^0_{d_j} + q^0_{d_j,1} + q^0_{d_j,2} \tag{14}$$

The decision on which velocity will be used is depended on the current value of that j^{th} position of S_d, as Equation (15).

$$v^{IT}_{d_j} \begin{cases} v^1_{d_j}, \text{if } (x^{IT-1}_{d_j} = 0) \\ v^1_{d_j}, \text{if } (x^{IT-1}_{d_j} = 1) \end{cases} \tag{15}$$

And finally, the value of each position is updated by

$$x_{d_j}^{IT} \begin{cases} \overline{x}_{d_j}^{IT-1}, \text{if } (rand_{IT} < S(v_{d_j}^{IT})) \\ x_{d_j}^{IT-1}, \text{otherwise} \end{cases} \tag{16}$$

where $\overline{x}_{d_j}^{IT-1}$ is the 2's complement of $x_{d_j}^{IT-1}$.

In the next section, some evaluation metrics which are able to use for calculating the fitness value are introduced.

2.5 Evaluation Metrics

Several kinds of the evaluation metrics for measuring the performance of RS methods were proposed [24], for example: accuracy, coverage, novelty, serendipity, and diversity. However, this section will focus only on the accuracy and the diversity, which are most related to this work.

2.5.1 Accuracy Metrics

The accuracy metrics measure how close the predictions made by RS methods are, compared to the actual preferences from the users. Many accuracy metrics for RS are proposed, but the one that popularly use in CARS techniques is the Root Mean Square Error (RMSE). The value of RMSE is calculated by:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{N}(\hat{r}_t - r_t)^2}{N}} \tag{17}$$

where

- N is the number of rating records,
- r_t is the actual rating at record t, and
- $\hat{r}_t$ is the predicted rating at record t.

Most of the RS methods make the recommendations based only on the accuracy of the model. However, the recent research [17] found that considering only the prediction accuracy alone might not be enough to recommend the items that satisfy the users. For example, the accuracy-based model might recommend the user only the same kind of movies he watched before, which might not interested him. Not like the accuracy, some metrics can be used to measure how satisfy the recommendation list will be for the user, including the diversity.

2.5.2 Diversity Metrics

The diversity of the recommendation has proved to be useful to enhance the user satisfaction on the recommended items [22]. The idea of the diversity is that the recommendation list should not be only personalized to the users' taste, but also provide as diverse list of items as possible. Most users are more likely to have the diverse tastes on things; even they are in the same domain. For example, a user might like both action and fantasy movies, rather that action alone or fantasy alone. Therefore, considering the diversity in the recommendation rises the opportunity for the items to be recommended to the users.

The diversity metrics can be categorized into two main types: individual diversity and aggregate diversity [1]. The individual diversity is measured on how diverse the items in the recommendation list for each user. In contrast, the aggregate diversity is measured on how diverse the recommended items are, across all users.

Several diversity metrics were proposed in the literatures, however; the one that related to this work is the genre diversity [27].

Baltrunas et al. [27] proposed a novel diversity metric called genre diversity. Instead of measuring how diverse the items themselves are in the recommendation list, this metric measures the diversity of the genres of these items. The authors claimed that the genre information from movies or books can be used to measure and enhance the diversity of the recommendation. Moreover, they presented the three important properties that the genre diversity should accomplish: genre coverage, genre redundancy, and recommendation list size-awareness.

- Genre coverage: the recommendation list should contain as many genres as possible. Also, those genres should be the ones that are interested by an active user.
- Genre redundancy: the recommendation list should not contain many items with the same kind of genre.

- Recommendation list size-awareness: the recommendation list should be made by considering the size of recommendation. For example, if it is a short recommendation list, only the items with the relevant genre to the user should be included in the list.

Based on these three properties, the authors proposed the genre diversity metric and the item re-ranking algorithm to produce the recommendation lists that are satisfied the user interests.

CHAPTER III
PROPOSED METHOD

As mentioned that this work consists of two main parts. In the first part, a novel way to incorporate the contextual features into the probabilistic model (FMM) by modeling their effects to both user classes and item classes is proposed. The optimization is done by using EM algorithm on the semi-synthetic dataset where the contextual information is generated. The second part, in contrast, is the improved version of the first part on the various aspects. First, the optimization technique is changed from the EM algorithm, which is the maximum likelihood estimation, into the Bayesian estimation since it is more reliable. Second, the experiment is done using the real context-aware dataset rather than the synthetic one. Finally, only the relevant contextual factors to each user class and each item class are incorporated into the model by combining the BFMM with the BPSO technique.

This chapter is organized as follow. First, the mechanism for incorporating the contextual information into the probabilistic model is presented. Then, two main proposed probabilistic models for context-aware recommendations, corresponding to the two part of this work are presented. The first probabilistic model is the extended FMM by incorporating with the synthetic contextual information, using EM algorithm as the optimization technique. On the other hand, the second model is the BFMM incorporated with the actual relevant contextual factors, using Bayesian estimation as the optimization technique. Finally, the fitness function for optimizing the proposed model in term of prediction accuracy and diversity is defined; for learning the relevant contextual factors.

3.1 Incorporating the Contextual Information

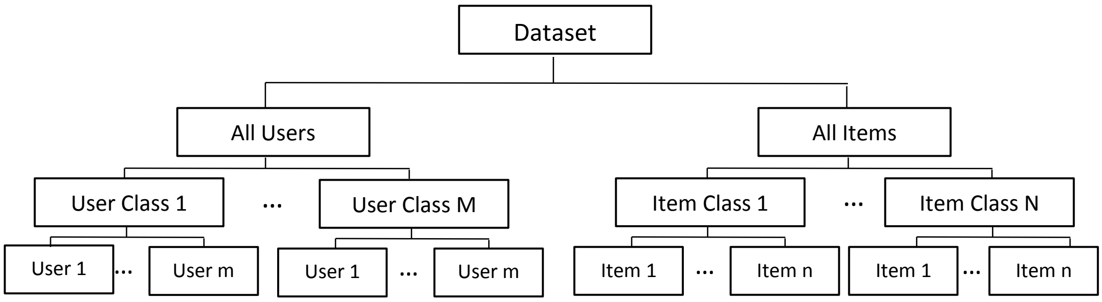


Figure 8. Hierarchical Structure of the Data

Because there are many possible ways to incorporate the contextual information, the most suitable way for the proposed model should be analyzed. Figure 8 presents the hierarchical representation of the data, showing the components of the system that can be affected by the contextual factors. The most general one (the top level) is the case that all data in the dataset have the same set of relevant contextual factors, which have been implemented in most of context-aware recommendation techniques [2], [5]. In contrast, the most specific way (the bottom level) is to find the set of relevant contextual factors to each individual user and/or each individual item, separately. The general one may be easier to implement and require less data to learn, but also less accurate. On the other hand, the specific one may provide more accurate prediction result since it considers more related data to each user and/or each item, but it might not be practical for most of the context-aware dataset, where the rating data is limited. For example [4] tried to capture the relation between each contextual feature and each item by extending the matrix factorization technique. In order to learn their relation effectively, it is required that an item must be rated by a significant amount of times under that contextual feature. This is almost impossible for most of the context-aware datasets, which contain limited amount of ratings. By considering the trade-off between these two cases, the experiment is made on the medium level, the case of finding the relevant contextual factors for each user class and each item class, which should be suitable for the latent probabilistic model. The process of learning the relevant contextual factors will be explained later on section 3.3.

3.2 Latent Probabilistic Model for Context-Aware Recomndation

Since the second probabilistic model (the BFMM incorporated with relevant contextual information) is the improved version of the first model (the extended FMM with contextual information), this section will focus mainly on its details, and provide a briefly explanation on the first one.

3.2.1 Flexible Mixture Model for Context-Aware Recommendation

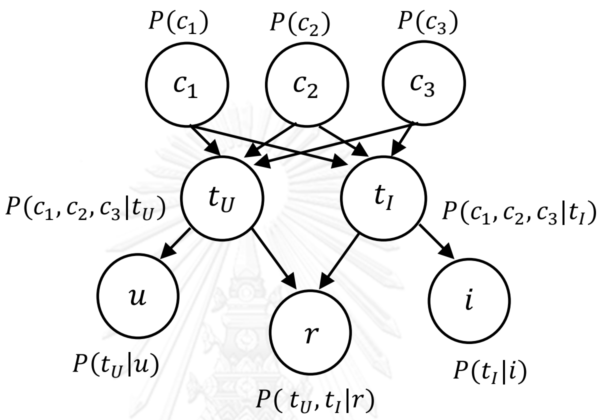


Figure 9. FMM incorporated with Context

Figure 9 shows the result of incorporating the contextual factors into the FMM by modeling their relationships to the user classes and item classes. The variables c_1, c_2, and c_3 represent the contextual conditions of the contextual factors incorporated into the model (in this case, there are 3 contextual factors). An arrow pointing from a contextual factor to the user classes or item classes means that there are relations between that contextual factor and those classes. For this model, the joint generation probability that u will give i a rating of r under contextual conditions c_1, c_2, and c_3 can be written as:

$$P(u, i, r, c_1, c_2, c_3) = \sum_{T_U, T_I} P(c_1)P(c_2)P(c_3)P(t_U|u)P(t_I|i) P(t_U, t_I|r)P(c_1, c_2, c_3|t_U)P(c_1, c_2, c_3|t_I) \tag{18}$$

where

- $P(c_1)$, $P(c_2)$, and $P(c_3)$ denote the probability that the context is c_1, c_2, and c_3, respectively,

- $P(c_1, c_2, c_3|t_U)$ denotes the probability that the user class is $t_U \in T_U$ under the context of c_1, c_2, and c_3, and
- $P(c_1, c_2, c_3|t_I)$ denotes the probability that the item class is $t_I \in T_I$ under the context of c_1, c_2, and c_3,

and the parameters $P(t_U|u), P(t_I|i)$ and $P(t_U, t_I|r)$ are the same with the ones defined in FMM as presented in Section 2.2.2.

