Mining Contextual Movie Similarity with Matrix Factorization for Context-Aware Recommendation

YUE SHI, MARTHA LARSON, and ALAN HANJALIC, Delft University of Technology

Context-aware recommendation seeks to improve recommendation performance by exploiting various information sources in addition to the conventional user-item matrix used by recommender systems. We propose a novel context-aware movie recommendation algorithm based on joint matrix factorization (JMF). We jointly factorize the user-item matrix containing general movie ratings and other contextual movie similarity matrices to integrate contextual information into the recommendation process. The algorithm was developed within the scope of the mood-aware recommendation task that was offered by the Moviepilot mood track of the 2010 context-aware movie recommendation (CAMRa) challenge. Although the algorithm could generalize to other types of contextual information, in this work, we focus on two: movie mood tags and movie plot keywords. Since the objective in this challenge track is to recommend movies for a user given a specified mood, we devise a novel mood-specific movie similarity measure for this purpose. We enhance the recommendation based on this measure by also deploying the second movie similarity measure proposed in this article that takes into account the movie plot keywords. We validate the effectiveness of the proposed JMF algorithm with respect to the recommendation performance by carrying out experiments on the Moviepilot challenge dataset. We demonstrate that exploiting contextual information in JMF leads to significant improvement over several state-of-the-art approaches that generate movie recommendations without using contextual information. We also demonstrate that our proposed mood-specific movie similarity is better suited for the task than the conventional mood-based movie similarity measures. Finally, we show that the enhancement provided by the movie similarity capturing the plot keywords is particularly helpful in improving the recommendation to those users who are significantly more active in rating the movies than other users.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: Collaborative filtering, context-aware recommendation, matrix factorization, mood-specific movie similarity, recommender systems

ACM Reference Format:

Shi, Y., Larson, M., and Hanjalic, A. 2013. Mining contextual movie similarity with matrix factorization for context-aware recommendation. ACM Trans. Intell. Syst. Technol. 4, 1, Article 16 (January 2013), 19 pages. DOI = 10.1145/2414425.2414441 http://doi.acm.org/10.1145/2414425.2414441

1. INTRODUCTION

Recently, context-aware recommendation has experienced an upsurge of interest in the recommender systems community [Said et al. 2010]. The interest has been spurred

Portions of this work are based on Shi et al. [2012b].

The research leading to these results was carried out within the PetaMedia Network of Excellence and has received funding from the European Commission's 7th Framework Program under grant agreement no. 216444.

Author's addresses: Y. Shi, M. Larson, and A. Hanjalic, Multimedia Information Retrieval Lab, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628CD Delft, Netherlands; email: {y.shi,m.a.larson,a.hanjalic}@tudelft.nl.

DOI 10.1145/2414425.2414441 http://doi.acm.org/10.1145/2414425.2414441

by a growing awareness of the potential of contextual information, if available, to improve the quality of recommendations [Adomavicius et al. 2005; Burke 2002]. Such information can include, for instance, relationships among users in social media sites, tags of products, introduction about products or timestamps of user actions. One of the most promising potential contributions of contextual information is its ability to alleviate the problem of data sparseness in the original user-item matrix. Contextual information can be exploited to more reliably estimate relationships between items compensating for cases in which the information in the original user-item matrix is insufficient.

In addition to relying on information sources beyond the conventional user-item matrix, context-aware recommendation also differs from traditional recommendation in the sense that its purpose is more specific, for instance, movies are recommended specifically for the week of Christmas [Gantner et al. 2010; Liu et al. 2010], or for a specific mood that they should elicit in the user [Shi et al. 2010b]. Accordingly, the research challenge in the area of context-aware recommendation involves two aspects. It is expected that the new recommendation technique/model retains the benefits of conventional recommender approaches, such as collaborative filtering (CF) [Adomavicius and Tuzhilin 2005], which infers the recommendation from a user-item matrix, while also allowing contextual information to steer the recommendation process towards results suitable for a given use case (purpose).

In this article, we address the recommendation task formulated in the Moviepilot mood track of the context-aware movie recommendation (CAMRa) challenge [Said et al. 2010], henceforth referred to as the Moviepilot challenge, in which the task is to recommend movies to a user given a specific mood. For this purpose, not only the user-item matrix is provided, capturing general preferences of the users for different movies, but also the contextual information consisting of various movie metadata, such as mood tags, plot keywords, movie locations and intended audience.

In order to maximize the benefit of the given contextual information and optimally address the two aspects of context-aware recommendation defined above, we propose a novel recommendation model, based on a joint matrix factorization (JMF), that factorizes the user-item (user-movie) matrix, while also exploiting the contextual information as additional regularization terms. Specifically, for generating context-based links between movies, we propose a set of contextual movie similarities, each of which steers the recommendation process and contributes to JMF in a specific fashion. From the contextual information available within the setting of our task, we deploy mood tags and plot keywords (PK). The potential of plot keywords to improve mood-based recommendation lies in the assumption that if mood-based movie similarity is difficult to infer reliably from the mood tags, then movies with similar moods might still be linked together if they have similar plots. Although we focus only on two types of contextual information, the proposed JMF model could be easily expanded to integrate other contextual information.

The novel contribution of this article can be summarized as follows. We propose a novel context-aware movie recommendation algorithm that extends the basic matrix factorization (MF) model to take into account context-induced links between movies. Furthermore, we propose a set of contextual movie similarities that evaluate the relationships between movies in view of specific contextual information, that is, mood-specific movie similarity and PK-based movie similarity, and integrate these similarities in our recommendation model. Finally, we apply these in the setting of the Moviepilot challenge for evaluation, demonstrating that the proposed algorithm outperforms a wide range of state-of-the-art approaches for context-aware recommendation.

The remainder of the article is structured as follows. In the next section, we present an overview of the Moviepilot challenge. Then, in Section 3, we summarize related work and position our approach with respect to it. The proposed contextual movie similarities and JMF model are described in detail in Section 4, after which, in Section 5, we present experimental evaluation on the Moviepilot challenge dataset. The last section sums up the key aspects of the proposed algorithm and briefly addresses the direction for future work.

2. OVERVIEW OF THE MOVIEPILOT CHALLENGE

2.1. Problem Statement

The task of the Moviepilot challenge can be defined as follows: Based on both the user-movie rating matrix and other provided contextual information, recommend a list of movies that have a specific mood property to each target/test user [Said et al. 2010]. In other words, the task is to design a model that takes the user-movie rating matrix, the contextual information and a specific mood as input, and outputs a list of movies with the specific mood for each target user. The recommendation list should contain as many relevant movies as possible, which are also ranked as high as possible. Note that within this challenge a movie is considered to be "relevant" to a user if it has been rated by that user and if it is characterized by the prespecified mood. In addition to user-movie ratings, the dataset provided for this challenge contains various types of contextual information, for instance, movie-emotion (mood) assignments, movie-PK assignments, and the release date of each movie. Note that all the data are provided in the form of identifiers and the real identities of the underlying entities are not made known to the public due to privacy considerations. For example, we do not have access to direct knowledge concerning the identity of the movie with identifier 10, or the mood with identifier 5.

2.2. Characteristics of the Challenge

In this section we discuss the characteristics of this challenge that make it distinctive from the traditional CF problem.

First, the evaluation of the recommendation performance is not based on the rating prediction error rate, for instance, as done in the Netflix contest[1], but on the recommendation list, the quality of which is assessed using metrics for the evaluation of ranked results lists, for instance, precision at N and mean average precision. As recently suggested in Liu et al. [2009] and Shi et al. [2010b], these evaluation criteria are more sensible, since the ultimate goal of a recommender system is to generate a list of recommended items for a user, rather than only provide the predictions of relevance scores for different items.

