Music Recommendation Based on Multiple Contextual Similarity Information

Chih-Ming Chen*, Ming-Feng Tsai*, Jen-Yu Liu†, Yi-Hsuan Yang†
*Department of Computer Science & Program in Digital Content and Technology
National Chengchi University, Taipei 11605, Taiwan
Email: {g10018, mftsai}@cs.nccu.edu.tw
†Research Center for Information Technology Innovation
Academia Sinica, Taipei 11564, Taiwan
Email: {ciaua, yang}@citi.sinica.edu.tw

***Abstract*—This paper proposes a music recommendation approach based on various similarity information via Factorization Machines (FM). We introduce the idea of similarity, which has been widely studied in the filed of information retrieval, and incorporate multiple feature similarities into the FM framework, including content-based and context-based similarities. The similarity information not only captures the similar patterns from the referred objects, but enhances the convergence speed and accuracy of FM. In addition, in order to avoid the noise within large similarity of features, we also adopt the grouping FM as an extended method to model the problem. In our experiments, a music-recommendation dataset is used to assess the performance of the proposed approach. The datasets is collected from an online blogging website, which includes user listening history, user profiles, social information, and music information. Our experimental results show that, with various types of feature similarities the performance of music recommendation can be enhanced significantly. Furthermore, via the grouping technique, the performance can be improved significantly in terms of Mean Average Precision, compared to the traditional collaborative filtering approach.**

I. INTRODUCTION

Similarity is an important concept in recommendation. Given the favorite items of a user, it is sensible to recommend the other items similar to those favorite ones. Similarity between items can be measured in several ways, and different methods in measuring similarity can be complementary to one another in practice. For example, for music recommendation, some users prefer songs similar in melody, while others prefer songs similar in lyrics. The more information we have regarding different aspects of similarity, the more likely we are able to give successful recommendation.

If similarity is measured in terms of the number of people who share the same taste regarding the items, the resulting model can be considered as a *collaborative filtering* (CF)-based model. On the other hand, if similarity is measured in terms of the affinity of the items in a feature space, the resulting model is usually known as *content-based* (CB) model. Hybrid models that blend the aforementioned two models have also been studied in the literature. In particular, Factorization Machine (FM) has emerged in recent years as a promising framework for hybrid recommendation. With proper features, FM is able to mimic many state-of-the-art CF/CB-based algorithms.

Under the FM framework, it is possible to exploit every co-occurrence pattern among items to capture more information. Moreover, representing similarity in the form of a matrix can be more informative than representing each item as a feature vector, because the latter requires an additional process to extract similarity information from the feature vectors, an operation which is performed only implicitly by FM.

Music preference is not only affected by personal factors of the listener and the musical factors of the music items; it is also highly dependent on the context of music listening. For example, people listen to different music when being in an office or when exercising; when feeling blue or when being in a happy mood. Therefore, it is important to consider contextual information for better recommendation performance. This study also features the use of multiple similarity information computed from the contextual factors of music listening.

From technical point of view, FM models the global bias, feature biases and weights of the interactions among all the features, including vector-based and matrix-based ones. Therefore, it is likely that some noisy information will be mixed in the final prediction model. To remedy this, we propose to adopt a grouping technique to remove unnecessary interactions. In other words, we divide the features into distinct group and only account for interactions among the features between the different groups. In this way, noises inherent from unnecessary interactions can be largely eliminated. Our evaluation shows that such grouping technique is in particular important when one considers matrix-based features as similarity matrices, due to the increase in the number of features (and accordingly the number of potential unnecessary interactions). Although there are multiple ways features can be grouped, our result shows that there are some guidelines in finding a good grouping.

In the experiments, a dataset crawled from a real-world social blogging website, LiveJournal [1], as it contains rich contextual information that is entered by users spontaneously in their day-to-day lives [1], [2]. The features are extracted from user profiles and music characteristics such as geographic information and audio information, showing that similarity computation can be easily applied to most kinds of features. Since some interactions between features provide little information, we generate different grouping methods to examine whether the grouping technique can improve the performance.

[1] http://www.livejournal.com/

Finally we conduct experiments with different parameters. The experimental results show that similarity information significantly enhance the recommendation performance. Furthermore, via grouping factorization machine, the performance can be further improved to 0.52 in terms of Mean Average Precision with p-value less than 0.01.

II. Related Work

Recommender systems are widely deployed in commercial business, with collaborative filtering (CF) being one of the most popular models . CF models filter out the useless information and keep similar patterns to predict user behavior. More recently, machine-learning techniques provide a promising way to perform recommendation. In this section, we survey on the related studies from the different perspectives.

A. Contextual Recommender System

Traditional recommendation methods can be divided into two main categories: CF and CB. Many famous commercial recommendation systems are based on these methods, such as the ones used by Youtube or Amazon [3], [4]. However, such methods are limited due to the difficulty of incorporating contextual information, which is gaining increasing importance due to the rapid growth of information on the Internet.