Let X be the set of training data. Each element $x \in X$ is a vector $\{x.u, x.i, x.r, x.c_1, x.c_2, x.c_3\}$, meaning that user $x.u$ gives rating $x.r$ to item $x.i$ under the contextual conditions of $x.c_1$, $x.c_2$, and $x.c_3$. $P(c_1)$, $P(c_2)$, and $P(c_3)$ are estimated by counting how many times each condition appears in the training data. Other parameters are estimated using the EM algorithm as the following.

- **E-step**: In this step, the joint posterior probability of latent variables $\{T_U, T_i\}$ is calculated as follows.

$$P(t_U, t_I \mid x) = \frac{P(u, i, r, c_1, c_2, c_3, t_U, t_I)}{\sum_{T_U, T_I} P(u, i, r, c_1, c_2, c_3, t_U, t_I)} \tag{19}$$

- **M-step**: This step updates the model parameters by using the posterior probabilities obtained in the E-step as shown below:

$$P(c_1, c_2, c_3|t_U) \propto \sum_{x \in X} \sum_{t_I \epsilon T_I} P(t_U, t_I \mid x) \cdot \delta(x.c_1, c_1) \cdot \delta(x.c_2, c_2) \cdot \delta(x.c_3, c_3) \tag{20}$$

where $\delta(a, b)$ denotes the indicator function, equal to one if $a = b$ and zero otherwise. Similarly, the other model parameters are estimated by.

$$P(c_1, c_2, c_3|t_I) \propto \sum_{x \in X} \sum_{t_U \epsilon T_U} P(t_U, t_I \mid x) \cdot \delta(x.c_1, c_1) \cdot \delta(x.c_2, c_2) \cdot \delta(x.c_3, c_3) \tag{21}$$

$$P(t_U|u) \propto \sum_{x \in X} \sum_{t_I \epsilon T_I} P(t_U, t_I \mid x) \tag{22}$$

$$P(t_I|i) \propto \sum_{x \in X} \sum_{t_U \epsilon T_U} P(t_U, t_I \mid x) \cdot \delta(x.i, i) \tag{23}$$

$$P(t_U t_I|r) \propto \sum_{x \in X} \cdot \delta(x.r, r) \tag{24}$$

After the parameters are optimized, the predicted rating that user u will give to item i under contextual conditions c_1, c_2, and c_3 is calculated using the sum of rates weighted with the joint generation probabilities $P(u,i,r,c_1,c_2,c_3)$ as follows.

$$\hat{r}(u,i,c_1,c_2,c_3) = \sum\nolimits_R r \frac{P(u,i,r,c_1,c_2,c_3)}{\sum_R P(u,i,r,c_1,c_2,c_3)} \tag{25}$$

The strength of this model comes from its flexibility to define the relations of contextual variables to the user classes and item classes to the specific scenario. For example, some contextual factors like "Time" might have an effect on the user decision for selecting movies to watch, but not affect the movie characteristics. For this case, this model just simply ignore such relation by removing the arrow connecting between context Time and the item classes, and then make the parameter estimation only on the remaining relations. However, the relations among context factors, user classes, and item classes have to be defined manually, which is quite challenging. Moreover, the model probability for each contextual condition (e.g. $P(c_1)$), along with the model probability for each combination of {context, class} (e.g. $P(c_1,c_2,c_3|t_{U_1})$) are needed to be estimated, which further increase the complexity and cost of the model.

All of the limitations from the extended FMM (with context) are eliminated in its improved version: the BFMM incorporated with relevant context. First, the relations among contextual factors, user classes and item classes are modeled automatically by the identification of relevant contextual factors using BPSO technique. Also, the contextual factors are incorporated into the model by merging themselves with the user classes and item classes. This provides less complexity and consumes less computation cost than the extended FMM.

3.2.2 Bayesian Flexible Mixture Model for Context-Aware Recommendation

By incorporating the contextual information into BFMM, the proposed model, compared with the original BFMM, is shown by Figure 10 (a) and (b).

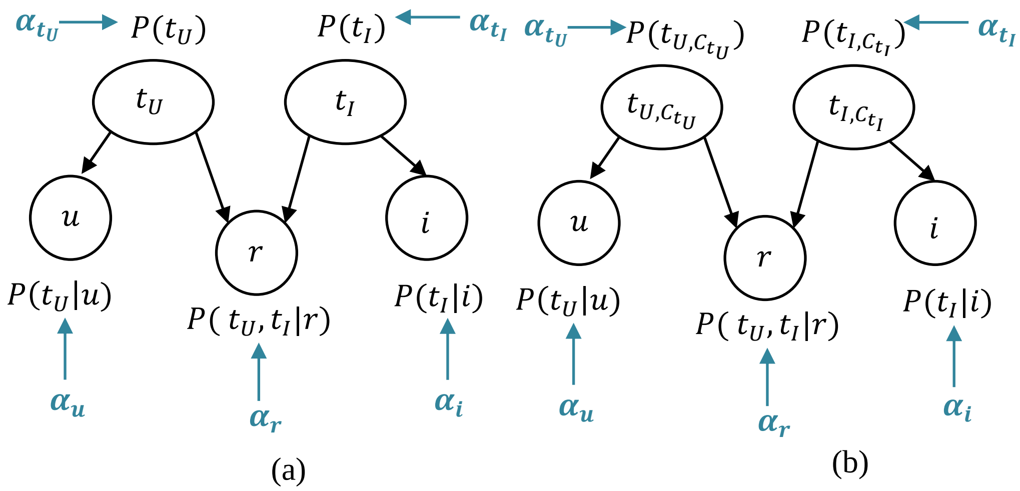


Figure 10. (a) BFMM (b) BFMM Incorporated with Context

Although it may look similar with the original BFMM (without context), the way its parameters are estimated is different. Table 2 shows the comparison on the model parameters between original BFMM and the proposed model. The difference is the way the parameters $P(t_U)$ and $P(t_I)$ are estimated. In original BFMM, $P(t_U)$ is calculated based on the constraint that all of the data used in its estimation must be given in the same user class t_U. Similarly, to estimate (t_I) , all of the data must be given in the same item class t_I. However, when the contextual information when the users consumed the items is known, these constraints can be modified further to make a better recommendation. As mentioned in the previous section, by incorporating the contextual information, the contextual factors which are relevant to each user class, and which are relevant to each item class can be identified. These relevant contextual factors can be used to categorize the original constraints to incorporate the contextual data. That is, the data used to calculate $P(t_U)$ and $P(t_I)$ must not only be assigned to the same classes, but also the same relevant contextual conditions as the relevant contextual factors to t_U and t_I.

Table 2. A Comparison Between Model Parameters

Model Probability	Bayesian FMM		Bayesian FMM with Context	
	Dirichlet	Occurrence	Dirichlet	Occurrence
$P(t_U)$	α_{t_U}	N_{t_U}	α_{t_U}	$N_{t_U,C_{t_U}}$
$P(t_I)$	α_{t_I}	N_{t_I}	α_{t_I}	$N_{t_I,C_{t_I}}$
$P(t_U\|u)$	α_u	$N_{t_U,u}$	α_u	$N_{t_U,u}$
$P(t_I\|i)$	α_i	$N_{t_I,i}$	α_i	$N_{t_I,i}$
$P(t_U,t_I\|r)$	α_r	$N_{t_U,t_i,r}$	α_r	$N_{t_U,t_i,r}$

where

- $\alpha_{t_U}, \alpha_{t_I}, \alpha_u, \alpha_i,$ and α_r are the Dirichlet parameters for the user class t_U, item class t_I, user u, item i, and rating r respectively,
- $N_{t_U}, N_{t_U,u}$ are the number of rating records which are given under the constraint t_U, and t_U with u respectively,
- $N_{t_I}, N_{t_I,i}$ are the number of rating records which are given under the constraint t_I, and t_I with i respectively,
- $N_{t_U,t_i,r}$ is the number of rating records which is given under the constraint t_U, t_I with r, and
- $N_{t_U,C_{t_U}}$, $N_{t_I,C_{t_I}}$ are the number of rating records which are given under the constraint t_U with C_{t_U}, and t_I with C_{t_I} respectively.