Second, the evaluation focuses on the recommended movies characterized by the prespecified mood, which requires a different approach to designing recommendation mechanisms, compared to the traditional, mainly CF-based practice. In view of such a focus, one could namely first apply a known recommendation approach based on an analysis of the user-item matrix to generate the initial recommendation and then remove those movies not having the desired mood. This approach is, however, likely lead to a recommendation performance being far short of what is targeted, as is demonstrated later during our experiments. A priori, a basis for this expectation could also be drawn from the distribution of ratings in the dataset, shown in Figure 1. The distribution reveals that many movies in the provided validation set (i.e., a set provided for tuning the parameters) have low ratings. Note that low-rated movies are significant in the Moviepilot challenge, since if the number of movies of the target mood is limited, these movies come into consideration as the most appropriate ones to

[1] www.netflixprize.com.

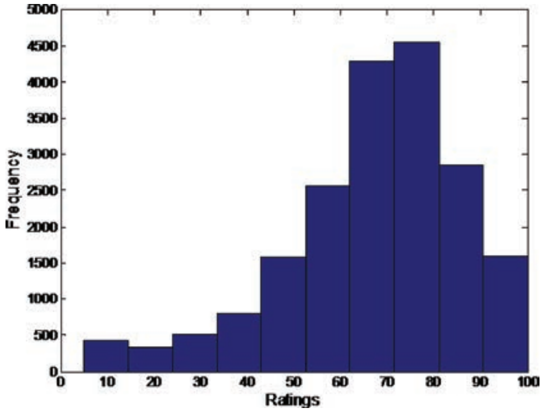


Fig. 1. The distribution of ratings in the validation set of Moviepilot challenge.

recommend to the user. As a consequence of this discrepancy between the rating predictions generated by the user-item matrix and the contextual information, the number of movies resulting from an initial traditional recommendation step and having the proper mood characteristics may be too small. Instead, one should conceptualize the recommendation process as not only involving the movies that the user would generally be interested in, but also simultaneously emphasizing the movies with the specified mood. In other words, context awareness of the recommendation is not likely to emerge from a simple filtering step of the initial recommended item list, but rather through letting the contextual information actively steer the recommendation process.

Finally, other alternatives for approaching collaborative recommendation, such as the ranking-oriented approaches [Liu et al. 2009; Rendle et al. 2009a; Shi et al. 2010a], are also expected not to perform well within this challenge, since both a pair-wise ranking approach and a list-wise ranking approach require training examples annotated with explicit or implicit item ratings. However, the value of ratings would not influence the evaluation according to the challenge setting.

3. RELATED WORK

This section briefly summarizes the existing related research in CF, context-aware recommendation and tag-aware recommendation, in order to position the recommendation approach we propose in this article.

3.1. Collaborative Filtering

CF approaches are usually categorized as memory-based or model-based [Adomavicius and Tuzhilin 2005; Breese et al. 1998; Herlocker et al. 2004]. Memory-based approaches can be further categorized as user-based or item-based, depending on whether the recommendation for a user is aggregated from users with similar preference, for instance, the work of [Herlocker et al. 1999; Shi et al. 2009; Zhang and Pu 2007], or from items that are similar to those he already liked, for instance, the work of [Deshpande and Karypis 2004; Linden et al. 2003; Sarwar et al. 2001]. The key drawback of memory-based CF approaches lies in the expensive computation for similarities among all users or items, which does not scale with the typically large numbers of users and items in real-world recommender systems. Compared to memory-based approaches, model-based approaches first employ statistical and machine learning techniques to learn a prediction model from a training set of user-item matrix data and then apply that model to generate recommendations, such as Gaussian mixture model [Kleinberg and Sandler 2008] and latent semantic model [Hofmann 2004]. Among different model-based approaches, matrix factorization (MF) techniques have attracted much research attention, due to the advantages of scalability and accuracy [Koren et al. 2009; Salakhutdinov and Mnih 2008], especially for large-scale data, as exemplified by the Netflix contest. Generally, MF techniques learn latent features of users and items from the observed ratings in the user-item rating matrix. These latent features are further used to predict unobserved ratings. Rather than targeting the rating prediction problem, recent research started to exploit possibilities for ranking-oriented CF approaches that focus on the quality of recommendation lists, for instance, EigenRank [Liu and Yang 2008], CoFiRank [Weimer et al. 2007], probabilistic latent preference analysis [Liu et al. 2009], Bayesian personalized ranking [Rendle et al. 2009a], and ListRank [Shi et al. 2010a].

The joint matrix factorization (JMF) is an extension of MF. The designation "joint" makes reference to the simultaneous factorization of more than one matrix. JMF is formulated similarly to relational learning, as defined by [Singh and Gordon 2008], which also factorizes multiple matrices from related domains. It has been widely applied e.g., for fusing document content and graph link information for document retrieval or web page classification [Deng et al. 2009; Zhu et al. 2007], or for fusing geographical location features and people activity correlation for location-based recommendation [Zhu et al. 2010]. In this article, we exploit JMF to fuse contextual movie similarities, that is, the mood-specific movie similarity and the plot-keyword-based movie similarity, with the user-movie rating matrix. The difference between our work and the aforementioned previous work on JMF is twofold. First, compared to the work of [Deng et al. 2009; Singh and Gordon 2008; Zhu et al. 2007], we exploit the available contextual information in the form of contextual movie similarities rather than original movie contextual information. By this means, we maintain the advantage of using contextual information for learning latent movie features, namely alleviating data sparseness in the user-movie rating matrix, while at the same time eliminating unnecessary learning for additional latent features representing other entities, for instance, movie mood and plot keywords. Second, above and beyond this body of existing work [Deng et al. 2009; Singh and Gordon 2008; Zhu et al. 2010; Zhu et al. 2007], we propose a new mood-specific movie similarity that explicitly addresses the recommendation bias of the Moviepilot challenge and propose a method to integrate this similarity into the JMF framework. We note that our work is consistent with the contemporary trend of new work in the area, falling into the category of approaches extending matrix factorization [Gantner et al. 2010; Liu et al. 2010].

3.2. Context-Aware Recommendation

Some of the earliest work on context-aware recommendation was done by Adomavicius et al. [2005], who deployed contextual information, for instance, time and place, to generate additional dimensions in the user and item rating vectors. Anand and Mobasher [2007] exploited users' preference information from both long-term memory and short-term memory in the recommendation process. Baltrunas and Ricci [2009] proposed to take into account the context, for instance, user age and gender, to split item ratings as a preprocessing step for CF approaches, in order to improve CF accuracy. Application of context-aware recommendation approaches in specific use cases has been explored in work exploiting contextual information for travel recommendation [Carolis et al. 2009], news recommendation [Cantador and Castells 2009], and music recommendation [Su et al. 2010]. Compared to this previous work, our proposed approach has the advantages of being adaptive to any application domain and being able to handle large-scale datasets.

Specifically concerning the context-aware movie recommendation tasks in the Moviepilot challenge, in addition to the mood-specific recommendation task other tasks have been addressed as well, such as recommending movies for a specific week (such as a holiday week) and recommending movies by exploiting social relationships [Said et al. 2010]. For week-based movie recommendation, timestamps of users ratings have been made available as the contextual information. Approaching this task, [Gantner et al. 2010] extended a pair-wise interaction tensor factorization model [Rendle and Schmidt-Thieme 2010], which was originally designed for tag recommendation, to factorize the {user, time, movie} ternary data for movie recommendation in a given week. [Liu et al. 2010] investigated both tensor factorization and sequential matrix factorization to integrate time-dependent characteristics of users and items into the recommendation process. In addition to the work presented in the Moviepilot challenge, [Koren 2009] has proposed to include temporal dynamics into neighbor-based CF and matrix factorization for improved performance in the Netflix contest.