In light of this, many methods have been proposed for the problem of contextual recommendation. For example, Meng *et al.* [5] investigated the individual preference and the interpersonal influence on online item adoption and recommendation. Yelong *et al.* [6] proposed a joint personal and social latent factor (PSLF) model that combines the state-of-the-art collaborative filtering and the social network modeling approaches for social recommendation. Kailong *et al.* [7] employed several interesting features form tweets, including social relation features, content-relevance features, tweets' content-based features and publisher authority features. From these prior arts, we can observe that most studies developing their model based on various types of features. In the competition of KDDCup 2012, Tianqi *et al.* [8] combined a variety of models by incorporating different features. Their result indicates the importance of the contextual features. Instead of focusing on the CF method, we propose an approach that integrates the advantages from the CF method, that is, incorporates the similarity information into the factorization model.

B. Music Recommender System

There are also many studies related to music recommendation. For example, Negar *et al.* [9] presented a context-aware music recommendation system that infers user's short-term music preference based on the most recent sequence of songs liked by the user using sequential data mining. Noam *et al.* [10] used a hierarchical track-album-artist-genre structure in modeling the biases of music items, and used music sessions to model session bias of users, showing the importance of bias modeling. Cai *et al.* [11] showed that emotion can be useful for matching songs to documents according to the audio and text content. Unlike these existing works, the contextual information considered in this work is mined from user-generated articles. Moreover, we use FM to study the effect of multiple types of features which are extracted from user profile, user-generated articles, geographic information, item characteristic and audio features.

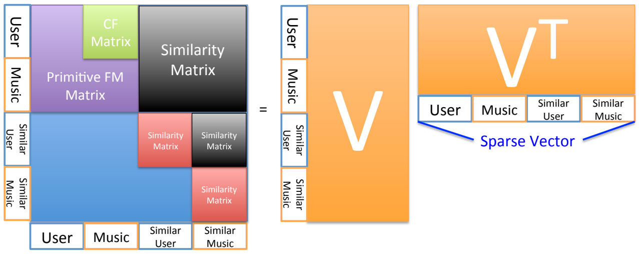


Fig. 1. Illustration of Our Proposed Approach vs. Factorization Machine

C. Factorization Model

The goal of a recommender system is to predict whether a user would like an object. In recent years, FM models has proven itself to be a competitive and flexible model for a variety of recommendation tasks [5], [8]. For example, Jason *et al.* [12] studied the joint problem of recommending items to a user with respect to a given query and introduced a factorized model for optimization. István *et al.* [13] took an MF-based approach with a simple rating-based predictor on the Netflix Prize Dataset.

It can be found that a common problem among FM-like models is the need to re-design the prediction model task by task. To solve this problem, Rendle described a generic FM framework called libFM [14], which is able to simulate many other successful models via factorization machine by feature engineering (i.e., by using corresponding features). As demonstrated in [14], libFM generalizes existing methods such as standard matrix factorization, Pairwise Interaction Tensor Factorization (PITF) and SVD++. Moreover, a system based on libFM has won the second title in a KDDcup competition [15]. Liangjie *et al.* [15] modified the original model to handle multiple aspects of the dataset at the same time. In contrast, in this work we aim at incorporating similarity information to libFM without major modification of its framework, thereby reserve the advantages of libFM.

III. Methodology

Figure 1 illustrates the main concept of incorporating similarity information into the FM framework. In general, a traditional CF-based matrix only keeps the records of user-to-item information, but FM factorizes this form to a multiplication of two feature vectors (i.e. V and V^T in Figure 1). Our proposed approach further integrates the similarity information with the framework to capture the similar patterns from the referred objects. Below we further describe the Factorization Machines and our proposed approaches.

A. Standard FM

Factorization Machines can act like most factorization models by feeding various types of features. It learns the weights of all interactions between the features. In general, a two-way factorization machine model can be defined as:

$$\hat{y}(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{l=1}^{n} \sum_{j=l+1}^{n} \hat{w}_{lj} x_l x_j, \tag{1}$$

where w_0 is the global bias, w_l is the weight of features x_l, and w_{lj} models the interaction of each pair of features. The interaction w_{lj} can be factorized into pairs of interaction parameters,

$$\hat{w}_{lj} = \sum_{f=1}^{\kappa} v_{lf} v_{jf}. \tag{2}$$

The parameter κ determines the model complexity. Rather than only using single parameter for each interaction, this way allows high quality parameters estimated by higher-order interactions under sparsity. Factorization Machine provides a promising framework for recommendation problem. Unlike the generic matrix factorization model, it can be easily used to conduct feature engineering. For more details of FM, please refer to [14].

B. Grouping FM

Factorization Machines provide a good framework for modeling the interactions between features, but sometimes similar type of features may cause confusion while learning, especially with a large number of features. Hence we can utilize the bag-of-feature concept to the standard factorization machine by grouping the features with similar characteristic. Therefore it can deal with the tasks in a more flexible way with different feature partitions. After removing the non-informative weights from the FM models, the original formula can be rewritten as:

$$\hat{y}(x) = w_0 + \sum_{i}^{n} w_i x_i + \sum_{l \in G(l)}^{n} \sum_{j \notin G(l)}^{n} x_l x_j \sum_{f=1}^{\kappa} v_{l,f} v_{j,f}, \tag{3}$$

where the x_l belongs to the group $G(l)$, and the mutual effect of x_l and x_j is dropped out while they are in the same group. By the grouping technique, it eliminates the unnecessary interactions such as the interaction between a user and the user's age is non-informative. The grouping technique not only speeds up the convergence of optimization but also provides a flexible way to construct different feature combinations. Note that the modified prediction function would be the same as the original one when every feature has its own group.