More technically, in the original BFMM the values of $P(t_U)$ and $P(t_I)$ are depended on the total number of rating records that have been assigned to class t_U (N_{t_U}) and class t_I (N_{t_I}), respectively. On the other hand, in order to calculate $P(t_U)$, the proposed model does not consider all of the rating records constrained by class t_U; instead, only those records given in the relevant contextual factors to the class t_U (denote by C_{t_U}) are taken into account. It can be said that in the proposed model, the class t_U is categorized into subclasses by its relevant context C_{t_U}, which are denoted by $t_{U,C_{t_U}}$. Therefore, the value of $P(t_{U,C_{t_U}})$ is depended on the number of rating records labeled by t_U and given in C_{t_U} (denoted by $N_{t_U,C_{t_U}}$). Similarly, $P(t_{I,C_{t_I}})$ is also estimated by the number of rating records constrained by t_I and given in C_{t_I} (denoted by $N_{t_I,C_{t_I}}$).

Figure 11, for example; shows how the user class t_{U1} is categorized by its relevant contextual factors. Suppose the relevant contextual factors to the class t_{U1} is $C_{t_{U1}}$= <Location, Daytype>, where Location = {home, theater} and Daytype = {weekday, weekend}. The class t_{U1} is then categorized into subclasses: $t_{U1,\text{<home,weekday>}}$, $t_{U1,\text{<home,weekend>}}$, $t_{U1,\text{<theater,weekday>}}$,and $t_{U1,\text{<theater,weekend>}}$. When an active user want to watch a movie at home on weekend, the parameter $P(t_{U1,\text{<home,weekend>}})$ is then calculated using $N_{t_{U1,\text{<home,weekend>}}} = 55$ instead of $N_{t_{U1}} = 200$. This can improves the optimization procedure to be more specific and personalized to an active user's contextual situation rather than using the original $P(t_{U1})$.

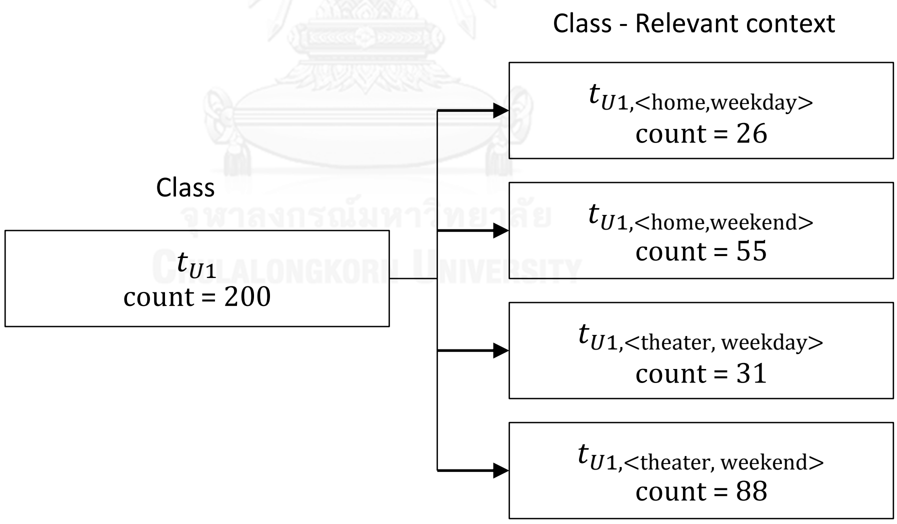


Figure 11. Categorizing a Class by Context

In order to learn the relevant contextual factors to each user class and each item class, the BFMM is combined with the BPSO technique. First, the model parameters are optimized using Gibbs sampling and Minka's fixed-point iteration in BFMM. The quality of the recommendation is then measured by the proposed fitness function as the combination of prediction accuracy and diversity. Finally, the relevant contextual factors to each user class and each item class are optimized based on the fitness values. Therefore, these relevant contextual factors are the ones that satisfying the objective of the proposed model: providing the diverse recommendation list, while maintaining the acceptable level of prediction accuracy.

The algorithm of the proposed method is presented in Figure 12. The details of the optimization will be shown next to this figure, and how the fitness function is defined for learning the relevant contextual factors will be discussed in Section 3.3.

Input:

- RT — the context-aware rating data, each rating record x_p is in the format x_p = <user, item, context, rating>
- max_it — the maximum iteration for optimizing the model parameters by BFMM
- MAX_IT — the maximum iteration for learning the relevant contextual factors by BPSO
- M — the number of particles used in BPSO
- $\sigma_{fitness}$ — the acceptable fitness value
- m, n — the number of user classes and item classes, respectively

Output:

- $C_{t_U}^{gBest}, C_{t_I}^{gBest}$ — the set of global best relevant contextual factors to each user class and each item class among M particles, respectively
- $fitness_{gBest}$ — the global best fitness value among M particles

Initialize:

- T_U, T_I — the set of m user classes and n item classes, respectively
- $\alpha_{t_U}, \alpha_{t_I}, \alpha_u, \alpha_i, \alpha_r$ — the Dirichlet parameters for each of user class, item class, user, item, and rating, respectively
- $x_p(t_u), x_p(t_i)$ — the user class and item class for each rating record $x_p \in RT$
- D — the set of M particles $\{d_1, \dots, d_M\}$
- $C_{t_U}^d, C_{t_I}^d$ — the set of binary vectors representing the relevant contextual factors to each user class and each item class of each particle $d \in D$

Optimization:

$\underline{for}$ each iteration IT until MAX_IT $\underline{or}$ until $fitness_{gBest}^{IT} \geq \sigma_{fitness}$ $\underline{do}$

$\underline{for}$ each particle $d \in D$ $\underline{do}$

$\underline{for}$ each iteration it until max_it $\underline{do}$

- **Gibbs sampling:** reassign new $x_p(t_u)$ and $x_p(t_i)$ to each x_p
- **Minka's fixed-point iteration:** optimize $\alpha_{t_U}, \alpha_{t_I}, \alpha_u, \alpha_i, \alpha_r$
- **Estimation:** estimate the model parameters
- **Prediction:** estimate the rating of each x_p

$\underline{endfor}$

Evaluate and update the local best fitness value of particle d at iteration $IT - fitness_{lBest}^{IT}$.

$\underline{endfor}$

Identify and update the global best fitness value among M particles at iteration $IT - fitness_{gBest}^{IT}$.

Optimize $C_{t_U}^d, C_{t_I}^d$ using BPSO technique.

$\underline{endfor}$

Figure 12. The Algorithm of the Proposed Method

The detail of the optimization procedure is presented below.

- **Gibbs sampling**: For each rating record $x_p^{(it)}$ at iteration it, reassign the new user class t_U and new item class t_I based on $P(t_U|\, x_p^{(it)})$ and $P(t_I|\, x_p^{(it)})$, respectively.

$$P(t_U|x_p^{(it)}) = \left(\alpha_{t_U}^{(it)} + N_{t_U,C_{t_U}^d}^{(it)}\right)\frac{\left(\alpha_r^{(it)} + N_{t_U,r}^{(it)}\right)}{\sum_R\left(\alpha_r^{(it)} + N_{t_U,r}^{(it)}\right)}\frac{\left(\alpha_u^{(it)} + N_{t_U,u}^{(it)}\right)}{\sum_U\left(\alpha_u^{(it)} + N_{t_U,u}^{(it)}\right)} \quad (26)$$

$$P(t_I|x_p^{(it)}) = \left(\alpha_{t_I}^{(it)} + N_{t_I,C_{t_I}^d}^{(it)}\right)\frac{\left(\alpha_r^{(it)} + N_{t_I,r}^{(it)}\right)}{\sum_R\left(\alpha_r^{(it)} + N_{t_I,r}^{(it)}\right)}\frac{\left(\alpha_i^{(it)} + N_{t_I,i}^{(it)}\right)}{\sum_I\left(\alpha_i^{(it)} + N_{t_I,i}^{(it)}\right)} \quad (27)$$

where

$N_{t_U,r}$ and $N_{t_I,r}$ are the number of rating records which are given under the constraint t_U with r and t_I with r respectively.