On the other hand, regarding the task targeting the integration of social relationships into recommendation, Liu et al. [2010] investigated both collective matrix factorization (equivalent to the work of Ma et al. [2008]) and network-regularized matrix factorization (equivalent to the work of Ma et al. [2009]) for this purpose. However, they found that including social relationships into MF leads to only a small improvement compared to basic matrix factorization.

3.3. Tag-Aware Recommendation

We also point out that our work on context-aware recommendation is related to tag-aware recommendation. [Tso-Sutter et al. 2008] proposed a fusion method to incorporate tags into traditional user-based CF and item-based CF for item rating prediction. Wetzker et al. [2009] proposed exploiting probabilistic latent semantic analysis [Hofmann 2004] to unify user-item relations and item-tag relations into one model, resulting in improved item recommendation performance. A similar principle was presented by Wang et al. [2006], but from a more fundamental perspective.

More recently, another group of state-of-the-art approaches has emerged that makes use of tensor factorization techniques, for instance, the work of Kolda and Sun [2008]. Under such approaches, latent features are learned from the {user, tag, item} triplet/ternary data directly for item recommendation [Karatzoglou et al. 2010], tag recommendation [Rendle et al. 2009b; Rendle and Schmidt-Thieme 2010; Symeonidis et al. 2008], or both [Symeonidis et al. 2010]. However, tensor factorization is known to be computationally expensive, that is, usually being cubic in the number of latent dimensions. In addition, in the Moviepilot challenge, where the mood tags and plot keywords of movies are not associated with users, there is no ternary data actually available for either exploiting or comparing with tensor factorization techniques.

Another way of benefiting from the relations among users, tags and items for the purpose of recommendation is to deploy a graph-based approach. For instance, Konstas et al. [2009] proposed a recommendation framework that infers the item preferences of the users from a hyper-graph including different types of nodes and links, which captures user-user, tag-user, tag-item and user-item relations. Preferences are inferred using a random-walk-with-restarts method. While the method was proved effective conceptually, it requires the availability of rich contextual information in order to result in significant recommendation benefits in a practical use case. Graph-based approaches are not suitable for our task, which mixes binary information (mood-movie relation) with scale information (user ratings). In Konstas et al. [2009], the links in the hyper-graph are assumed to be binary, which represents a radical simplification of the relationships between the nodes. For integrating user-item ratings and other contextual information into a hyper-graph, it would also be necessary not only to

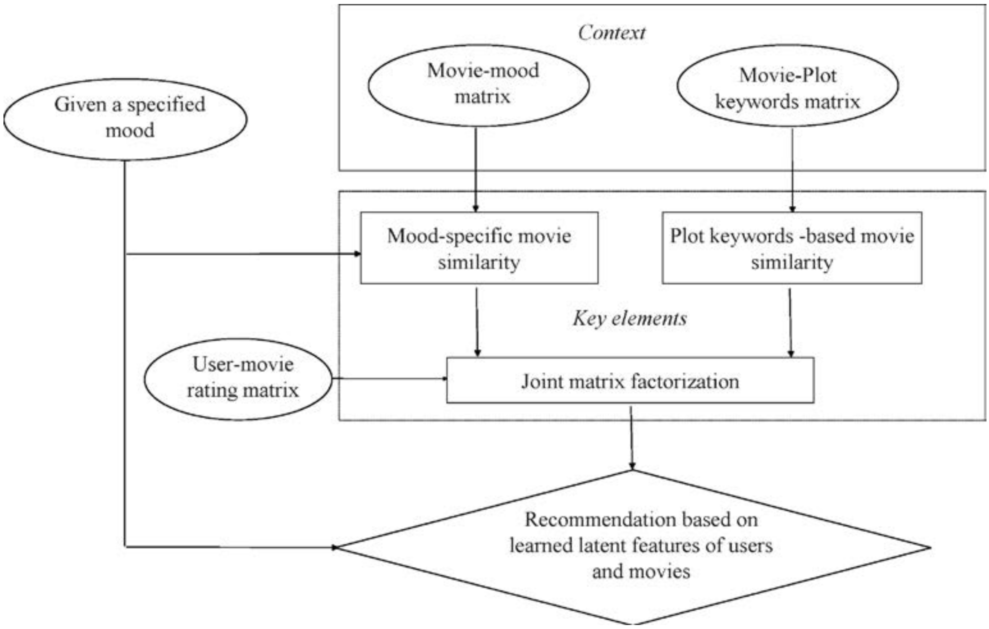

Context
Movie-mood matrix
Movie-Plot keywords matrix
Given a specified mood
Mood-specific movie similarity
Plot keywords -based movie similarity
Key elements
User-movie rating matrix
Joint matrix factorization
Recommendation based on learned latent features of users and movies

Fig. 2. The flow chart of the proposed algorithm.

impose this simplification, but also to impose it in a way that retains the balance between the mix of different information types without information loss, which could be introduced during the simplification step. For this reason, graph-based approaches are not an obvious choice for application in our work.

4. THE PROPOSED ALGORITHM

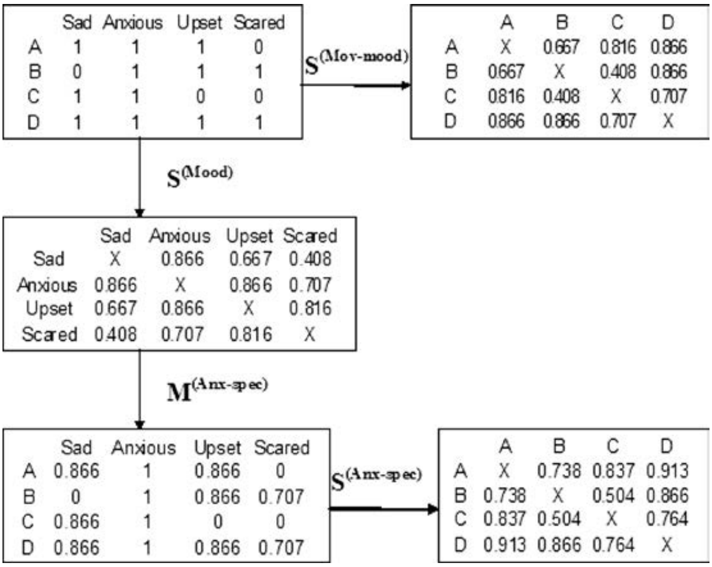
In this section, we introduce our proposed algorithm for the context-aware movie recommendation task of the Moviepilot challenge. The flow chart of the proposed algorithm is given in Figure 2. While the use case and rationale behind our general approach have been discussed in Section 1, we focus in this section on the analysis of three key components of the algorithm, namely the mood-specific movie similarity and the similarity based on plot keywords, and the joint matrix factorization model. Then, we also perform a complexity analysis of the proposed algorithm.

4.1. Mood-Specific Movie Similarity

According to the traditional item-based CF, item-to-item similarity can be computed as the cosine similarity between two item rating vectors [Deshpande and Karypis 2004]. Similarly, given the movie-mood (binary) matrix $\mathbf{M}$ (consisting of N movies and K_1 mood tags), we can compute mood-based similarity between movie j and movie n as:

$$S_{jn}^{(Mov-mood)} = \frac{\sum_{k=1}^{K_1} M_{jk} M_{nk}}{\sqrt{\sum_{k=1}^{K_1} M_{jk}^2}\sqrt{\sum_{k=1}^{K_1} M_{nk}^2}}. \tag{1}$$

Here, $M_{jk} = 1$ indicates that the movie j has the mood tag k, otherwise $M_{jk} = 0$. The mood-based similarity in Equation (1), however, only indicates general closeness of two movies in terms of all their mood properties. For example as shown in Figure 3, two movies ($\mathbf{A}$ and $\mathbf{B}$) sharing different mood properties could be equally similar to


$\mathbf{S}^{(\mathbf{Mov\text{-}mood})}$
$\mathbf{S}^{(\mathbf{Mood})}$
$\mathbf{M}^{(\mathbf{Anx\text{-}spec})}$
$\mathbf{S}^{(\mathbf{Anx\text{-}spec})}$

