

Matrix Factorization with Content Relationships for Media Personalization

Alexandros Nanopoulos¹, Dimitrios Rafailidis², Ioannis Karydis³

¹Catholic University of Eichstätt-Ingolstadt, Germany
alexandros.nanopoulos@ku.de

²Aristotle University of Thessaloniki, Greece
draf@csd.auth.gr

³Ionian University, Greece
karydis@ionio.gr

Abstract. Content personalization is identified as a key technology for enabling ubiquitous access to social media. Recommender systems implement media personalization, by suggesting relevant content and helping users in addressing the “information overload” problem. In this paper, our aim is to improve personalization by increasing the accuracy of recommendations. We propose a novel method, called Content Relationships Matrix Factorization (CRMF), which exploits additional information in the form of content relationships that express relevance between items. We model content relationships based on affinity graphs and use them in the context of matrix-factorization, which are currently the state-of-the-art prediction models for recommender systems. In our experimental evaluation with a real data set, we demonstrate the accuracy improvement of CRMF compared to matrix factorization models that do not take into account content relationships. Our experimental results show that CRMF compares favorably to the baseline method, demonstrating the usefulness of considering content relationships.

Keywords: Media; Personalization; Recommender Systems; Matrix Factorization; Content.

1 Introduction

Recent reports about trends in consumer-technology markets indicate that users, ubiquitously connected to social networks, place an ever increasing quantity of media online, thus, posing a challenge to traditional brand relationships and business models.¹ What is, therefore, required to address this challenge, is the development of services that will offer personalized access to content.

¹ Gartner, April 2012 (<http://www.gartner.com/it/page.jsp?id=1984415>)

Recommender systems, which suggest to users relevant content, are the key-technology for media personalization. Recommender systems traditionally develop models based on machine learning and statistics for predicting items suited to the personal preferences of users [1,12]. Such models learn users' preferences through either implicit feedback, such as click rates or time spent (e.g., track listening, video watching); or explicit feedback, such as ratings (e.g., in a 1-5 star scale).

The interest of researchers in recommender systems for media applications has increased rapidly in the previous years, mainly due to the Netflix Prize², an open competition for the best recommender system to predict user ratings for films, based on previous ratings, without providing any additional information about the users or films. Netflix Prize has clearly demonstrated the superiority of latent-factor models, especially matrix factorization [7], compared to classic collaborative-filtering techniques. Nevertheless, the problem of *sparsity* is still a major obstacle in the case of recommender systems for media, due to the appearance of *power laws* in users' preference data; i.e., a larger portion of the preferences data is available only for a very small percentage of users and items.

In this paper, we propose a novel way to address the aforementioned challenge by exploiting *content relationships* that express relevance between items (such items can have various media formats). Thus, in contrast to traditional recommender systems, our method takes into account not only data about preference of users to items, but additionally considers the relationships between items themselves; see Fig. 1. Content relationships provide an additional source of information that can be exploited to develop more accurate prediction models so as to improve recommendation.



Fig. 1.User preferences and content relationships.

Our approach models content relationships in the form of an *affinity graph* between items, with higher the affinity between two items the more strong their relationship. Nowadays, Web 2.0 content providers offer information about content relationships, and thus, enable the development of such affinity graphs. For instance, last.fm Web Services³ discloses information related to the artists, geography, usage (playlists, popularity), or social tags [9]. As another example, Flickr⁴ provides information about

²<http://www.netflixprize.com/>

³<http://www.lastfm.de/api>

⁴<http://www.flickr.com/services/api/>

relationships between collections pertaining to photographs, user comments and tags, as well as annotations about geo-location. In our study, we propose to utilize information offered by Web 2.0 content providers in order to develop affinity graphs used in the context of matrix-factorization, which constitute the state-of-the-art prediction models for recommender systems. Based on the premise that items with a strong relationship should be also related in the latent-factor space that matrix factorization creates, our proposed method is able to define an effective way of interaction between items during the learning process of matrix factorization, by updating the positions of items in the latent-factor space according to the position of their related items from the affinity graph. This approach is especially beneficial for items in the “*long-tail*” of power laws (i.e., items with sparse feedback data), because information from items with abundant feedback propagates to them. Our contributions are summarized as follows:

- We propose a novel and general approach to model content relationships between items through an affinity graph.
- We exploit affinity graphs to allow matrix-factorization models to propagate information between items during the learning process.
- We conduct experimental results with real data from the Million Song dataset and last.fm, which indicate the superiority of the proposed approach compared to the state-of-the-art matrix factorization method that does not utilize content relationships.