- **Minka's fixed-point iteration**: Optimizing the Dirichlet parameters by the following.

$$\alpha_{t_U}^{(it+1)} = \alpha_{t_U}^{(it)}\frac{\psi\left(\alpha_{t_U}^{(it)} + N_{t_U}^{(it)}\right) - \psi(\alpha_{t_U}^{(it)})}{\psi\left(\sum_{T_U}\alpha_{t_U}^{(it)} + \sum_{T_U}N_{t_U}^{(it)}\right) - \psi\left(\sum_{T_U}\alpha_{t_U}^{(it)}\right)} \quad (28)$$

$$\alpha_{t_I}^{(it+1)} = \alpha_{t_I}^{(it)}\frac{\psi\left(\alpha_{t_I}^{(it)} + N_{t_I}^{(it)}\right) - \psi(\alpha_{t_I}^{(it)})}{\psi\left(\sum_{T_I}\alpha_{t_I}^{(it)} + \sum_{T_I}N_{t_I}^{(it)}\right) - \psi\left(\sum_{T_I}\alpha_{t_I}^{(it)}\right)} \quad (29)$$

$$\alpha_u^{(it+1)} = \alpha_u^{(it)} \frac{\sum_{T_U} \left(\psi\left(\alpha_u^{(it)} + N_{t_U,u}^{(it)}\right) - \psi(\alpha_u^{(it)})\right)}{\sum_{T_U} \left(\psi\left(\sum_U \alpha_u^{(it)} + \sum_U N_{t_U,u}^{(it)}\right) - \psi\left(\sum_U \alpha_u^{(it)}\right)\right)} \tag{30}$$

$$\alpha_i^{(it+1)} = \alpha_i^{(it)} \frac{\sum_{T_I} \left(\psi\left(\alpha_i^{(it)} + N_{t_I,i}^{(it)}\right) - \psi(\alpha_i^{(it)})\right)}{\sum_{T_I} \left(\psi\left(\sum_I \alpha_i^{(it)} + \sum_I N_{t_I,i}^{(it)}\right) - \psi\left(\sum_I \alpha_i^{(it)}\right)\right)} \tag{31}$$

$$\alpha_r^{(it+1)} = \alpha_r^{it} \frac{\sum_{T_U,T_I} \left(\psi\left(\alpha_r^{(it)} + N_{t_U,t_I,r}^{(it)}\right) - \psi(\alpha_r^{(it)})\right)}{\sum_{T_U,T_I} \left(\psi\left(\sum_R \alpha_r^{(it)} + \sum_R N_{t_U,t_I,r}^{(it)}\right) - \psi\left(\sum_R \alpha_r^{(it)}\right)\right)} \tag{32}$$

where

ψ is a digamma function, defined as $\psi = \frac{d}{dx} \ln \Gamma(\mathrm{x}) = \frac{\Gamma'(x)}{\Gamma(x)}$

N_{t_U} and N_{t_I} are the number of rating records which are given under the constraint t_U, and t_I respectively.

Note that the parameters α_{t_U} and α_{t_I} are not categorized by their relevant contextual factors to become $\alpha_{t_U,c_{t_U}}$ and $\alpha_{t_I,c_{t_I}}$. This is because categorizing those Dirichlet parameters not only increase the complexity of the model, but also not suitable for the small and limited dataset. Moreover, using the parameters $N_{t_U,c_{t_U}}$ and $N_{t_I,c_{t_I}}$ to calculate $P(t_{U,c_{t_U}})$ and $P(t_{I,c_{t_I}})$ is already provided the acceptable result. By considering this trade-off, the way the Dirichlet parameters are optimized is maintain from the original (non-context) BFMM.

- **Parameter estimation**: Calculating the model parameters using the parameters optimized in Gibbs sampling and Minka's fixed-point iteration steps by the following.

$$P(t_{U,c_{t_U}}) = \frac{\left(\alpha_{t_U} + N_{t_{U,c_{t_U}}}\right)}{\sum_{T_U}\left(\alpha_{t_U} + N_{t_{U,c_{t_U}}}\right)} \tag{33}$$

$$P(t_{I,c_{t_I}}) = \frac{(\alpha_{t_I} + N_{t_{I,c_{t_I}}})}{\sum_{T_I}(\alpha_{t_I} + N_{t_{I,c_{t_I}}})} \tag{34}$$

$$P(t_U \mid u) = \frac{(\alpha_u + N_{t_U,u})}{\sum_{T_U}(\alpha_u + N_{t,u})} \tag{35}$$

$$P(t_I \mid i) = \frac{(\alpha_i + N_{t_I,i})}{\sum_{T_I}(\alpha_i + N_{t_I,i})} \tag{36}$$

$$P(t_U, t_I \mid r) = \frac{(\alpha_r + N_{t_U,t_I,r})}{\sum_{T_U,T_I}(\alpha_r + N_{t_U,t_I,r})} \tag{37}$$

- **Rating prediction**: The rating that user u is likely to rate item i under the contextual features C can be predicted by:

  First, calculating the joint generation probability $P(u, i, C, r)$:

$$P(u, i, C, r) = \sum_{T_U, T_I} P(t_{U,C_{t_U}})\, P(t_{I,C_{t_I}}) P(t_U \mid u) P(t_I \mid i) P(t_U, t_I \mid r) \quad (38)$$

  Then, the rating is estimated by the sum of rates weighted with $P(u, i, C, r)$:

$$\hat{r}(u, i, C) = \frac{\sum_R r \times P(u, i, C, r)}{\sum_R P(u, i,, C, r)} \quad (39)$$

The predicted ratings will be used to calculate the fitness value, which will be presented in Section 3.3. After that, the set of relevant contextual factors to each user class and each item class, along with the corresponding fitness value will be used as the input for the BPSO; in order to identify the best relevant contextual factors.

3.3 Identify the Relevant Contextual Factors

Now, the method for identifying the relevant contextual factors to user classes (C_{t_U}) and item classes (C_{t_I}) is presented. Before explaining the procedure for learning those contextual factors, the meaning of "relevant" in the proposed model must be defined. The relevant contextual factors should be the factors that, when consider them in the recommendation process, provide the recommendation results which satisfy the users' preferences. That is, for each candidate of the relevant contextual factors, the fitness function to measure how well it can produce the recommendation results has to be specified.

Most of CARS methods [2], [4], [5], and [28] define the relevant contextual factors as the factors that affect the prediction accuracy of the recommendation. However, the recent research [17] found that only good prediction accuracy might not be able to satisfy the user's interest. This is because the high accurate model may recommend limited only the set of items that can be expected by the user. For example, in movie recommendation, the results of the accuracy based recommendation may consist of the movies with similar characteristics. These movies might have the same set of directors or the same set of genres as the movies the active user has watched before, which is already known by this user. Therefore, many researchers in RS tried to improve the recommendation by considering the other evaluation metrics rather than the accuracy alone; for example: novelty and diversity. In this work, the quality of the relevant contextual factors is measured by the fitness function considering in both accuracy and diversity as they are often studied together [19], [30]. For the accuracy metric, the Root Mean Square Error (RMSE) is used as it is popularly use in most of CARS techniques. For the diversity metric, the similar idea of genre diversity proposed in [27] is derived; to measure how diverse and relevant the genres in Top-N recommendation list are for each user. For an active user u, the items are ranked by their predicted ratings in descending order to get his Top-N recommended items: $RecItem_u$. Then, the genre diversity for user u is defined as:

$$GenreDiv_u = \sum_{g \in G(RecItem_u)} \sum_{t_U \epsilon T_U} \left(\frac{N_{t_U,g}}{\sum_{t_U \epsilon T_U} N_{t_U,g}} \right) \left(\frac{N_{t_U,u}}{\sum_{t_U \epsilon T_U} N_{t_U,u}} \right) \tag{40}$$

where

- $N_{t_U,g}$ is the number of times the genre g is assigned to class t_U,
- $N_{t_U,u}$ is the number of times the user u is assigned to class t_U,
- $G(RecItem_u)$ is the set of unique genres in user u's Top-N recommended items.

The idea behind this is that, in order to provide high genre diversity, the Top-N recommended items for user u should contain as many genres as possible. Also, these genres should be relevant to the user classes where the user u is assigned. The relevant of each genre to each user class is measured by the first term; while the last term measures the relevant of each user to each user class. These two terms are easily derived from the proposed model since they can be considered as two model probabilities $P(t_U|,g)$ and $P(t_U|,u)$ without the Dirichlet parameters.

After the genre diversity for all users are estimated, the average genre diversity is calculated by:

$$GenreDiv = \frac{\sum_{u \in U} GenreDiv_u}{|U|} \tag{41}$$

where U is the set of all users.

Finally the fitness function is computed by combining the accuracy metric with genre diversity metric as:

$$fitness^{IT} = \delta^{IT} \cdot GenreDiv^{IT} + (1 - \delta^{IT}) \cdot \frac{1}{RMSE^{IT}} \tag{42}$$

where δ_{IT} is the diversity weight at iteration IT, which is defined as

$$\delta_{IT} = \delta_{min} + (IT - 1)\left(\frac{\delta_{max} - \delta_{min}}{MAX\_IT}\right) \tag{43}$$

where

- δ_{max}, δ_{min} are the maximum and the minimum diversity weight respectively,
- IT, $MAX_IT$are the current and the maximum iteration of the optimization respectively.

The values of δ_{min} and δ_{max} are set to 0.2 and 0.3, respectively. The weight of the genre diversity will be increased from 0.2 to 0.3 as the number of iterations increases. The idea is that in the beginning of the optimization, the relevant contextual factors will be guided mainly by the accuracy. Near the end of the optimization, when the acceptable level of accuracy is achieved, the relevant contextual factors will be optimized to provide more diversity.

In order to learn the relevant contextual factors of the proposed model, this fitness function is applied in BPSO technique. By applying BPSO technique, each particle evaluates its performance using Equation (42). The relevant contextual factors for each particle are then updated based on the local and global best performances. The optimization is executed repeatedly until the acceptable fitness value is achieved or until reaching the defined maximum iteration.