	Sad	Anxious	Upset	Scared
A	1	1	1	0
B	0	1	1	1
C	1	1	0	0
D	1	1	1	1

	A	B	C	D
A	X	0.667	0.816	0.866
B	0.667	X	0.408	0.866
C	0.816	0.408	X	0.707
D	0.866	0.866	0.707	X

	Sad	Anxious	Upset	Scared
Sad	X	0.866	0.667	0.408
Anxious	0.866	X	0.866	0.707
Upset	0.667	0.866	X	0.816
Scared	0.408	0.707	0.816	X

	Sad	Anxious	Upset	Scared
A	0.866	1	0.866	0
B	0	1	0.866	0.707
C	0.866	1	0	0
D	0.866	1	0.866	0.707

	A	B	C	D
A	X	0.738	0.837	0.913
B	0.738	X	0.504	0.866
C	0.837	0.504	X	0.764
D	0.913	0.866	0.764	X

Fig. 3. An illustrative example of mood-specific movie similarity.

another movie (**D**). If the required mood of a movie is specified, this similarity fails to differentiate between movies **A** and **B**.

In view of the above, we expect that an accurate approach would involve adjusting the movie-to-movie similarity in a way to make it more biased towards a particular mood in case movies with that mood need to be recommended. We therefore propose a novel mood-specific similarity measure to address this issue. Although we focus on the Moviepilot challenge that targets mood-specific movie recommendation, the concept of mood-specific similarity can easily be generalized for recommendation with different contextual constraints.

Instead of evaluating the consistency of mood tags between two movies, as in Equation (1), we can also compute the normalized cooccurrence of the mood i and mood k in the movie collection as:

$$S_{ik}^{(Mood)} = \frac{\sum_{j=1}^{N} M_{ji} M_{jk}}{\sqrt{\sum_{j=1}^{N} M_{ji}^2}\sqrt{\sum_{j=1}^{N} M_{jk}^2}}. \tag{2}$$

Once the mood cooccurrence matrix $\mathbf{S}^{(\mathbf{Mood})}$ is obtained, we can generate a mood-specific movie-mood matrix that is biased towards a given mood m, as expressed in Equation (3):

$$M_{jk}^{(m-spec)} = \begin{cases} S_{jk}^{(Mood)}, k \neq m \\ M_{jk}, k = m. \end{cases} \tag{3}$$

While preserving the original movie-mood matrix values in the column corresponding to the mood m, the values in the matrix as in Equation (3) for any other mood k are replaced by the values of the similarities in Equation (2), implicitly indicating to which extent mood k is informative about mood m. Note that the mood-specific movie-mood matrix $\mathbf{M}^{(\mathbf{m-spec})}$ is not binary. Then, we define the mood-specific movie similarity biased towards a given mood m as:

$$S_{jn}^{(m-spec)} = \frac{\sum_{k=1}^{K_1} M_{jk}^{(m-spec)} M_{nk}^{(m-spec)}}{\sqrt{\sum_{k=1}^{K_1} M_{jk}^{(m-spec)2}} \sqrt{\sum_{k=1}^{K_1} M_{nk}^{(m-spec)2}}}. \tag{4}$$

In order to illustrate the effect of this similarity, we again focus on the example in Figure 3 and assume that there is specific demand for movies corresponding to the mood "anxious." In the search for all movies satisfying this recommendation criterion, we can derive the mood-specific movie similarity for the mood "anxious." In contrast with the case where general mood-based similarity in Equation (1) is computed, the mood-specific movie similarity now indicates movie **D** to be more similar to movie **A** than to movie B. This is because movie **A** has mood tags ("sad" and "upset") that are more informative about "anxious" than the mood tags of the movie **B**. In this way, the movie-specific movie similarity steers the movie comparison towards the target mood and helps the context-aware recommendation.

4.2. Plot Keyword-Based Movie Similarity

Similar to the mood-based movie similarity, we also define the similarity between movies in terms of movie plot keywords (PKs). Since PKs represent the movie content, this similarity can improve the mood-based links between movies. Since both PKs and moods potentially reflect the underlying movie content, it is reasonable to expect that movies having similar plots could evoke similar emotions in users. We first create a binary movie-PK matrix **P** consisting of N movies and K_2 PKs, where $P_{jk} = 1$ if the movie j has the PK k, and $P_{jk} = 0$ otherwise. Then, the PK-based similarity between movie j and movie n can be defined as:

$$S_{jn}^{(Mov-PK)} = \frac{\sum_{k=1}^{K_2} P_{jk} P_{nk}}{\sqrt{\sum_{k=1}^{K_2} P_{jk}^2} \sqrt{\sum_{k=1}^{K_2} P_{nk}^2}}. \tag{5}$$

4.3. Joint Matrix Factorization

The basic MF [Koren et al. 2009] can be formulated as in Equation (6):

$$U, V = \arg\min_{U,V} \left\{ \frac{1}{2} \sum_{u=1}^{K} \sum_{j=1}^{N} I_{uj}^{R} \left(R_{uj} - U_u^T V_j \right)^2 + \frac{\lambda_U}{2} \|\mathbf{U}\|_F^2 + \frac{\lambda_V}{2} \|\mathbf{V}\|_F^2 \right\}. \tag{6}$$

Given the user-item rating matrix **R** consisting of K users and N items, the MF represents the user-item rating matrix **R** by two low-rank matrices, **U** and **V**. A d-dimensional set of latent features is deployed to represent both users (in **U**) and items (in **V**). Note that we use U_u to denote a column d-dimensional feature vector of user u, V_j is a column d-dimensional feature vector of movie j, and R_{uj} denotes the user u's rating on movie j. I_{uj}^{R} denotes an indicator function that is equal to 1 when $R_{uj} > 0$, and 0 otherwise. $\|\mathbf{U}\|_F$ and $\|\mathbf{V}\|_F$ are the Frobenius norms of **U** and **V**, that contribute to alleviating overfitting. λ_U, λ_V are regularization parameters for which we set $\lambda_U = \lambda_V = \lambda$ to simplify the model in this article.

In view of the discussion in the previous section, we now require that the movies being similar to each other with respect to the mood-specific similarity criterion in Equation (4) share similar latent movie features. For this purpose, we formulate a context-aware loss function $L_1(\mathbf{V})$ as shown in Equation (7).

$$L_1(\mathbf{V}) = \frac{1}{2}\sum_{j=1}^{N}\sum_{n=1}^{N} I_{jn}^{MS}\left(S_{jn}^{(m-spec)} - V_j^T V_n\right)^2, \tag{7}$$

where I_{jn}^{MS} denotes an indicator function that is equal to 1 when $S_{jn}^{(m-spec)} > 0$, and 0 otherwise.

Furthermore, we also assume that the movies similar to each other with respect to the PK-based similarity as in Equation (5) should also share similar latent movie features, implying that the similarity of the plots is informative for mood-specific movie recommendation. Therefore, we formulate another context-aware loss function $L_2(\mathbf{V})$ as shown in Equation (8).

$$L_2(\mathbf{V}) = \frac{1}{2}\sum_{j=1}^{N}\sum_{n=1}^{N} I_{jn}^{PK}\left(S_{jn}^{(Mov-PK)} - V_j^T V_n\right)^2. \tag{8}$$

Here, I_{jn}^{PK} denotes an indicator function that is equal to 1 when $S_{jn}^{(Mov-PK)} > 0$, and 0 otherwise.