The rest of this paper is organized as follows. Section 2 describes the related work. In Section 3 we present the proposed method. Experimental results are presented in Section 4. Finally, Section 5 concludes this paper.

2 Related Work

2.1 Collaborative Filtering (CF)

Collaborative Filtering (CF) systems [14] generate predictions based on preference data (e.g., ratings) of similar users. CF has attracted a lot of interest and researches have been improving its performance continually. Users of CF systems receive recommendations mainly based on *memory-based* (a.k.a., nearest-neighbor) algorithms, which can be either user-based or item-based.

User-based CF first finds for users a rating pattern similar to the one of the active user, and then uses the ratings of these users to perform the predictions about the ratings of the active user for specific items, by calculating the weighted average of the ratings of similar users for the same item. In contrast, item-based CF generates predictions by first detecting asset of similar items. CF, due to its simplicity and efficacy, has attracted popularity in e-commerce applications. For example, Amazon’s recommendation system has been reported to use item-based CF [8].

2.2 Content-based (CB) Recommender Systems

Content-based (CB) recommender systems [11] typically perform predictions by utilizing content features of items. In particular, based on content features, similarity functions can detect the most related items, following the assumption that items that are similar in content will be rated similarly. The main advantage of CB recommender systems is that their performance is not based on the existence of preference data, and thus they are suitable in the case of the so called “cold-start” problem. However, pure CB recommender systems ignore information about user preferences, in case it is available, which can lead to overspecialization and low level of personalization.

CB recommender systems are used in several real-world applications. Pandora Radio is a popular example that recommends music with similar characteristics to that of a song provided by the user as an initial seed.

2.3 Hybrid Recommender Systems (HS)

As mentioned previously, CF and CB recommender systems have advantages and disadvantages. The important difference between them is that CF systems are based mainly on preference data, whereas CB systems are based on the content of items. For this reason, recent research has focused on the combination of CF and CB, which led to the so called *hybrid* recommender systems [4]. A prominent hybrid method is the Content-Boosted Collaborative Filtering (CBCF) [10], which learns a content-based model over the training data to generate ratings for unrated items. This process results in a dense rating matrix, because of the predictions made for all empty places in the original rating matrix (i.e., cells without any given rating). The derived matrix is then used by a CF recommender system.

2.4 Model-based Recommender Systems (MB)

Model-based recommender systems differ from the previously mentioned categories, because they develop a prediction model based on patterns detected in training data. Several techniques have been used for this purpose, such as Bayesian networks, clustering models, probabilistic latent semantic analysis, etc. [15].

Matrix factorization is technique that has demonstrated its ability to build accurate prediction models for model-based recommender systems [7]. Matrix factorization generates recommendations based on latent features that determine users’ preferences. A state-of-the-art matrix factorization method is called Probabilistic Matrix Factorization (PMF) [13].

2.5 Comparison of Approaches and Motivation

Table 1 provides a comparison of the presented approaches in recommender systems, by summarizing their advantages and disadvantages. It also reports (last column) whether the approaches take into account content relationships.

Table 1. Comparison of approaches in recommender systems.

Approach	Comparison		Content Relationships
	Advantages	Disadvantages	
CF	explainability of results, easy to implement; fast execution	accuracy decreases with sparsity and cold-start	no
CB	addresses cold-start	not high accuracy	partially
HS	improved accuracy compared to CF and CB	high complexity and expensive to implement	partially
MB	handles sparsity; best reported accuracy	expensive model building	no

As a main conclusion from Table 1, model-based (MB) approaches present the best accuracy of recommendations compared to the rest approaches (CF, CB, and HS). Recent advances in the field of recommender system have demonstrated the superiority of model-based approaches, which are nowadays considered as state-of-the-art methods [7]. For this reason, in contrast to CF, CB, and HS approaches, our proposed method is model-based.