CHAPTER IV
EXPERIMENTAL RESULTS

In this chapter, the evaluation results of the proposed model are presented by comparing with other popular methods. First, the details of the dataset used in this experiment presented. Then, the experimental results of the methods; in terms of prediction accuracy alone, and considered the trade-off between accuracy and diversity are displayed. Finally, the results of the relevant contextual factors for each context-aware model are shown.

4.1 Dataset

The LDOS-CoMoDa dataset [16] is used to perform the experiments. The website encourages the users to provide the ratings to the movies along with the contextual information. Since the user is required to provide the ratings and the contextual information immediately after they watched the movies, these ratings are, therefore; being as close as the users' real preference at the situations they watched the movies. This makes the dataset more reliable than the other context-aware datasets.

At the time the experiments were performed, this dataset consists of 2296 ratings provided by 121 users on 1232 items, along with 12 contextual factors. The data is then pre-filtered by, first; discarding the incomplete rating records, which contain the missing context and genre values. Then only the rating records from the users who have rated more than 10 times are extracted. The basic statistical information of this pre-filtered data is shown in Table 3.

Finally, only 6 contextual factors that are proved to be useful for the preference prediction by this dataset, as determined by [18] are selected. These contextual factors are Daytype, Location, endEmo, dominantEmo, Mood, and Physical. More details of these contextual factors are shown in Table 4.

Table 3. Basic Sstatistic of Pre-Filtered LDOS-CoMoDa Dataset

Number of users	30
Number of items	1121
Number of ratings	1867
Maximum ratings per user	269
Minimum ratings per user	13
Average ratings per user	62.2

Table 4. Contextual Factors along with Their Basic Information

Contextual Factors	**Description**	**Conditions**
Daytype	the types of the day the movie was seen	working day, weekend, holiday
Location	the places where the movie was seen	home, public place, friend's house
endEmo	the user's emotions after watching the movie	sad, happy, scared, surprised, angry, disgusted, neutral
dominantEmo	the user's dominant emotions toward the entire movie	sad, happy, scared, surprised, angry, disgusted, neutral
Mood	the user's feelings for the movie	positive, neutral, negative
Physical	the states of the user when the movie was seen	healthy, ill

To evaluate the performance of the models, 70% of the data was used as the training set and the remaining 30% was used as the test set.

4.2 Performance Evaluation

In this section, the results of the proposed model in different configurations are presented, compared with the model-based RS techniques (Matrix Factorization [15] and Bayesian Flexible Mixture Model [23]) and Context-Aware Collaborative Filtering [2] techniques. The abbreviation and the description of each method are shown in Table 5.

Table 5. Abbreviations and Description for the Models

Methods		Description
Model-based (ignore context)	MF	Standard Matrix Factorization
	BFMM	Bayesian Flexible Mixture Model
CACF		Context-Aware Collaborative Filtering. Using exhaustive search as feature selection technique.
Proposed	BFMM-CR (main model)	BFMM incorporating with the relevant contextual factors to both user classes and item classes. Using BPSO as feature selection technique.
	BFMM-CA	BFMM incorporating with all contextual factors to both user classes and item classes (no feature selection technique is applied).
	BFMM-CU	BFMM incorporating with the relevant contextual factors to user classes (ignore the contextual information to item classes). Using BPSO as feature selection technique.
	BFMM-CI	BFMM incorporating with the relevant contextual factors to item classes (ignore the contextual information to user classes). Using BPSO as feature selection technique.

The experiments are divided into two sections. First, the results of the accuracy based models, which is optimized by using accuracy alone are presented in section 4.2.1. Then, the results of the accuracy-diversity based models which are optimized by considering the trade-off between accuracy and diversity are shown in section 4.2.2.

4.2.1 Accuracy based

First, the performances of the models are presented in term of prediction accuracy. Figure 13 shows the RMSE and the prediction coverage of each method mentioned in Table 5. It appears that the BFMM-CR (the main proposed model) provides the lowest RMSE, followed by CACF, BFMM-CI, BFMM-CU, BFMM-CA, MF, and BFMM respectively. On the other hand, MF and BFMM are able to provide the highest prediction coverage, followed by BFMM-CU, BFMM-CI, BFMM-CR, BFMM-CA, and CACF respectively. For the better explanation, the results of the main model (BFMM-CR) is separately explained, compared with each method as the following.

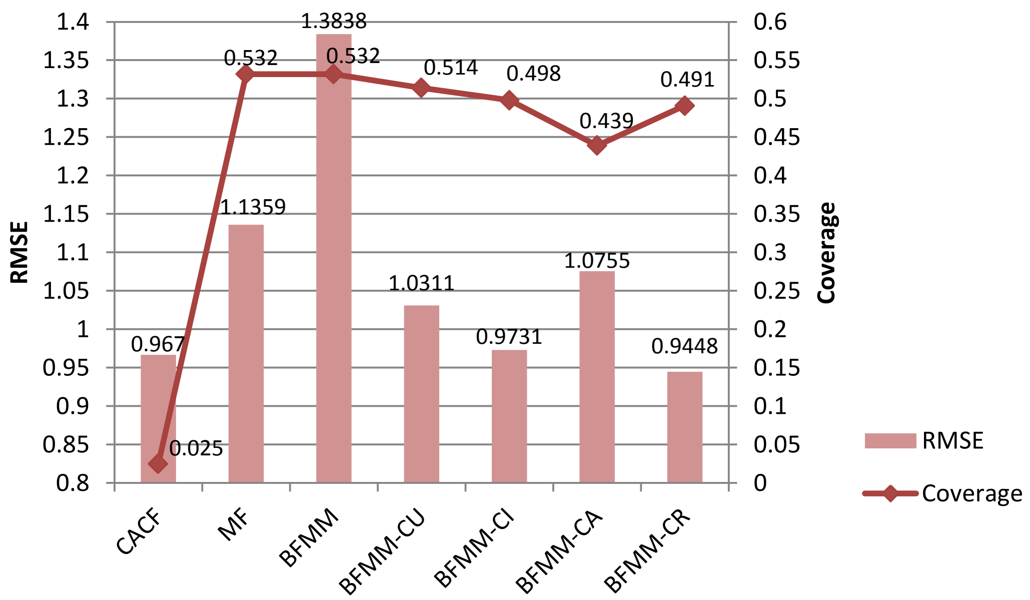


Figure 13. Comparison of RMSEs and Prediction Coverage

i) Comparison with CACF

First, the proposed model is compared with the primary approach in context-aware recommender systems: the Context-Aware Collaborative Filtering (CACF). The pre-filtering method selected only the ratings which are given under the active user's contextual situation and then apply the standard 2D CF-based technique to make the predictions. The exhaustive search is applied on all possible combinations of the contextual factors (since there are 6 contextual factors, all possible combinations are $2^6 = 64$ combinations) to find the one that provides the most accurate result (lowest RMSE), and used such combination to pre-filter the ratings before the prediction. As compare to the other models, although CACF is able to produces the second lowest RMSE (after the BFMM-CR), it provides the lowest numbers of prediction coverages: only 2.5% of the test data.

ii) Comparison with model-based techniques

The proposed model is now compared with the model-based techniques without contextual information: MF and BFMM. The results show that the proposed model in all configurations yield the higher accuracies compared to both MF and BFMM. However, MF and BFMM are able to produce the highest prediction coverage among the other methods.

iii) Comparison with BFMM-CA

Now, it is time to analyze the comparison among the proposed models in different configurations: the BFMM-CA and BFMM-CR. The BFMM-CA is the case where all of the contextual factors are being considered in the optimization step, i.e. no feature selection algorithm (BPSO) is applied. By comparing to the main configuration, BFMM-CR, which considers only the relevant contextual factors; the later performs better. Also, by identifying the relevant contextual factors to each user class and each item class differently, the BFMM-CR is able to provide higher in both accuracy and number of coverages.

iv) Comparison with BFMM-CU and BFMM-CI

Finally, the performances of BFMM-CU, which context is related only to user classes, and BFMM-CI, which context is related only to item classes are inspected. The results show that the later achieve a higher RMSE than the prior. However, both of them are less accurate than BFMM-CR, which context is related to both user and item classes. Also, these three configurations provide no significant different in term of the prediction coverage, and their coverages are almost the same level as the model-based approaches that do not incorporate contextual information.

4.2.2 Accuracy-Diversity based

In the previous section, the performances of the models which are optimized using prediction accuracy alone are presented. This section, the performances of the proposed models when considering the tradeoff between accuracy and diversity are analyzed. By using the proposed fitness function defined in Equation (42), the learning curves of the BPSO of the BFMM-CR is shown by Figure 14.