LibFM provides three major optimization criteria to learn the data: stochastic gradient descent [16] (SGD), alternating least-squares [17] (ALS) and Markov Chain Monte Carlo [18] (MCMC). In our experiments, the MCMC method is chosen because it can automatically learn the data without giving the external parameters such as learning rate [2] and the regularization term [3]. For MCMC, the gradient for the grouping Factorization Machine is derived as follows:

$$h_\theta(x) = \frac{\partial \hat{y}(x)}{\partial \theta} = \begin{cases} 1, & if\ \theta = w_0 \\ x_j, & if\ \theta = w_j \\ x_j \sum_{j' \notin G(j)} v_{j',f} x_j, & if\ \theta = v_{j,f} \end{cases} \tag{4}$$

C. Similarity Computation

Motivated by the strength and efficiency of CF method, we seek to combines the advantages with the factorization model. Since FM has a good framework for modeling the input features, we can directly extract the similarity information from the users and items. This is similar to CF methods, and can be easily embedded into a feature vector. In general, the utilized features are divided into following three types, and each type has its own computation method.

1) **ID Domain:** The ID variable is used to identify a target, and it only belongs to a specific target. For instance, *User ID* is in the ID domain, which means that each user has his/her own unique ID variable. Technically a similarity measurement is a function that computes the degree of similarity between a pair of targets, e.g. the similarity of listening histories of two users. Given two vectors of attributes, A and B, the similarity score is computed by the extended version of cosine similarity:

$$similarity = \frac{A \cap B}{|A|^{1-\alpha}|B|^{\alpha}}, \tag{5}$$

where $\alpha \in [0, 1]$ is a tuning parameter.

2) **Categorical Domain:** The categorical variable represents the extracted features from the user and item attributes such as the *User Age* and *Music Genre*. The similarity computation is also based on Equation 5.

3) **Real Value Domain:** If the attribute is already a number $\in \mathbb{R}$, such as *Audio Information*. The similarity score is calculated by the Euclidean distance. In general, for an n-dimensional space, the distance between feature vector q and feature vector p is:

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}. \tag{6}$$

For the ID domain, the function O represents the referred objects from target i and target j. For example, given the listening histories of two users the α determines whether the similarity score considers the amount of referred objects from another target or not. Take the following three users with the listening records as an example:

$$\begin{aligned} O(User_i) &= [1, 2, 3], \\ O(User_j) &= [1, 2, 3], \\ O(User_k) &= [1, 2, 3, 4]. \end{aligned}$$

Then $User_j$ is more similar to $User_i$ than $User_k$ based on the listening history while the $\alpha = 1$; on the other hand, they will get a same score while the $\alpha = 0$.

[2] The learning rate is a common parameter for controlling the learning steps.

[3] The regulation term is used to prevent the model from overfitting problem.

For the categorical indicators, because this kind of feature usually occurs in different objects, the function O will be the collection of referred objects for a target. Take the User Age as an example, if we want to know the similarity of listening history between 15-year-old users and 30-year-old users, the function O will collect all the songs of the users whose age is between 15 and 30.

For the real-value indicators, the feature vector is normalized by the standard score: $\frac{x-\mu}{\sigma}$, where μ is the mean of the population and σ is the standard deviation of the population. The score indicates how many standard deviations an observation is above or below the mean.

Finally suppose we have a set of similarity scores for a specific target and seek to embed them into a feature vector, a simple way is to directly index them with corresponding scores. However, the popular object generally contains more similar objects than the others. It may leads to an unbalance problem that unpopular objects are hard to get the similarity score. In order to take the balance issues into account, we only keep the top-k similar objects as the new score basis, and normalize the new vector of k values to 1:

$$\bar{s}_{ij} = \frac{s_{ij}}{\sum_{j'=1}^{n} |s_{ij'}|}. \tag{7}$$

The purpose of this step is to avoid the unbalance of similarity information. For example, $s(User_i) = (0, 0.8, 0.6)$ and $s(User_j) = (0.1, 0, 0.2)$, $User_i$ will have more probability of getting high scores because of the high values of the similarity vector.

IV. Experimental Setup

This section describes the experimental setup we employed to study the influence of different factors on the performance of music recommendation.

A. Evaluation Metric

We employed two metrics to evaluate the recommendation performance: the truncated mean average precision at k (MAP@k) and recall. For each user, let $P(k)$ denotes the precision at cut-off k:

$$AP(u, o) = \frac{\sum_{p=1}^{k} P(k) \times r_{uo(p)}}{I(u)}, \tag{8}$$

where $o(p) = i$ describes the item i is ranked at position p in the order list o, and r_{ui} means whether the user u has listened to song i or not($1 = yes, 0 = no$). MAP@k is the mean of the average precision scores for the top-k results:

$$MAP@k = \frac{\sum_{u=1}^{U} AP(u, o)}{U}, \tag{9}$$

where U is the total number of target users. Higher MAP@k indicates better recommendation accuracy.