However, existing MB approaches, in contrast to CB and HS approaches, do not take into account content relationships. This fact forms the motivation behind our study. Our proposed method opts for combining the superior accuracy of state-of-the-art model-based approaches with the ability of CB and HS approaches to take into account content relationships.

Additionally, compared to CB and HS, our proposed method opts for generalizing content relationships, which are only partially considered by CB and HS, because they are based solely on the similarity between items. Our proposed method uses the concept of affinity graphs to model potentially more general relationships. Moreover, compared to CB, our proposed method does not use only relationships between items, since it considers preference data, too. Finally, compared to HS, our proposed method combines content relationships and preference data, but this is performed in a principled, model-based way, and thus it does not make arbitrary combinations of CF and CB, in contrast to most of the variations of hybrid techniques.

Recently, Jamali and Ester [5] proposed an extension to PMF, which takes into account social relationships between users. Our proposed method shares with this work the principle of considering relationships between users or items, but is complementary to [5], because it considers relationships between items instead of users. Moreover, our proposed method considers affinity graphs with varying degrees.

3 Content Relationships Matrix Factorization

In this section we describe our proposed method, denoted as Content Relationships Matrix Factorization (CRMF). We first describe the problem definition and notation,

then we provide the necessary background information about generic matrix factorization models, and finally we explain how CRMF extends matrix factorization through the use of affinity graphs that represent content relationships.

3.1 Problem Definition

Assume that D_U is a domain of N users and D_I is a domain of M items. As described in Section 1, users can express their preference to items either implicitly or explicitly. In this paper, we focus on explicit information in the form of an $N \times M$ rating matrix $R = [R_{u,i}]$, where each element $R_{u,i}$ denotes the rating of the u -th user of D_U on the i -th item of D_I . If such rating is not available, then element $R_{u,i}$ is assumed to be unknown (i.e., null). Matrix R is usually very sparse, since most of its elements are expected to be missing.

An affinity graph is represented with an $M \times M$ matrix $A = [A_{i,j}]$, which expresses the content relationships between items. Each element $A_{i,j}$ is set to one, if the i -th item of D_I is related to the j -th item of D_I ; otherwise is set to zero. Matrix A can be in general non-symmetric, but in our experimental study we will use a symmetric one.

The recommendation problem that we study is described in technical terms as follows: Given a matrix R with existing ratings of users from domain D_U on items from domain D_I , as well as an affinity matrix A , the goal is to predict the rating $R_{u,i}$, in case $R_{u,i}$ is unknown. Our aim is to improve the accuracy of predicted ratings.

3.2 Matrix Factorization Models

Matrix factorization techniques create a latent-feature D -dimensional space in which they represent each user and item. Let $U \in \mathbf{R}^{D \times N}$ be a $D \times N$ matrix whose u -th column vector, denoted as U_u , represents the coordinates of the u -th user of domain D_U in this D -dimensional space.⁵ Similarly, let $V \in \mathbf{R}^{D \times M}$ be a $D \times M$ matrix whose i -th column vector, denoted as V_i , represents the coordinates of the i -th item of domain D_I in the same D -dimensional space.

Using a $N \times M$ rating matrix R as training data, matrix factorization techniques *learn* (i.e., compute the elements of) matrices U and V so that they can approximate matrix R with matrix \hat{R} such that:

$$R \approx \hat{R} = UV^T \quad (1)$$

⁵ \mathbf{R} denotes the set of real numbers.

Having learned matrices U and V , and computed matrix \hat{R} based on Equation 1, each element $\hat{R}_{u,i}$ comprises the prediction for the rating of the corresponding u -th user of D_U on the corresponding i -th item of D_I .