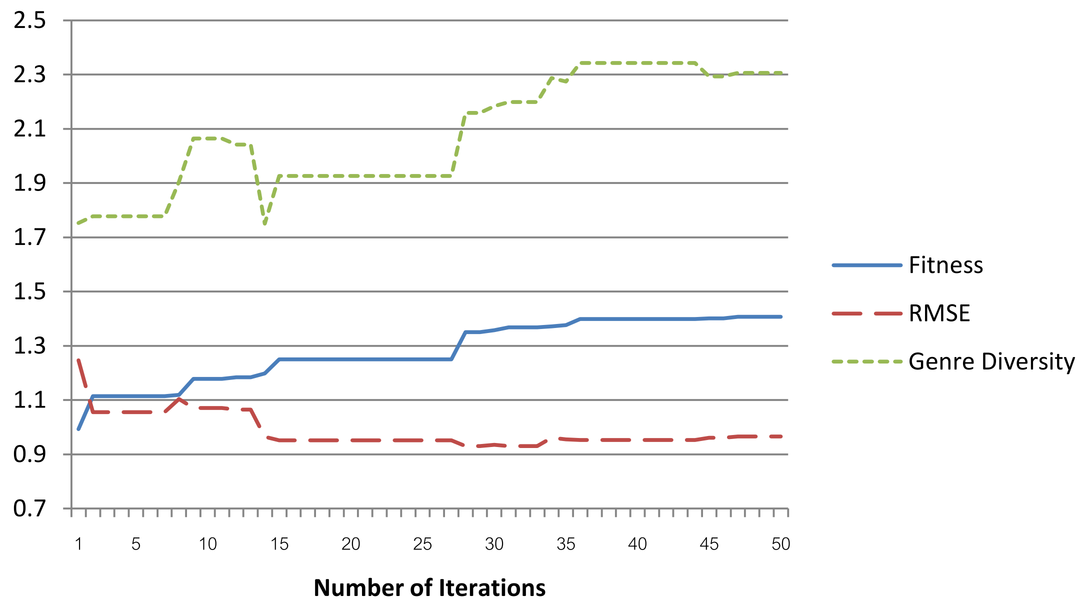


Figure 14. The Learning Curves of the Fitness values, RMSEs and Genre Diversity for BFMM-CR

On Figure 14, it can be seen that the fitness values increase simultaneously from the first iteration, and seem to reach the convergence point around iteration 35 of BPSO. When separately inspect the values of RMSE and genre diversity, which are used to calculate the fitness value for each iteration; it can be noticed that the RMSE values are significantly reducing in the first half iterations of BPSO, and remaining constant in the later half iterations. In contrast, the genre diversity values make a small increasing in the first half iterations, and rise up significantly in the middle to the last iterations of BPSO. This proves that the accuracy and diversity have the inverse relationship to each other.

The learning curves of BFMM-CU and BFMM-CI are also analyzed as shown by Figure 15 and Figure 16. However, their learning patterns are not recognizing like BFMM-CR. In BFMM-CU, both RMSE and genre diversity values are unstable: the RMSE values have almost not been optimized, while the genre diversity values only make a great increasing on the first 5 iterations of BPSO. As in BFMM-CI, it received the lowest genre diversity among the other two models. Also, both BFMM-CU and BFMM-CI produced the lower final fitness values compared to BFMM-CR. This means the BFMM-CR is more appropriate for modeling the trade-off between the accuracy and the diversity than BFMM-CU and BFMM-CI.

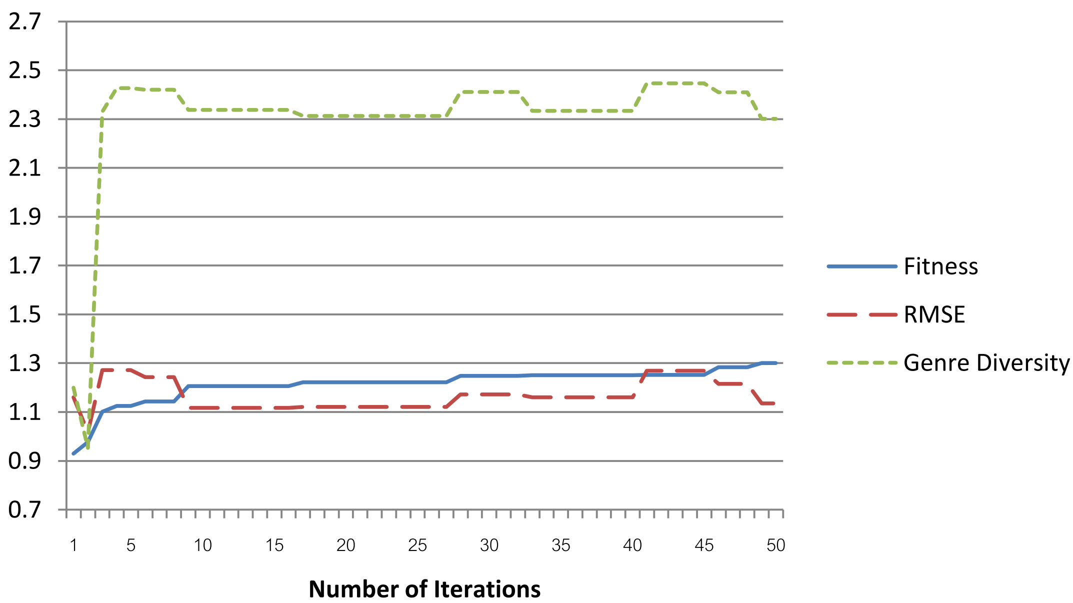


Figure 15. The Learning Curves of the Fitness values, RMSEs and Genre Diversity for BFMM-CU

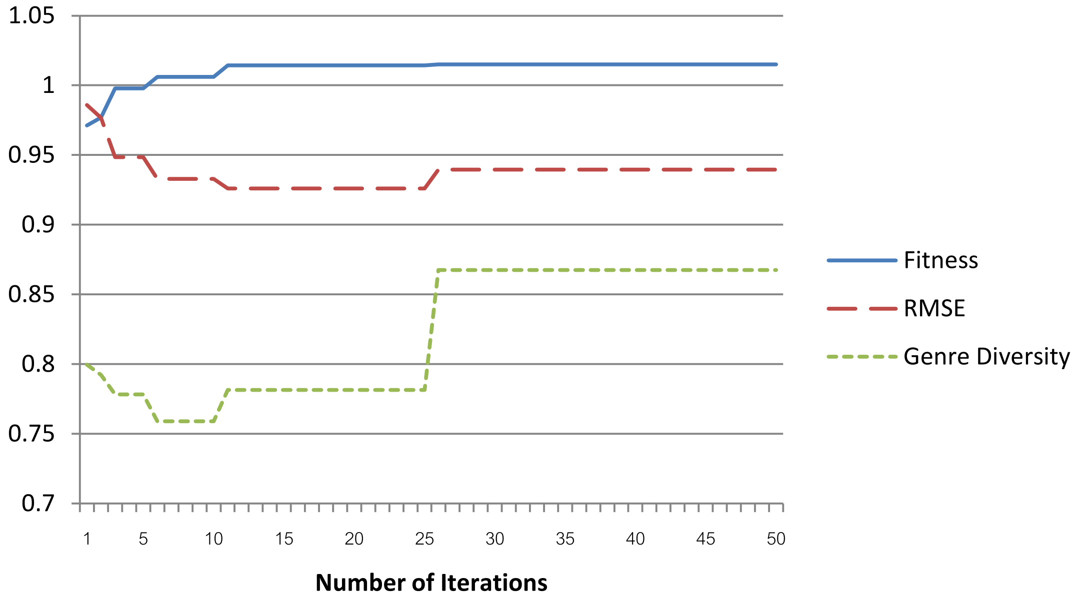


Figure 16. The Learning Curves of the Fitness values, RMSEs and Genre Diversity for BFMM-CI

4.3 Relevant Contextual Factors

In this section, the results of identifying the relevant contextual factors for each method are presented. First, the results of the accuracy-based contextual factors are shown. Then, the results of contextual factors that are learned by considering the trade-off between accuracy and diversity are shown.

Table 6. Relevant Contextual Factors for Accuracy based

Methods		Contextual Factors						RMSE	Coverage (%)
		Daytype	Location	endEmo	domEmo	Mood	Physical		
CACF		✓	✓	✓	✓	✓	✓	0.967	2.5
BFMM-CR	t_{U1}			✓			✓	0.9448	49.1
	t_{U2}			✓	✓	✓	✓		
	t_{U3}	✓	✓		✓				
	t_{I1}	✓		✓	✓		✓		
	t_{I2}	✓		✓	✓				
BFMM-CU	t_{U1}			✓		✓	✓	1.0311	51.4
	t_{U2}	✓	✓	✓			✓		
	t_{U3}	✓	✓	✓		✓			
BFMM-CI	t_{I1}	✓	✓	✓			✓	0.9731	49.8
	t_{I2}	✓		✓	✓		✓		

Table 6 shows results of the relevant contextual factors obtained by the accuracy based models. To acquire the most accurate prediction, CACF selected all of the contextual factors as its relevant context. For BFMM-CU, all of the contextual factors except domEmo are incorporated, while all except Mood are incorporated into BFMM-CI. Also, in both BFMM-CU and BFMM-CI, the context endEmo is relevant to all user classes and all item classes. This means that endEmo has an important role on producing accurate prediction, while domEmo and Mood have no effect on the accuracy of BFMM-CU and BFMM-CI, respectively. For BFMM-CR, the context endEmo and domEmo are relevant to the most of the classes, meaning they have the greatest effect on the accuracy. On the other hand, the Location and Mood are not likely related to the accuracy since they are only relevant to some user classes.