Recall measures how many songs the user really likes are recommended by the automatic system. It is computed by:

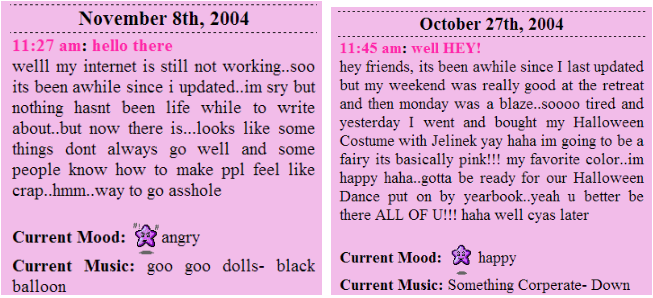
$$Recall = \frac{|\{Correct\ Songs\}| \cap |\{Returned\ Top\ k\ Songs\}|}{|\{Correct\ Songs\}|}. \tag{10}$$

High recall means that most of songs the user actually likes or listens to are recommended.

TABLE I. The feature sets considered in this work.

abbr.	Feature	Unique Index	Type
U	User ID	19,596	-
S	Song ID	30,260	-
H	Listening History	30,260	-
BY	Birth Year (of users)	100	Cb
LR	Live Region (of users)	208	Cb
M	Mood Tags (of users)	132	Cx
VAD	VAD values (of articles)	3	Cx
A	Artists (of songs)	5,175	Cb
Au	Audio Information	53	Cb
SR	Social Relation	674,932	Cx

Note: *P* denotes the feature of user profile, *Cb* denotes the content-based feature that are extracted from songs, and *Cx* denotes the context-based feature that are extracted from user.



November 8th, 2004

11:27 am: hello there

welll my internet is still not working..soo its been awhile since i updated..im sry but nothing hasnt been life while to write about..but now there is...looks like some things dont always go well and some people know how to make ppl feel like crap..hmm..way to go asshole

Current Mood: angry

Current Music: goo goo dolls- black balloon

October 27th, 2004

11:45 am: well HEY!

hey friends, its been awhile since I last updated but my weekend was really good at the retreat and then monday was a blaze..soooo tired and yesterday I went and bought my Halloween Costume with Jelinek yay haha im going to be a fairy its basically pink!!! my favorite color..im happy haha..gotta be ready for our Halloween Dance put on by yearbook..yeah u better be there ALL OF U!!! haha well cyas later

Current Mood: happy

Current Music: Something Corperate- Down

Fig. 2. Livejournal sample posts.

B. Dataset

Our experiments are performed on a real-world dataset collected from a well-known social blogging websites – LiveJournal. LiveJournal is unique in that, in addition to the common feature of blogging, each post is accompanied with a "Mood" column and a "Music" column so that users can write down their moods and songs in their minds while posting, as Figure 2 exemplifies. From LiveJournal, we crawled a total number of 1,928,868 listening records covering 674,932 users and 72,913 songs as an initial set. For the purpose of retaining enough number of data in the training and test sets for this study, we only considered users who have more than 10 listening records and discarded the records of the other users. This filtering resulted in the final set of 225,652 listening records (11.7% of the initial set) among 19,596 users and 30,260 songs.

For evaluation, we split the dataset for each user according to the following 80/20 rule: keeping full listening history for the 80% and the half of listening history for the remaining 20% users as the training data, and the other half of the remaining 20% users as the testing data. For each record, we randomly add 10 songs as negative records to construct the testing pool.

C. Feature

The structure of collected music dataset is depicted in Figure 3, as these factors affect how people choose the music. *Personal factors* indicate the characteristics that people would possess for a long period of time, such as age and gender. People with different levels of music background may appreciate music differently, which in turn affects music preference. *Musical factors* consist of the audio content, its profile, and even the artwork of the CD. People may choos a song because its melody or the singer. *Situational factors* include those that

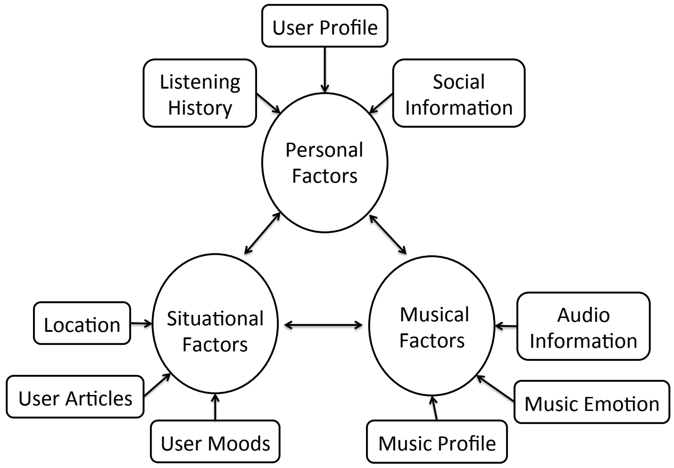


Fig. 3. The structure of LiveJournal dataset

TABLE II. AFFECTIVE NORMS FOR ENGLISH WORDS

Description	Valence	Arousal	Dominance
dream	6.73	4.53	5.53
eat	7.47	5.69	5.60
favor	6.46	4.54	5.67
good	7.47	5.43	6.41
hate	2.12	6.95	5.05

Note: 5 example words of ANEW dictionary.

persist for a short period of time such as when and where you listen to music, what you are doing and what your mood is. People often express their feelings through listening to music, and the user-generated article reflects their recent mood.