Taking into account the context-aware loss functions as regularization terms in the basic MF model, a joint matrix factorization (JMF) model can be formulated as:

$$\begin{aligned} L(\mathbf{U},\mathbf{V}) =& \frac{1}{2}\sum_{u=1}^{K}\sum_{j=1}^{N} I_{uj}^{R}\left(R_{uj} - U_u^T V_j\right)^2 \\ &+ \frac{\alpha}{2}\sum_{j=1}^{N}\sum_{n=1}^{N} I_{jn}^{MS}\left(S_{jn}^{(m-spec)} - V_j^T V_n\right)^2 \\ &+ \frac{\beta}{2}\sum_{j=1}^{N}\sum_{n=1}^{N} I_{jn}^{PK}\left(S_{jn}^{(Mov-PK)} - V_j^T V_n\right)^2 + \frac{\lambda}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2). \end{aligned} \tag{9}$$

In this model, α and β are the trade-off parameters for weighting the contributions of regularization by the mood-specific movie similarity and by the PK-based movie similarity, respectively. Note that when both $\alpha = 0$ and $\beta = 0$, the JMF model converges to the basic MF model. For notation convenience, we use JMF-MS-PK to indicate the case that both $\alpha > 0$ and $\beta > 0$ and JMF-MS to indicate the model with $\alpha > 0$ and $\beta = 0$. Furthermore, we use JMF-MB to indicate the model with $\alpha > 0$ and $\beta = 0$ that exploits general mood-based movie similarity instead of the mood-specific movie similarity. A more elaborate investigation of the variations of the proposed JMF model is given in Section 5.

In summary, the advantage of the JMF model is two-fold: First, the additional contextual information, that is, mood-specific movie similarity and PK-based movie similarity, alleviate the usual deficiencies of the rating matrix $\mathbf{R}$, that is, data sparseness, since latent features of movies could be learned from contextual movie similarity matrices as well as from the rating matrix. Second, the mood-specific movie similarity could contribute to biasing recommendations towards movies with the specific mood.

Minimization of the objective function in Equation (9) can be solved by gradient descend with alternatively fixed $\mathbf{U}$ and $\mathbf{V}$. This process results in a local minimum

ALGORITHM 1: JMF-MS-PK

```
Input: user-movie rating matrix $\mathbf{R}$, mood-specific movie-to-movie similarity $\mathbf{S^{(m-spec)}}$, PK-based
        movie-to-movie similarity $\mathbf{S^{(Mov-PK)}}$, tradeoff parameters $\alpha$, $\beta$, regularization parameter
        $\lambda$, stop condition $\epsilon$.
Output: Complete user-movie relevance matrix $\hat{R}$.
Initialize $\mathbf{U^{(0)}}$, $\mathbf{V^{(0)}}$ with random values;
$t = 0$;
$f = 0$;
Compute $L^{(t)}$ as in Equation (9);
repeat
    $\eta = 1$;
    Compute $\frac{\partial L}{\partial U^{(t)}}$, $\frac{\partial L}{\partial V^{(t)}}$ as in Equation (10) and (11);
    repeat
        $\eta = \eta/2$; // maximize learning step size
    until $L(U^{(t)} - \eta\frac{\partial L}{\partial U^{(t)}}, V^{(t)} - \eta\frac{\partial L}{\partial V^{(t)}}) < L^{(t)}$;
    $U^{(t+1)} = U^{(t)} - \eta\frac{\partial L}{\partial U^{(t)}}, V^{(t+1)} = V^{(t)} - \eta\frac{\partial L}{\partial V^{(t)}}$;
    Compute $L^{(t+1)}$ as in Equation (9);
    if $1 - L^{(t+1)}/L^{(t)} \leq \epsilon$ then
        $f = 1$; // indicator of convergence
    end
    $t = t + 1$;
until $f = 1$;
$\hat{R} = U^{(t)T}V^{(t)}$;
```

solution. The gradients of $L(\mathbf{U}, \mathbf{V})$ with respect to $\mathbf{U}$ and $\mathbf{V}$ can be computed as:

$$\frac{\partial L}{\partial U_u} = \sum_{j=1}^{N} I_{uj}^{R}(U_u^T V_j - R_{uj})V_j + \lambda U_u \tag{10}$$

$$\begin{aligned}\frac{\partial L}{\partial V_j} =& \sum_{u=1}^{K} I_{uj}^{R}(U_u^T V_j - R_{uj})U_u + 2\alpha \sum_{n=1}^{N} I_{jn}^{MS}(V_j^T V_n - S_{jn}^{(m-spec)})V_n \\ &+ 2\beta \sum_{n=1}^{N} I_{jn}^{PK}(V_j^T V_n - S_{jn}^{(Mov-PK)})V_n + \lambda V_j.\end{aligned} \tag{11}$$

Note that in Equation (11) we exploit the symmetry of $\mathbf{S^{(m-spec)}}$ and $\mathbf{S^{(Mov-PK)}}$. The JMF-MS-PK algorithm is described in detail in Algorithm 1.

4.4. Complexity Analysis

The complexity of computing the contextual movie similarity matrices is normally quadratic to the number of movies, thta is, $O(N^2)$. In the case that a new movie appears, the complexity of updating each contextual movie similarity matrix is linear in the number of current movies, that is, $O(N)$. However, this computation could be carried out completely offline, since it is independent of learning latent features in JMF. By exploiting the data sparseness, the computation of the objective function in Equation (9) is of complexity $O(d|\mathbf{R}| + d|\mathbf{S^{(m-spec)}}| + d|\mathbf{S^{(Mov-PK)}}| + d(K + N))$, where $|\mathbf{R}|$ denotes the number of observed ratings in a given user-movie rating matrix, $|\mathbf{S^{(m-spec)}}|$ the number of non-zero similarities in the mood-specific similarity matrix, and $|\mathbf{S^{(Mov-PK)}}|$ the number of non-zero similarities in the PK-based similarity matrix. The complexity of the gradients in Equations (10) and (11) is $O(d|\mathbf{R}| + dK)$ and $O(d|\mathbf{R}| + d|\mathbf{S}^{(\mathbf{m-spec})}| + d|\mathbf{S}^{(\mathbf{Mov-PK})}| + dN)$, respectively. Considering the fact that we often have $|\mathbf{R}| >> K, N$, that is, the number of observed ratings is much larger than both the number of users and the number of movies in a collection, the total complexity in one iteration is $O(d|\mathbf{R}| + d|\mathbf{S}^{(\mathbf{m-spec})}| + d|\mathbf{S}^{(\mathbf{Mov-PK})}|)$. In practice, there are many more users than movies in movie recommender systems, for instance, the Netflix dataset involves around 480K users and around 18K movies (www.netflixprize.com), and the MovieLens dataset involves around 72K users and around 11K movies (www.grouplens.org/node/73). Therefore, the number of contextual links between movies could be much lower than user-movie links (ratings), leading to $|\mathbf{S}^{(\mathbf{m-spec})}|$, $|\mathbf{S}^{(\mathbf{Mov-PK})}| << |\mathbf{R}|$. The total complexity of the proposed algorithm could approximate to $O(d|\mathbf{R}|)$, which is linear with the number of observed ratings in the user-movie matrix. This analysis indicates that the proposed algorithm is computationally efficient and can be applied to large-scale cases.

5. EXPERIMENTAL EVALUATION

In this section, we present the experiments we conducted to evaluate the proposed algorithm. The research questions that need to be answered through the experiments can be formulated as follows.

(1) Does minimizing the objective function in Equation (9) contribute to improving recommendation performance?
(2) Can the proposed algorithm JMF-MS-PK outperform other state-of-the-art approaches?
(3) How does the mood-specific movie similarity contribute to the performance of mood-specific recommendation in the Moviepilot challenge?
(4) What is the contribution of integrating the PK-based movie similarity in addition to mood-specific movie similarity to the recommendation performance?