Table 7. Relevant Contextual Factors for Accuracy-Diversity based models

Methods		Contextual Factors						Fitness
		Daytype	Location	endEmo	domEmo	Mood	Physical	
BFMM-CR	t_{U1}				✓			1.4067
	t_{U2}	✓		✓	✓	✓		
	t_{U3}	✓	✓	✓	✓		✓	
	t_{I1}			✓		✓	✓	
	t_{I2}			✓	✓	✓		
BFMM-CU	t_{U1}	✓	✓	✓	✓	✓	✓	1.3006
	t_{U2}						✓	
	t_{U3}	✓	✓	✓			✓	
BFMM-CI	t_{I1}	✓	✓	✓			✓	1.0151
	t_{I2}	✓		✓	✓		✓	

Finally, the results of relevant contextual factors retrieved from accuracy-diversity based optimization are presented in Table 7. As compared to the accuracy based, the BFMM-CI has exactly the same set of relevant contextual factors to each item class; which means the diversity has no effect on selecting the relevant context. On the other hand, in BFMM-CU; the context domEmo is changed from irrelevant in accuracy based to be relevant in accuracy-diversity based, meaning it has a part on making more diverse recommendations. Finally, for BFMM-CR, the contextual factors endEmo and domEmo are still relevant to most of the user and item classes (although not the same classes), while the location is still relevant to only one (and the same) user class like in the accuracy based model. This means that endEmo and domEmo play an important role on making the accurate and diverse recommendation results. In contrast, not only the location is less related to the accuracy, it also has no effect on producing the diverse recommendations.

CHAPTER V
DISCUSSION

In this chapter, the evaluation results presented in chapter 4 are discussed. For the better explanation, the performance of each model is separately discussed in the various aspects.

5.1 CACF

From the accuracy results shown by Figure 13, it appears that CACF provide an accurate result with a very low coverage. As presented in Table 6, to produce the highest accurate result, CACF needs to considered as many contextual factors to be relevant as possible (in this case, it considered all 6 contextual factors as relevant), and resulting in very poor prediction coverage. In order to predict the rating on an item to an active user in a specific contextual situation using CACF method, it is required that the target item must be rated before in the training data at least once, under the same contextual situation. For example, to predict the rating on movie Titanic on the contextual situation <weekend, home, surprised, sad, positive, healthy>, it is required that Titanic must be rated before in the training data with this contextual situation. This becomes rather difficult for the small or sparse dataset that each item have been rated few times. Moreover, by using CF-based for prediction; even if the target item has been rated in the same contextual situation, the users who rated that item must be similar to the active user. If the ratings from the users who are not similar to an active user are used for the prediction, the predicted rating might be inaccurate.

Although it is possible for CARS to increase its prediction coverage by ignoring some relevant contextual factors in the prediction, doing so would reduce the prediction accuracy. On the other hand, using too many relevant contextual factors may lead to the overfitting, and degrade both accuracy and coverage of the model as well.

Fortunately, the proposed model can produce high prediction accuracy, while maintaining high prediction coverage. This is because instead of identifying the relevant contextual factors of the entire dataset, the proposed method tries to identify the set of relevant contextual factors to each user class and each item class differently. This can help relaxing the constraint that used for selecting the rating data on the learning step—resulting in more coverage in predictions. Moreover, in the proposed model, the items are independent from the contextual situation for making prediction. This means that there is no requirement an item must be rated in the specific contextual situation to make a prediction. The proposed model requires only two separated conditions: first, the target item has been rated in the training data at least once; and second, the relevant contextual conditions to each user class and each item class must be rated at least once in the training data.

For example, considering the same example as CACF, i.e. to predict the rating of the movie Titanic in the contextual situation <weekend, home, surprised, sad, positive, healthy> for an active user. It is required that, first, Titanic must be rated at least one time in the training data. Second, there must be at least one rating record in the training data that contains a set of the relevant contextual conditions to each user class and item class. As in Table 6 in Chapter 4, the relevant contextual factors to class t_{U1} of BFMM-CR are endEmo and Physical; therefore, there must be at least one rating record that contains the contextual conditions <surprised, healthy>.

5.2 Model-based approaches

The performance of the two model-based approaches: MF and BFMM is now discussed. The results from Figure 12 shown that BFMM is the lowest accurate method, followed by MF. However, both of them provided the highest prediction coverage compared to the context-aware models. This proved that the prediction accuracy can be improved by incorporating the contextual information into the model. However, since there is no contextual constraint to filter the rating data for the prediction, MF and BFMM are, therefore; providing the higher number of predictable ratings compared to the context-aware methods, which filter the rating data by context on making prediction. On the other side, even considering the contextual information, the proposed models (BFMM-CR, BFMM-CU and BFMM-CI) are still able to produce almost the same number of coverages compared to these non-context models. This is due to the benefit of identifying the relevant contextual factors among the classes differently, which helps relaxing the contextual constraints for filtering the data; the same reason as mentioned in section 5.1.

5.3 BFMM-CA

From Figure 13, it can be noticed that among the other context-aware model, the BFMM-CA provided the poorest performance in both accuracy and prediction coverage. This proves that considering all of the contextual factors—both relevant and irrelevant, can significantly degrade the prediction accuracy and affects the coverages. In BFMM-CR, BFMM-CU and BFMM-CI, they are all shown that the prediction accuracy can be improved if the contextual factors are carefully analyzed and incorporated into the model. Notice that both BFMM-CA and BFMM-CR incorporated all of 6 contextual factors into their models. However, the different is that; in BFMM-CA, every user classes and every item classes have the same set of all contextual factors as their relevant context. In contrast, in BFMM-CR, each user class and each item class have their relevant contextual factors different from the others. This is the reason that BFMM-CA provided the lower number of coverages than BFMM-CR, since it requires at least one rating record in the training data that matches all 6 contextual conditions of the test rating record.

5.4 BFMM-CR, BFMM-CU and BFMM-CI

The accuracy result from Figure 13, shown that considering the relations of the relevant context to both users and items like BFMM-CR provide a better accuracy than considering the relations of context to the users alone (BFMM-CU) or items alone (BFMM-CI). This proves that, context has relation to both users and items, not only the item like the other works [4], [5] and [10] presented.

Moreover, BFMM-CR also performs better when considering the trade-off of the accuracy and diversity. In the optimization of BFMM-CU, the accuracy (RMSE) is dominated by the genre diversity in the very first iterations, leading to poor convergence rate in both accuracy and diversity in the rest of the optimization. This is because the genre diversity is calculated based on: 1) the number of times each genre in Top-N recommendation list is assigned to each user class ($N_{t_u,g}$), and 2) the number of times an active user is assigned to each user class ($N_{t_u,u}$). Notice that these two numbers are depended mainly on the number of each user class. In BFMM-CU, which considered the contextual information only on the reassignment of the user classes, can create a very distinctive in number of each user class in the dataset—leading to more genre diversity. In contrast, the reassignment of user classes is not related to any contextual information on BFMM-CI. On the other hand, the BFMM-CR which considered the contextual information in the reassignment of both user classes and item classes can help adjusting the appropriate values of genre diversity. As shown in Figure 14, using the proposed fitness function (Equation (42)), the first half of the optimization is guided by the accuracy. After the accuracy is in the acceptable level, the latter half of the optimization is guided by diversity: to create more diverse recommendation result.

In the aspect of prediction coverage, these three models provided almost the same number of coverages since they are all finding the relevant contextual factors to each user class and/or each item class. Also, their coverages are very close to the coverages of the non-context models (as explained in section 5.2). The coverage around 50% of the test data might not be very high, but it is normal for this dataset since most items have been rated only one time. When the dataset has been split into training and test data, there is very highly chance that the items on test data are not stored in the training data, making the ratings for those items unpredictable.

Now it is time to discuss the results of the relevant contextual factors on both accuracy based model, and accuracy-diversity based model.

Start with BFMM-CU, as shown in Table 6, it can be inferred that there are three kinds of user classes: the class where endEmo, Mood, and Physical are relevant to the ratings; the class where Daytype, Location, endEmo, and Physical are relevant; and the class where Daytype, Location, endEmo, and Mood are relevant. Each user can be classified into all these three classes at the same time with different proportion. If he is more likely to be in the first class, meaning that he uses the emotion after he watched the movies, his mood when he watched the movies, and his physical condition at the time he watch the movies to determine the rating he gives to the movies. Notice that the context domEmo is relevant to none of the user classes, meaning that it has no effect on the ratings from the users. However, as presented in Table 7, the context domEmo is changed from irrelevant in accuracy based to be relevant in accuracy-diversity based. This means that although domEmo have no effect on the ratings on the accuracy-based model, it has an effect when considering both accuracy and diversity of the recommendations.