Table I summarizes the features used in the experiments, which are described in detail below.

1) Content-based Features: Content-based features refer to features that describe either the user or the item. For describing users, we have *Birth Year* (BY), *Live Region* (LR) and *Social Relations* (SR) features. The birth years for the users in our dataset fall in a window of 100 years. Moreover, the users are from 208 regions. We consider users who were born in the same year or users who were from the same region as similar. On the other hand, from LiveJournal we can obtain friendship and construct the social network among the users. This gives rise to the social relation based similarity matrix. People who are friends to one another are likely to share similar music taste.

For describing songs, we have *Artist* (A) and *Audio Information* (Au) features. The artist feature simply indicates the artist (among the 5,175 possible artists) of the songs. If two songs are sung/performed by the same artist, they are likely to be more similar. The audio features consists of 53 perceptual dimensions of music, including danceability, loudness, mode, and tempo. They are extracted by using the EchoNest API [4], a commonly used audio feature extraction tool developed in the field of music information retrieval [19]. We can measure the similarity between two songs in this 53-dimensional feature space.

TABLE III. EVALUATION RESULT OF CF-BASED ALGORITHMS

Model	MAP@10	Recall
Randomize	0.0578	0.1656
User-based CF	0.3668	0.4748
Item-based CF	0.3093	0.5115
SVD++	0.3506	0.4844
FM	**0.3817**	**0.5216**

2) Context-based Features: The user-generated articles are interesting context-based features in the dataset, but it may contains too many redundant words. Motivated by the idea of emotional matching, we convert the original content of an article into a vector of emotional words by referring to the dictionary of Active Norms for English Words (ANEW) [20], which provides a set of normative emotional ratings for English words. We retain the words which can be found in the ANEW dictionary and weight them by the TF-IDF weighting. Specifically, a word is scored by $tf(t,d) \times idf(t,d)$, where

$$tf(t,d) = \frac{f(w,d)}{max\{f(w,d) : w \in d\}}\,, \tag{11}$$

$$idf(t,d) = log\frac{|D|}{|\{d \in D : t \in d\}|}\,, \tag{12}$$

and D is the set of all articles. A term with higher score indicates that the term has a higher term frequency wight and a lower document frequency of the term in the whole collection of articles. In addition, the ANEW dictionary also provides a set of normative emotional ratings for English words. The emotional words are rated by Valence (or pleasantness; positive/negative active states) , Activation (or arousal; energy and stimulation level) and Dominance (or potency; a sense of control or freedom to act), the fundamental emotion dimensions found by psychologists [21]. Finally each word vector of articles is converted to valence, arousal, and dominance (VAD) values. For example, for the sentence "*I had a dream last night, I was eating a marshmallow*," the VAD values would be 14.2, 10.22, and 11.13, respectively, according to Table II. Moreover, we also collected the recent mood tags which are recent used by each user.

V. EXPERIMENTAL RESULTS

We conducted a series of experiments with different settings. First of all, we attempted to demonstrate the similarity information is effective on most kinds of features under the factorization model. Second, we compared the performance of the standard Factorization Machine with that of the grouping Factorization Machine, and then examined the effects of different feature combinations. Finally, we studied the sensitivity of the proposed method to the parameters

A. Similarity Approach

The similarity indicator can be represented as the categorical set domain as used in [17]. For instance, suppose that "*Alice* is similar to *Charlie* and *Sandy*", the corresponding similarity indicator may be the vector $z(Bob, Charlie, Sandy) = (0, 0.2, 0.8)$, where the sum of all values equals to 1 according to Equation 7.

[4] http://echonest.com/

TABLE IV. PERFORMANCE OF ID SIMILARITY

	LiveJournal Dataset	
Features	MAP@10	Recall
U + S	0.3816	0.5217
U + S + H	0.4409	0.5821
U + S + US	0.4310	0.5712
U + S + H + US	0.4427	0.5810
U + S + SS	0.4635	0.6194
U + S + H + SS	0.4897	0.6413
U + S + US + SS	0.4712	0.6251
U + S + US + SS + H	0.5021	0.6491

Note: For the feature abbreviation, please refer to Table I.