5.1. Experimental Setup

5.1.1. Dataset. Our experiments are conducted on the dataset of the “Moviepilot mood track,” which consists of around 4.5M ratings (scale 0-100) assigned by around 105K users to a collection of around 25K movies. The data sparseness of the user-movie rating matrix is around 99.83%. Apart from the user-movie rating matrix, various contextual information is provided, for instance, gender and age of users, production year of movies, intended audience of movies, etc. The detail of statistics of the dataset is presented at Said et al. [2010]. As mentioned in the introduction, we only exploit the mood tags of movies and the plot keywords of movies in this work. The movie-mood tag (binary) matrix consists of around 25K movies and 16 mood tags, which in total involves 6712 mood tag assignments on movies. The movie-PK (binary) matrix consists of around 25K movies and 5683 PKs, which in total involves 92124 PK assignments to movies.

5.1.2. Evaluation Metrics. We use the precision of top-N recommendation list (P@N) and the Mean Average Precision (MAP) as the evaluation metrics for measuring the quality of the recommendation list [Gunawardana and Shani 2009; Herlocker et al. 2004]. The P@N reflects the average ratio of the number of relevant movies over the top-N recommended movies for all test users. The definition of MAP is given as:

$$MAP = \frac{1}{K_{ts}} \sum_{u=1}^{K_{ts}} \frac{\sum_{j=1}^{N_u} (rel_u(j) \times P_u@j)}{\sum_{j=1}^{N_u} rel_u(j)}, \tag{12}$$

where K_{ts} is the number of users for testing, and N_u denotes the number of recommended movies for the user u. $rel_u(j)$ is a binary indicator, which is equal to 1 if the movie of rank j is relevant to user u, and is equal to 0 otherwise. $P_u@j$ is the precision of the top j recommended movies for the user u, that is, the ratio of movies in the top j recommendation that are relevant to the user u. Since it is required that relevant movies are recommended as early as possible, usually a small value of N is chosen for P@N. In our experiments, we evaluated in the cases of N = 1, 5, 10. In addition, MAP reflects the quality of the entire recommendation list by considering the positions of all the relevant movies. Higher values for P@N and MAP indicate a better recommendation performance.

5.1.3. Experimental Protocol. The Moviepilot challenge dataset contains three predefined subsets: a training set, a validation set and a test set. The training set involves all users and all movies. The training set is used to generate recommendations. The validation and test sets involve a small number of users, that is, 160 and 80 users, respectively, whose ratings are disjoint with their ratings in the training set. The validation set is used to tune the parameters in the proposed algorithm. The parameters in the baseline approaches, as discussed in Section 5.4, are also tuned to the validation set. Performance is reported based on recommendations for all the users in the test set, as demonstrated in Section 5.4. Moreover, according to the requirement in the Moviepilot challenge (cf. Section 2), the evaluation only concerns the movies with a specific mood tag (i.e., the mood with identifier 16) to be potentially relevant for the users. Recommended movies that are in the validation/test set, but do not have the specified mood tag, are counted as irrelevant for the target users, that is, they do not contribute to improvement in recommendation as reflected by the evaluation metrics.

Note that in the proposed JMF algorithm, we set the dimensionality of latent features to be 10. Although the variation of dimensionality of latent features could influence the performance, we notice that, just like in a common MF technique [Weimer et al. 2008], a further increase in the number of latent features would not introduce a large improvement, while requiring more computational cost (cf. section 4.4). The regularization parameter λ is set to 1 based on the observation of the performance of the basic matrix factorization, which is also discussed in section 5.4. The stopping condition ϵ in the learning process is set to 0.0001 in our experiments.

5.2. Impact of Trade-Off Parameters

The trade-off parameters α and β in the proposed algorithm influence the relative contributions from the contextual movie similarities. By using the validation set, we investigate the impact of the trade-off parameters by varying their values and measuring the recommendation performance in terms of P@5 and MAP. We first set $\beta = 0$ and investigate the impact of α, as shown in Figure 4. It can be seen that for both P@5 and MAP, the optimal value of α lies around 0.1. It also indicates that by only introducing the mood-specific movie similarity, additional improvement can be achieved over the basic MF model, that is, the case when $\alpha = 0$. Then, we further investigate the impact of β, and keep the value of α fixed as $\alpha = 0.1$, as shown in Figure 5. It can also be seen that for both P@5 and MAP, the optimal value of β is nearly 0.1. This indicates that in addition to the mood-specific movie similarity, there is still potential to further improve the recommendation performance by incorporating the PK-based movie similarity. Moreover, we can observe that the additional improvement stemming from the PK-based movie similarity is only slight compared to the case when only the mood-specific movie similarity is used, that is, $\beta = 0$ in Figure 5. This implies that the mood-specific movie similarity makes the major contribution among the used contextual information.

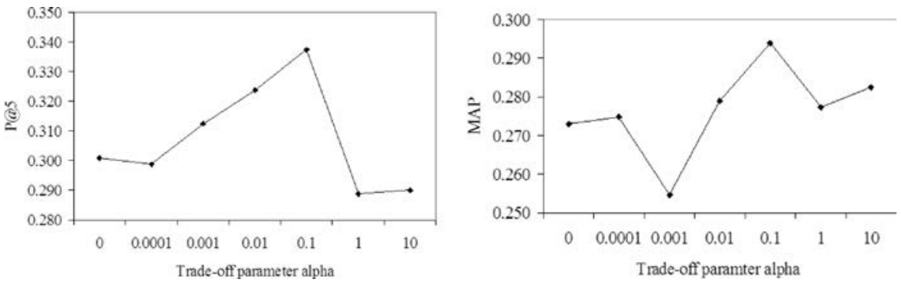


Fig. 4. The impact of trade-off parameter α on the recommendation performance of the proposed algorithm.

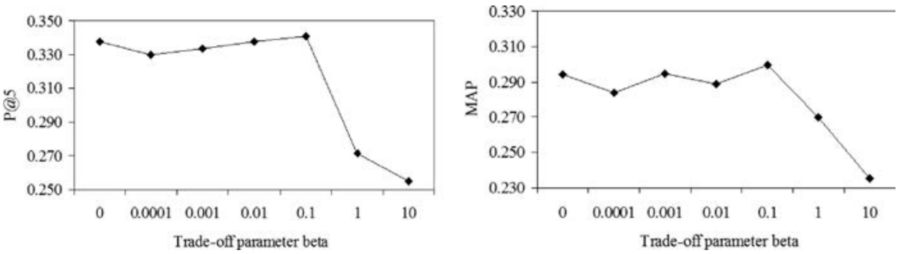


Fig. 5. The impact of trade-off parameter β on the recommendation performance of the proposed algorithm, when $\alpha = 0.1$.

5.3. Effectiveness

We further investigate the research question (1), namely to what extent the minimization of the objective function in Equation (9) contributes to the improvement of the recommendation. To this end, we demonstrate the variation of the output of the objective function (normalized for demonstration purposes) and evaluation metrics, that is, P@5 and MAP on the validation set, simultaneously during the iterations of the optimization process, as shown in Figure 6. Note that in this experiment the trade-off parameters used are the optimal values from the previous section, that is, $\alpha = 0.1$ and $\beta = 0.1$. The results show that the recommendation performance generally increases monotonically with the minimization of the objective function, allowing us to give a positive answer to the first research question.

5.4. Performance Comparison

In this section, we compare the performance of the proposed algorithm JMF-MS-PK with a set of alternative recommendation approaches listed below. The performance is reported based on the test set. The trade-off parameters are the optimal ones determined using the validation set as stated in Section 5.2.

—*PopRec*. Movies are recommended to users based on their popularity, which is defined in terms of the number of users who rated them in the training set. This approach constitutes a non-personalized and naive baseline since every test user will receive the same recommendation, that is, a list of movies ranked in a descending order according to the number of their ratings in the training set.

—*RWR*. This algorithm is a state-of-the-art recommendation approach that uses random walk with restarts (RWR) [Konstas et al. 2009] on a graph encoding the relations between the users and items. Here, we set the restart probability to 0.8 based on the optimal performance achieved in the validation set.