By inspecting the relevant contextual factors of BFMM-CI as shown in Table 6, it can also be inferred that there are two kinds of item classes: the class where Daytype, Location, endEmo, and Physical are relevant to the ratings; and the class where Daytype, endEmo, domEmo and Physical are relevant. As compared to Table 7, it can be seen that BFMM-CI have the same set of relevant contextual factors in both accuracy based and accuracy-diversity based model. This means the genre diversity does not have effect on deciding the relevant contextual factors for item classes. Because the values of genre diversity in BFMM-CI is very small and is dominated by the RMSEs as shown in Figure 16, only accuracy alone has a role on judging the relevant contextual factors.

Finally, for BFMM-CR; the context endEmo and domEmo are relevant to most of the classes, in both accuracy based and accuracy-diversity based models. Therefore, these two contextual factors should always be kept in the model in order to produce the accurate but diverse recommendation results. In contrast, the context Location is related to only one and the same user class in both accuracy based model and accuracy-diversity based model. Therefore, if it necessary to reduce the contextual dimensions on BFMM-CR, the context Location should be the first one to be discarded.

CHAPTER VI
CONCLUSION

This work is consisted of two main parts. In the first part, a novel way to incorporate the contextual information into a latent probabilistic model is proposed. This model has the flexibility to adjust the effects of context on users and items based on various context–user–item relations to suit specific situations. In the latter part, the optimization technique for identifying the relevant context to the user classes and item classes for the context-aware latent probabilistic model is proposed. The optimizations are done by considering the trade-off between accuracy and diversity metrics. From the experimental results comparing the proposed models with previous well-known recommendation techniques, it can be concluded that

- The context-aware recommendation techniques produce a better recommendation results than the typical recommendation that do not consider the contextual information. From the experiment, even the memory-based approach in CARS like CACF can make more accurate prediction than the model-based approaches like MF and BFMM, which do not consider contextual information.

- The model-based CARS methods outperform the memory-based CARS methods. As in the experiment, all of the proposed model-based CARS (BFMM-CR, BFMM-CA, BFMM-CU, and BFMM-CI) apparently provided higher prediction coverages than the memory-based CARS like CACF. Furthermore, the BFMM-CR also provided more accurate prediction results than the CACF method.
- The performance of the model can be improved if the relations of relevant context to both users and items are carefully considered. By comparing the performance of various context-aware models, it has been shown that:
    - First, all of the proposed models which exploit the relevant contextual factors (BFMM-CR, BFMM-CU, and BFMM-CI) are able to

provide almost the same number of coverages as the non-context model-based approaches.

- From the experiment, the BFMM-CR (identifying the relevant contextual factors to both user classes and item classes) provided better accuracy performance than the BFMM-CU (identifying the relevant contextual factors to user classes alone) and BFMM-CI (identifying the relevant contextual factors to item classes alone).
- Also, the BFMM-CR is more appropriate for modeling the trade-off between the accuracy and the diversity, compared to BFMM-CU and BFMM-CI.

Therefore, BFMM-CR is able to provide the final recommendation results with the variety of items that satisfying the users' personal interests.

REFERENCES

[1] Adomavicius, G., & Kwon, Y. (2012). Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 896–911. http://doi.org/10.1109/TKDE.2011.15

[2] Adomavicius, G., Sankaranarayanan, R., Sen, S., & Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, *23*(1), 103–145. http://doi.org/doi: 10.1145/1055709.1055714

[3] Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., … Schwaiger, R. (2011). InCarMusic: Context-aware music recommendations in a car. *Lecture Notes in Business Information Processing*, *85 LNBIP*, 89–100. http://doi.org/10.1007/978-3-642-23014-1_8

[4] Baltrunas, L., Ludwig, B., & Ricci, F. (2011). Matrix Factorization Techniques for Context Aware. *Proceedings of the Fifth ACM Conference on Recommender Systems*, 301–304. http://doi.org/10.1145/2043932.2043988

[5] Baltrunas, L., & Ricci, F. (2009). Context-based splitting of item ratings in collaborative filtering. *Proceedings of the Third ACM Conference on Recommender Systems - RecSys '09*, 245. http://doi.org/10.1145/1639714.1639759

[6] Burke, R. (2007). Hybrid web recommender systems. *The Adaptive Web*, 377–408. http://doi.org/10.1007/978-3-540-72079-9_12

[7] Campos, P. G., Fernández-Tobías, I., Cantador, I., & Díez, F. (2013). Context-aware movie recommendations: An empirical comparison of pre-filtering, post-filtering and contextual modeling approaches. *Lecture Notes in Business Information Processing*, *152*, 137–149. http://doi.org/10.1007/978-3-642-39878-0_13

[8] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–38. http://doi.org/10.1.1.133.4884

[9] Dey, A. K., & Abowd, G. D. (1999). Towards a Better Understanding of Context and Context-Awareness. *Computing Systems*, *40*(3), 304–307. http://doi.org/10.1007/3-540-48157-5_29

[10] Hariri, N., & Burke, R. (2012). Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns. *Proceedings of the 6th ACM Conference on Recommender Systems - RecSys '12*, 131–138. http://doi.org/10.1145/2365952.2365979

[11] Karatzoglou, A., Amatriain, X., & Oliver, N. (2010). Multiverse Recommendation : N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. *Context*, 79–86. http://doi.org/10.1145/1864708.1864727

[12] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, *4*, 1942–1948 vol.4. http://doi.org/10.1109/ICNN.1995.488968

[13] Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, *5*, 4–8. http://doi.org/10.1109/ICSMC.1997.637339

[14] Koren, Y. (2009). Collaborative filtering with temporal dynamics. *Knowledge Discovery and Data Mining {KDD}*, 447–456. Retrieved from http://doi.acm.org/10.1145/1557019.1557072

[15] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, *42*(8), 42–49. http://doi.org/10.1109/MC.2009.263

[16] Košir, A., Odić, A., Kunaver, M., Tkalčič, M., & Tasič, J. F. (2011). Database for contextual personalization. *Elektrotehniški Vestnik*, *78*(5), 270–274. Retrieved from http://ev.fe.uni-lj.si/5-2011/Kosir.pdf

[17] McNee, S. M., Riedl, J., & Konstan, J. a. (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, 1101. http://doi.org/10.1145/1125451.1125659

[18] Odić, A., Tkalčič, M., Tasič, J. F., & Košir, A. (2012). Relevant context in a movie recommender system: Users' opinion vs. statistical detection. *CEUR Workshop Proceedings*, *889*.

[19] Panniello, U., Tuzhilin, A., & Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, *24*(1-2), 35–65. http://doi.org/10.1007/s11257-012-9135-y

[20] Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., & Pedone, A. (2009). Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. *Proceedings of the Third ACM Conference on Recommender Systems - RecSys '09*, 265. http://doi.org/10.1145/1639714.1639764

[21] Pearson, K. (2009). Notes on the History of Correlation. *Biometrika*, *13*(1), 25–45. http://doi.org/10.1093/biomet/13.1.25

[22] Pu, P., & Chen, L. (2010). A user-centric evaluation framework of recommender systems. *CEUR Workshop Proceedings*, *612*, 14–21. http://doi.org/10.1145/2043932.2043962

[23] Samatthiyadikun, P., Takasu, A., & Maneeroj, S. (2012). Multicriteria collaborative filtering by Bayesian model-based user profiling. *Proceedings of the 2012 IEEE 13th International Conference on Information Reuse and Integration, IRI 2012*, 124–131. http://doi.org/10.1109/IRI.2012.6303000

[24] Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. *Recommender Systems Handbook*, 257–298. http://doi.org/10.1007/978-0-387-85820-3_8

[25] Si, L., Jin, R., & University, C. M. (2003). Flexible mixture model for collaborative filtering. *Machine Learning-International Workshop*. Retrieved from http://www.aaai.org/Papers/ICML/2003/ICML03-092.pdf

[26] Sitkrongwong, P., Maneeroj, S., & Takasu, A. (2013). Latent probabilistic model for context-aware recommendations. *Proceedings - 2013 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2013*, *1*, 95–100. http://doi.org/10.1109/WI-IAT.2013.14

[27] Vargas, S., Baltrunas, L., Karatzoglou, A., & Castells, P. (n.d.). Coverage , Redundancy and Size-Awareness in Genre Diversity for Recommender Systems. http://doi.org/10.1145/2645710.2645743

[28] Zheng, Y., Burke, R., & Mobasher, B. (2012). Optimal feature selection for context-aware recommendation using differential relaxation. *CEUR Workshop Proceedings*, *889*.

[29] Zheng, Y., Burke, R., & Mobasher, B. (2012). Differential context relaxation for context-aware travel recommendation. *Lecture Notes in Business Information Processing*, *123 LNBIP*, 88–99. http://doi.org/10.1007/978-3-642-32273-0_8

[30] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., & Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, *107*(10), 4511–4515. http://doi.org/10.1073/pnas.1000488107

APPENDIX

VITA

Padipat Sitkrongwong was born on February 6th 1991 in Suphan Buri, Thailand. He received his bachelor’s degree in Computer Science from Chulalongkorn University in 2013. . He is currently a graduate student in Computer Science and Information Technology in Chulalongkorn University.

Publication:

“Latent probabilistic model for context-aware recommendations”

Proceedings of 2013 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2013, vol. 1, pp. 95–100, 2013.