1) CF-based Recommendation: In the first step, we evaluated the performance on some well-known CF-based Recommendation algorithms to verify the strength of factorization machine. We compare FM with user-based CF, item-based CF, and a SVD-based approach using only the user-item matrix, which is a standard input to recommendation models. Note that context information or similarity information is not exploited in this comparison. Table III lists the result of these methods. As the table shows, the resulting MAP of all the CF-based approaches fall within 0.30–0.38. Among the four methods, FM performs the best. The performance difference between FM and other methods is significant under the t-test. This validates the effectiveness of FM. Therefore, we employed FM in the subsequent experiments.

Under the CF-based framework, there are two ID indicators: *User ID* and *Song ID*. Therefore, we can obtain the following similarity information according to Equation 5:

- **User Similarity (US):** Two users are similar if they listen to the same songs.
- **Song Similarity (SS):** Two songs are similar if they are listened by the same users.

Both of them are directly mined from the listening history. Therefore, they are always available for a standard recommendation problem. US is applied to users, whereas the SS is applied to items.

We evaluated the performance on every possible feature combination. As shown in Table IV, both the user similarity and the song similarity (U+S+US or U+S+SS) lead to significantly better result, comparing to the baseline U+S.

We have also implemented KNN-based FM of [14] by adding the listening history to libFM, as shown in from the second row of Table IV (i.e., U+S+H). It can be seen that the incorporation of listening history ('H') generally improves the result as well. Note that the SS feature is the top-k most similar music which is not extracted from listening history. If we compare H, US, and SS, SS achieves the highest MAP@10 (0.4635), showing that the similarity approach is more effective than the KNN approach is. Moreover, KNN approach may fail when the amount of listening histories is limited or overwhelmed, but it is easy to determine the number of most similar features used in the whole data.

By combining all the available information from the listening records (U+S+US+SS+H), we obtained the best result 0.5021 in MAP@10 in Table IV, which is significantly better than the baseline 0.3816. A simple idea as it is, using the proposed ID similarity indicators greatly improve the accuracy of recommendation. Moreover, the ID similarity indicators are suitable for other recommendation problems because they are in the same problem structure: to predict whether an item would be accepted by a user.

TABLE V. PERFORMANCE OF FEATURE SIMILARITY

Features	MAP@10	Recall
U + S + BY	0.4301	0.5751
U + S + BY + BYS	0.4348	0.5830
U + S + A	0.5025	0.6538
U + S + A + AS	0.5125	0.6640
U + S + LR	0.4283	0.5723
U + S + LR + LRS	0.4382	0.5834
U + S + Au	0.4254	0.5809
U + S + Au + AuS	0.4576	0.6114

TABLE VI. PERFORMANCE OF FEATURE SIMILARITY

Features	MAP@10	Recall
U + S + M	0.4134	0.5539
U + S + M + MS	0.4202	0.5652
U + S + VAD	0.4483	0.5905
U + S + VAD + VADS	0.4511	0.5935
U + S + SRS	0.4213	0.5653

2) Content-based Recommendation: Four similarity features were extracted from the dataset:

- **Birth Year Similarity (BYS):** Two users are similar if they are born in the same year.
- **Live Region Similarity (LRS):** Two users are similar if they live in the same region geographically.
- **Artist Similarity (AS):** Two songs are similar if they are sung by the same artist.
- **Audio Similarity (AuS):** Two songs are similar if they are close in the audio feature space spanned by the 53 audio features considered in this work.

Note that BYS and LRS are personal information that is not always available for a recommendation problem. Similarly, AS and Aus are musical information that is only available if we have access to the metadata or the audio content of the songs.

Table V lists the improvement introduced by the use of feature similarity. The results show that four similarities perform well in recommendations. Among the four similarities, *Birth Year Similarity* cannot obtain a significant improvement in the experiments. This is possibly due to the incompleteness of the metadata, because only half of the users have birth year information in our dataset. Moreover, another interesting observation is that the audio features significantly enhance on the recommendation performance after the audio similarity is added. The result implies that the abstract information such as the audio feature is hard to be organized directly, but its similarity information provides insightful information.

3) Context-based Recommendations: Next, we evaluated context-based recommendation by using *Mood Tag* and *Emotional Words*. These two features reflect the user's mood when writing the article. We want to utilize the emotional information from user-generated articles and mood tags. The similarity information can be obtained in the same way:

- **Mood Similarity (MS):** Two user are similarly if they tend to express similar moods in their articles.

TABLE VII. PERFORMANCE ON COMPLETE FEATURE VECTOR.

Features	MAP@10	Recall	Note
U + S	0.3817	0.5216	Base-line
U + S + C*	0.5120	0.6614	
U + S + C* + S*	0.5236	0.6684	Hybrid
U + S + C* + S*	0.5251	0.6708	Hybrid + Grouping

Note: C* denotes all the categorical features, and S* denotes all the extracted similarity features.

TABLE VIII. PERFORMANCE OF DIFFERENT GROUPING SCHEME.

Features	MAP@10	Recall
U + US + S + SS	0.4712	0.6251
(U, US) + (S, SS)	0.4670	0.6206
(U + US) + (S + SS)	0.4570	0.5975
(U) + (US) + (S) + (SS)	0.4845	0.6334

Note: The first result is run on standard factorization machine.