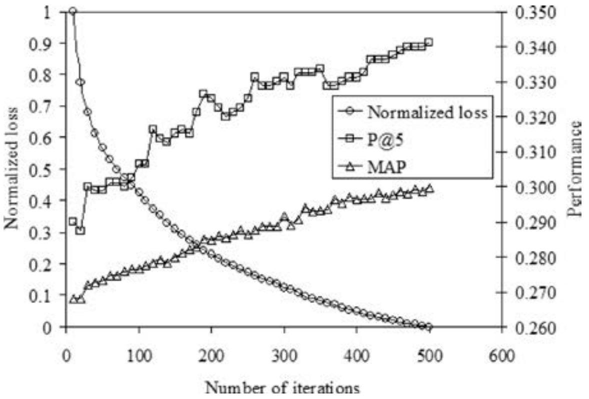


Fig. 6. The effectiveness of the proposed algorithm in improving the recommendation performance.

Table I.
Comparison of recommendation performance between the proposed algorithm and other baseline approaches.

	P@1	P@5	P@10	MAP
PopRec	0.213	0.248	0.251	0.264
RWR	0.238	0.253	0.274	0.281
MF	0.325	0.305	0.241	0.252
JMF-MB	0.338	0.328	0.286	0.273
JMF-MS	0.350	0.335^	0.295^	0.289^
JMF-MS-PK	0.363	0.335	0.306*	0.290

Note: "^" denotes a significant improvement of JMF-MS over JMF-MB, and "*" denotes a significant improvement of JMF-MS-PK over JMF-MS, according to Wilcoxon signed rank significance test with $p<0.05$.

—*MF*. This algorithm represents a basic state-of-the-art matrix factorization approach as in Koren et al. [2009], which is also equivalent to Equation (6). The dimensionality of the latent user and movie features is also set to 10. The regularization parameter λ is set to 1, which is tuned to achieve the optimal performance on the validation set. Note that the same corresponding parameters are used in the proposed algorithm as well.

—*JMF-MB*. This algorithm is a joint matrix factorization approach that incorporates the general mood-based movie similarity rather than mood-specific movie similarity. It also shares the same corresponding parameters as used in MF. In addition, we set the trade-off parameter to 0.01 to again give the best performance on the validation set. The JMF-MB is used to compare with the proposed algorithm especially for validating the usefulness of the mood-specific movie similarity.

—*JMF-MS*. This algorithm is a joint matrix factorization approach that incorporates the mood-specific movie similarity as in Equation (9). Note that in both JMF-MB and JMF-MS we do not use the PK-based movie similarity, that is, $\beta = 0$.

The results of the comparative analysis are shown in Table I, from which we can see the relative improvement achieved by the proposed algorithm JMF-MS-PK in terms of P@1, P@5, P@10, and MAP.

First, we can see that the JMF-MB approach outperforms non-context-aware approaches, that is, PopRec, RWR, and MF, by over 5% in terms of P@1, P@5 and P@10. This improvement is statistically significant (based on Wilcoxon signed rank significance test, $p<0.05$) in all cases and indicates that the contextual information indeed

Table II.
Comparison of P@10 performance between the proposed algorithm and other baseline approaches with respect to users with various numbers of rated movies.

Num. rated movies (Num. users)	MF	JMF-MB	JMF-MS	JMF-MS-PK
1∼50 (19)	0.305	0.326	0.374	0.379
51∼100 (16)	0.250	0.269	0.313	0.313
101∼150 (13)	0.208	0.223	0.238	0.238
151∼200 (12)	0.242	0.317	0.250	0.267
>200 (20)	0.195	0.285	0.270	0.300

has the potential to help improve the recommendation performance. By incorporating both the mood-specific movie similarity and the PK-based movie similarity, the proposed algorithm JMF-MS-PK outperforms the non-context-aware approaches by over 10% across all the evaluation metrics, also with constant statistical significance. This gives an affirmative answer to the research question (2). In addition, the improvement of JMF over the basic MF approach also empirically indicates that exploiting contextual information via the JMF model could indeed contribute to alleviating data sparseness problem in the user-movie rating matrix.

Second, we observe that the JMF-MS achieves around 3%–5% improvement over the JMF-MB in most of the evaluation metrics, indicating that the mood-specific movie similarity is more beneficial for the mood-specific recommendation purpose compared to the general mood-based movie similarity. This observation provides an affirmative answer to the third research question.

Third, by further incorporating the PK-based movie similarity in addition to the mood-specific movie similarity, the JMF-MS-PK achieves around 3% significant improvement over the JMF-MS with respect to P@10, which results in an affirmative answer to the last research question. It can also be seen that the mood-specific movie similarity makes the major contribution under the mood-specific recommendation purpose, while other contextual movie similarities, for instance, the PK-based movie similarity, can be exploited via the JMF approach to make further enhancements of the recommendation performance.

Finally, we investigate the performance of the proposed algorithm with respect to users with varying numbers of rated movies, such as the results reported for P@10 in Table II. Since the non-context-aware approaches perform generally worse than the proposed JMF algorithm as indicated before, we only select MF to represent the non-context-aware approaches. As can be seen from Table II, the JMF algorithms outperform the basic MF across all users with various numbers of rated movies. We also notice that the users with relatively fewer rated movies (i.e., with no more than 100 rated movies) benefit most from the mood-specific movie similarity (i.e., from JMF-MS and JMF-MS-PK) compared to the general mood-based movie similarity (JMF-MB). We conjecture that the difference arises due to the following effect. Under mood-specific similarity less orthogonal pairs of movies exist in the set than under mood-based similarity. Effectively, the structure of the movie similarity space as defined by pair-wise relationships between movies is smoothed by the use of mood-specific similarity. Users who have rated large numbers of items do not benefit from this smoothing since their profiles already contain enough information for reliable estimation of predictions. Users who have rated fewer items, however, effectively are able to cast a wider net if more pairs of movies are similar.

In real-world systems, users with limited numbers of rated movies are usually the majority in the community as reflected in the typically high data sparseness encountered by recommender systems. Our results indicate that exploiting the mood-specific movie similarity has the potential to benefit these users in particular. Compared to the JMF-MS, the JMF-MS-PK could be more beneficial for the users who rated relatively more movies. This observation may imply that the latent movie features that are learned from PK-based similarity can be better leveraged if there are more available ratings on those movies. If the user rates more movies, she could raise the chance that some of her rated movies are similar to other movies with respect to PK-based similarity, which could be recommended even if they are rated by few users overall within the collection. This observation also implies an affirmative answer to our last research question, that is, the additional information introduced using PKs serves to improve recommendations in the case of users rating many movies.

6. CONCLUSION AND FUTURE WORK

In this article, we present a novel context-aware recommendation algorithm that integrates contextual movie similarity, that is, mood-specific movie similarity and PK-based movie similarity, together with the user-movie rating matrix into joint matrix factorization for the purpose of mood-specific movie recommendation, as defined in the Moviepilot challenge. The proposed algorithm is analyzed to be scalable for large-scale use cases. Our experiments at the Moviepilot challenge dataset show that the proposed algorithm outperforms several other state-of-the-art recommendation approaches. Substantial improvement can be achieved by exploiting contextual movie similarities, among which the mood-specific movie similarity is shown to make the major contribution to the recommendation performance and the PK-based movie similarity could further enhance contribution. In addition, we specifically validate the usefulness of the mood-specific movie similarity compared to general mood-based movie similarity, which indeed leads to a substantial performance improvement. We also show the JMF with both mood-specific movie similarity and movie similarity in terms of plot keywords could be the most beneficial option for users across profiles containing different numbers of rated movies, compared to other variants.