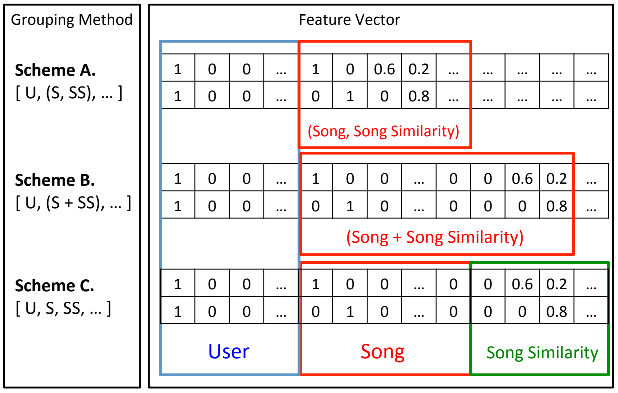


Fig. 4. An example for explaining different grouping method.

- **VAD Similarity (VADS):** Two users are similar if the affective qualities of the articles they wrote are similar.

Note that contextual information is also not always available for a recommendation problem. We only considered context information extracted from mood tags and articles in this work, but the proposed method is also applicable for other contextual information as well.

As the first and third rows of Table VI shows, the performance of adding the *Mood Tags* feature is 0.4134 in terms of MAP@10, which is lower than the contextual VAD feature computed from user-generated articles. This result indicates that the VAD feature provides more affective information of the user context. Although the mood similarity does not lead to remarkable improvement, the VAD similarity feature is still considered effective.

4) Hybrid Recommendation: Finally, we studied if we can further boost the accuracy by integrating all the proposed similarity features, including categorical ones (denoted as C* collectively) and similarity features (denoted as S* collectively). As Table VII shows, using more data generally leads to better accuracy. When all the features are considered (U+S+C*+S*), we are able to obtain 0.5251 in MAP@10 and 0.6708 in recall, both of which are the highest ones in our evaluation. This result confirms again the ability of the proposed method in incorporating multiple similarity information.

B. Grouping Approaches

Since there are too many similarity features that could be utilized in the FM model, the increasing interactions would lead to a high computation cost. Moreover, these interactions might also be non-informative or noisy. For instance, the interaction between *User ID* and *User Age* is non-informative; and the inner-connection among the similarity features will increase the too much computation cost. To avoid the situation, we applied grouping Factorization Machine as an extended model to offer a more flexible framework for combining the features. Since the self-group interactions are eliminated from the model, we have more choices to build different useful features. In the case of similarity feature, we employ following three index schemes:

A. Same Group and Same Index
B. Same Group and Different Index
C. Different Group and Different Index

Figure 4 illustrates the three grouping schemes. The scores of similar songs will be directly added in the same indexes for the first scheme, but will have additional index value in the second scheme. Scheme A can reduce the index dimension, but it is inappropriate for combing multiple features since the index may encounters the duplication problem. For the third scheme, the song similarities are indexed in another group. The main difference is that the interactions between the song and similar songs will be counted in learning process. In short, the feature vector can be organized with more different forms.

We evaluated the three schemes on user-item features with the grouping factorization machine. Table VIII shows the result. The first result is conducted by standard factorization machine and the remaining results are obtained from the grouping factorization machine. The interactions between object and similar objects are helpful according to the performance, but the noise may occurs when the interactions between similar features. In view of this, we adopt the grouping factorization machine as an extended method to further improve the quality of recommendations. The final result is shown in Table VII. Moreover, in order to demonstrate the strength of grouping factorization, we further examine performance on the standard factorization machine with different settings.

1) Training Loss: Figure 5 plots the training loss in the training step of the two approaches. The Y-axis corresponds to the root mean square error (RMSE), measuring the difference between predicted values and true values:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}. \tag{13}$$

The X-axis indicates the number of iterations. The grouping factorization machine is much faster to achieve the convergence. In addition, although it does not get a lower RMSE score, it still gets a better performance on testing data. In other words, the grouping action helps prevent the model from overfitting the training data.

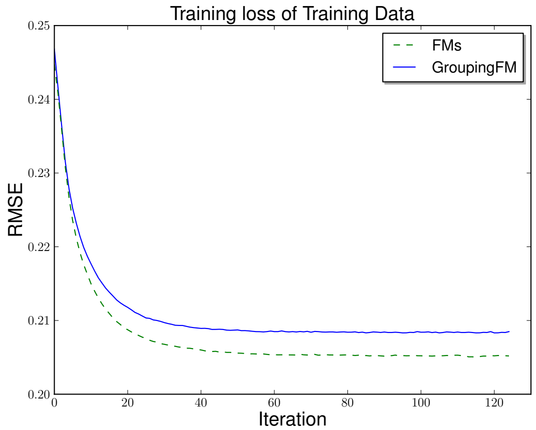


Fig. 5. Training loss between standard FM and grouping FM

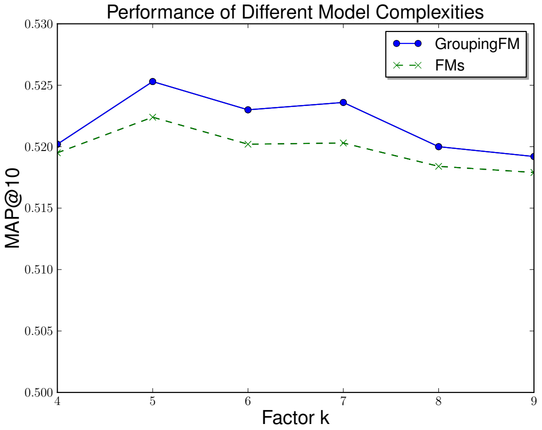


Fig. 6. Different factor κ of standard FM and grouping FM

2) Model Complexity: The parameter κ controls the model complexity of FM. We change it from 3 to 10, and plot the performance in Figure 6. The grouping factorization machine gets better performance for each factor κ. The results indicate that the grouping FM is useful to enhance the quality of recommendations.

VI. Conclusion

In this paper, we have presented a novel approach that incorporates multiple feature similarity to factorization model via feature engineering. The similarity computation captures the similar patterns from the objects and enhances the convergence speed and accuracy of FM. The proposed method is applicable to many kinds of features, which means we can obtain the higher level information from multiple aspects. Our experimental results show that feature similarity indeed benefits the recommendation performance. In addition, we also propose several features, including CF-based, content-based and context-based ones. Among these features, we try to capture the relationship between users and songs by matching users' emotions. The results show that the idea is able to enhance the quality of recommendations. Then, in order to avoid the noise within large similarity features, we adopt the grouping FM as the extended method to model the problem. The unnecessary connection can be eliminated if the features are within a same group. With the aforementioned technical contributions, we are able to improve the Mean Average Precision in music recommendation for a real-world, large-scale dataset from 0.3817 to 0.5251, comparing to the tradition CF-based baseline.

Acknowledgment

This work was partially supported by the National Science Council of Taiwan via the grants NSC 100-2218-E-004-001, 101-2221-E-004-017, 102-2221-E-001-004-MY3, and a research grant from KKBOX.

References

[1] Y.-H. Yang and J.-Y. Liu, "Quantitative study of music listening behavior in a social and affective context," *IEEE Trans. Multimedia*, vol. 15, no. 6, 2013.

[2] C.-M. Chen, M.-F. Tsai, J.-Y. Liu, and Y.-H. Yang, "Using emotional context from article for contextual music recommendation," in *Proc. ACM MM*, 2013.

[3] J. Davidson, B. Liebald *et al.*, "The youtube video recommendation system," in *Proc. ACM RecSys*, 2010, pp. 293–296.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan. 2003.

[5] M. Jiang, P. Cui *et al.*, "Social contextual recommendation," in *Proc. ACM CIKM*.

[6] Y. Shen and R. Jin, "Learning personal + social latent factor model for social recommendation," in *Proc. ACM SIGKDD*, 2012, pp. 1303–1311.

[7] K. Chen, T. Chen *et al.*, "Collaborative personalized tweet recommendation," in *Proc. ACM SIGIR*, 2012, pp. 661–670.

[8] T. Chen, L. Tang *et al.*, "Combining factorization model and additive forest for collaborative followee recommendation."

[9] N. Hariri, B. Mobasher, and R. Burke, "Context-aware music recommendation based on latenttopic sequential patterns," in *Proc. ACM RecSys*, 2012, pp. 131–138.

[10] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy," in *Proc. ACM RecSys*, 2011, pp. 165–172.

[11] R. Cai, C. Zhang *et al.*, "Musicsense: contextual music recommendation using emotional allocation modeling," in *Proc. ACM MM*, 2007, pp. 553–556.

[12] J. Weston, C. Wang, R. Weiss, and A. Berenzweig, "Latent collaborative retrieval," *CoRR*, vol. abs/1206.4603, 2012.

[13] I. Pilászy and D. Tikk, "Recommending new movies: even a few ratings are more valuable than metadata," in *Proc. ACM RecSys*, 2009, pp. 93–100.

[14] "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 57:1–57:22, May 2012.

[15] L. Hong, A. S. Doumith, and B. D. Davison, "Co-factorization machines: modeling user interests and predicting individual decisions in twitter," in *Proc. ACM WSDM*, 2013, pp. 557–566.

[16] S. Rendle, "Factorization machines," in *Proc. IEEE ICDM*, 2010.

[17] S. Rendle, Z. Gantner *et al.*, "Fast context-aware recommendations with factorization machines," in *Proc. ACM SIGIR*, 2011, pp. 635–644.

[18] S. Rendle, "Bayesian factorization machines."

[19] D. T. Derek Tingle, Youngmoo E. Kim, "Exploring automatic music annotation with acoustically-objective tags," 2010, pp. 55–61.

[20] M. Bradley and P. J. Lang, "Affective norms for english words ANEW: Instruction manual and affective ratings." The Center for Research in Psychophysiology, Univ. Florida, Tech. Rep., 1999.

[21] D. T. Derek Tingle, Youngmoo E. Kim, "Exploring automatic music annotation with acoustically-objective tags," 2010, pp. 55–61.