We note that the algorithm proposed in this article could be generally applicable to other recommendation purposes, for instance, recommending movies with a specific actor, recommending music with a specific style. Exploiting contextual item similarity that is related to the specific recommendation purpose could be a way to make recommender systems context-aware.

Our future work will involve further exploration of the use of context-based information for recommendation. In particular, note that the Moviepilot challenge dataset was issued in an encoded form: we do not have direct knowledge of the identities of moods or plot keywords, rather these are represented within the dataset as codes. In the future, we would like to experiment on a dataset where we do have access to this information in order to compare our approach to approaches informed by external knowledge sources that could add explicit information on the relationship between the sources. We are also interested in moving beyond moods and plot keywords and understanding the suitability of our approaches for exploiting other sources of knowledge. In particular, we will address the question of what characteristics of a knowledge source must hold in order to be successfully exploited by our approach for the purpose of context-aware recommendation.

REFERENCES

ADOMAVICIUS, G. AND TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng. 17*, 6, 734–749.

ADOMAVICIUS, G., SANKARANARAYANAN, R., SEN, S., AND TUZHILIN, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst. 23*, 103–145.

ANAND, S. S. AND MOBASHER, B. 2007. In Contextual recommendation. *From Web to Social Web: Discovering and Deploying User and Content Profiles*, Springer, 142–160.

BALTRUNAS, L. AND RICCI, F. 2009. Context-based splitting of item ratings in collaborative filtering. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, 245–248.

BREESE, J. S., HECKERMAN, D., AND KADIE, C. M. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98)*. 43–52.

BURKE, R. 2002. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact. 12*, 331–370.

CANTADOR, I. AND CASTELLS, P. 2009. Semantic contextualisation in a news recommender system. In *Proceedings of the Workshop on Context-Aware Recommender Systems*.

CAROLIS, B. D., MAZZOTTA, I., NOVIELLI, N., AND SILVESTRI, V. 2009. Using common sense in providing personalized recommendations in the tourism domain. In *Proceedings of the Workshop on Context-Aware Recommender Systems*.

DENG, H., LYU, M. R., AND KING, I. 2009. Effective latent space graph-based re-ranking model with global consistency. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining (WSDM '09)*. ACM, New York, NY, 212–221.

DESHPANDE, M. AND KARYPIS, G. 2004. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst. 22*, 143–177.

GANTNER, Z., RENDLE, S., AND LARS, S.-T. 2010. Factorization models for context-/time-aware movie recommendations. In *Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa '10)*. ACM, New York, NY, 14–19.

GUNAWARDANA, A. AND SHANI, G. 2009. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res. 10*, 2935–2962.

HERLOCKER, J. L., KONSTAN, J. A., BORCHERS, A., AND RIEDL, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99)*. ACM, New York, NY, 230–237.

HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G., AND RIEDL, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. 22*, 5–53.

HOFMANN, T. 2004. Latent semantic models for collaborative filtering. *ACM Trans. Inf. Syst. 22*, 89–115.

KARATZOGLOU, A., AMATRIAIN, X., BALTRUNAS, L., AND OLIVER, N. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, 79–86.

KLEINBERG, J. AND SANDLER, M. 2008. Using mixture models for collaborative filtering. *J. Comput. Syst. Sci. 74*, 49–69.

KOLDA, T. G. AND SUN, J. 2008. Scalable tensor decompositions for multi-aspect data mining. In *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 363–372.

KONSTAS, I., STATHOPOULOS, V., AND JOSE, J. M. 2009. On social networks and collaborative recommendation. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, New York, NY, 195–202.

KOREN, Y. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. ACM, New York, NY, 447–456.

KOREN, Y., BELL, R., AND VOLINSKY, C. 2009. Matrix factorization techniques for recommender systems. *Computer 42*, 30–37.

LINDEN, G., SMITH, B., AND YORK, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput. 7*, 76–80.

LIU, N. N. AND YANG, Q. 2008. Eigenrank: a ranking-oriented approach to collaborative filtering. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '08)*. ACM, New York, NY, 83–90.

LIU, N. N., CAO, B., ZHAO, M., AND YANG, Q. 2010. Adapting neighborhood and matrix factorization models for context aware recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa '10)*. ACM, New York, NY, 7–13.

LIU, N. N., ZHAO, M., AND YANG, Q. 2009. Probabilistic latent preference analysis for collaborative filtering. In *Proceeding of the 18th ACM Conference on Information and Knowledge Management (CIKM '09)*. ACM, New York, NY, 759–766.

MA, H., LYU, M. R., AND KING, I. 2009. Learning to recommend with trust and distrust relationships. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, 189–196.

MA, H., YANG, H., LYU, M. R., AND KING, I. 2008. Sorec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. ACM, New York, NY, USA, 931–940.

RENDLE, S. AND SCHMIDT-THIEME, L. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*. ACM, New York, NY, 81–90.

RENDLE, S., BALBY MARINHO, L., NANOPOULOS, A., AND SCHMIDT-THIEME, L. 2009b. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. ACM, New York, NY, 727–736.

RENDLE, S., FREUDENTHALER, C., GANTNER, Z., AND SCHMIDT-THIEME, L. 2009a. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, VA, 452–461.

SAID, A., BERKOVSKY, S., AND DE LUCA, E. W. 2010. Putting things in context: Challenge on context-aware movie recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa '10)* ACM, New York, NY, 2–6.

SALAKHUTDINOV, R. AND MNIH, A. 2008. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*. Vol. 20.

SARWAR, B., KARYPIS, G., KONSTAN, J., AND REIDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, 285–295.

SHI, Y., LARSON, M., AND HANJALIC, A. 2009. Exploiting user similarity based on rated-item pools for improved user-based collaborative filtering. In *Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys '09)*. ACM, New York, NY, 125–132.

SHI, Y., LARSON, M., AND HANJALIC, A. 2010a. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys '10)*. ACM, New York, NY, USA, 269–272.

SHI, Y., LARSON, M., AND HANJALIC, A. 2010b. Mining mood-specific movie similarity with matrix factorization for context-aware recommendation. In *Proceedings of the Workshop on Context-Aware Movie Recommendation (CAMRa '10)*. ACM, New York, NY, 34–40.

SINGH, A. P. AND GORDON, G. J. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. ACM, New York, NY, 650–658.

SU, J.-H., YEH, H.-H., YU, P. S., AND TSENG, V. S. 2010. Music recommendation using content and context information mining. *IEEE Intell. Syst. 25*, 16–26.

SYMEONIDIS, P., NANOPOULOS, A., AND MANOLOPOULOS, Y. 2008. Tag recommendations based on tensor dimensionality reduction. In *Proceedings of the ACM Conference on Recommender Systems (RecSys '08)*. ACM, New York, NY, 43–50.

SYMEONIDIS, P., NANOPOULOS, A., AND MANOLOPOULOS, Y. 2010. A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. *IEEE Trans. Knowl. Data Eng. 22*, 179–192.

TSO-SUTTER, K. H. L., MARINHO, L. B., AND SCHMIDT-THIEME, L. 2008. Tag-aware recommender systems by fusion of collaborative filtering algorithms. In *Proceedings of the ACM Symposium on Applied Computing (SAC '08)*. ACM, New York, NY, 1995–1999.

WANG, X., SUN, J.-T., CHEN, Z., AND ZHAI, C. 2006. Latent semantic analysis for multiple-type interrelated data objects. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*. ACM, New York, NY, 236–243.

WEIMER, M., KARATZOGLOU, A., AND SMOLA, A. 2008. Improving maximum margin matrix factorization. *Mach. Learn. 72*, 263–276.

WEIMER, M., KARATZOGLOU, A., LE, Q., AND SMOLA, A. 2007. Cofirank: Maximum margin matrix factorization for collaborative ranking. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'07)*. 1593–1600.

Received December 2010; revised April 2011, July 2011, August 2011; accepted August 2011