The process of learning matrices U and V can be expressed in a probabilistic framework developed by Salakhutdinov and Mnih [13]. According to this framework, the likelihood of observing a specific set of ratings represented with a given rating matrix R , can be expressed as:

$$p(R|U, V; \sigma_R^2) = \prod_{u=1}^N \prod_{i=1}^M \left[\mathfrak{N}(R_{u,i} | U_u^T V_i; \sigma_R^2) \right]^{1_{R(u,i)}} \quad (2)$$

where $\mathfrak{N}(\mu, \sigma^2)$ denotes the *normal distribution* with mean μ and variance σ^2 ; and $1_{R(u,i)}$ denotes the *indicator function* with value 1 when element $R_{u,i}$ is known (i.e., not null), or 0 otherwise. More precisely, Equation 2 makes the premise that each known rating, represented with the element $R_{u,i}$, is an independent and identically distributed (*iid*) random variable that follows a normal distribution whose mean value is equal to the element $\hat{R}_{u,i} = U_u^T V_i$ (see Equation 1) and whose variance σ_R^2 is treated as a *hyper-parameter*⁶.

Based on Bayes theorem, from the likelihood function of Equation 2 we can obtain the posterior probability of U and V :

$$p(U, V | R; \sigma_R^2, \sigma_U^2, \sigma_V^2) \propto \prod_{u=1}^N \prod_{i=1}^M \left[\mathfrak{N}(R_{u,i} | U_u^T V_i; \sigma_R^2) \right]^{1_{R(u,i)}} \times \prod_{u=1}^N \left[\mathfrak{N}(U_u | 0; \sigma_U^2 I) \right] \times \prod_{i=1}^M \left[\mathfrak{N}(V_i | 0; \sigma_V^2 I) \right] \quad (3)$$

where Equation 3 makes the premise that the coordinates U_u of each user u , as well as the coordinates V_i of each item i in the D -dimensional latent space, are also *iid* random variables following normal distribution with zero mean and variances σ_U^2 and σ_V^2 , respectively (both variances are treated as hyper-parameters).

In this section we presented the representation of the original rating matrix R by an approximation matrix \hat{R} that is a product of matrices U and V . Matrix \hat{R} is not sparse and its elements give the predicted ratings. To compute \hat{R} we have to find those U and V matrices that maximize the probability of observing the given ratings in matrix R , i.e., we pose the computation of U and V as a problem of maximizing the posterior probability in Equation 3. The procedure for this maximization is explained in the following.

⁶ A hyper-parameter is a parameter that is not automatically tuned by the learning algorithm and, thus, left to be tuned “manually” using a cross-validated grid search.

3.3 Exploiting Content Relationships

The proposed method, denoted as *Content Relationships Matrix Factorization* (CRMF), extends the matrix factorization models presented in Section 3.2. CRMF exploits an affinity graph, A , between items, as they it has been defined in Section 3.1.

Dependence on Affinity Graph. Given an affinity graph, A , CRMF has the following principle: Consider the i -th and the j -th item of domain D_I . Let V_i and V_j denote the coordinate vectors of the i -th and the j -th item, respectively, in the D -dimensional latent-feature space. If the i -th item is related to the j -th item, i.e., $A_{i,j}=1$, then the learning process of matrix factorization should compute V_i by taking into account V_j . This should hold for all items related to the i -th item through the affinity graph A . Equation 4 expresses this dependency of matrix V on the affinity graph A :

$$p(V|T; \sigma_V^2, \sigma_A^2) \propto p(V|T; \sigma_A^2) \times p(V|\sigma_V^2) \quad (4)$$

The second factor $p(V|\sigma_V^2)$ in Equation 4 is the prior probability of V , for which we make the same assumption as in Equation 3, i.e., that V is an *iid* random variable following normal distribution with zero mean and variances σ_V^2 . The first factor $p(V|T; \sigma_A^2)$ expresses the dependence of V on A . We make the same assumption for normal distribution, thus:

$$p(V|T; \sigma_A^2) = \prod_{i=1}^M \left[\mathfrak{N} \left(V_i \mid \sum_{j \in N(i)} A_{i,j} V_j; \sigma_A^2 I \right) \right] \quad (5)$$

where $N(i)$ denotes the neighborhood of the i -th item, i.e., all items j for which $A_{i,j} = 1$. More specifically, Equation 5 assumes that the coordinate vector of the i -th item, denoted as V_i , follows normal distribution with mean equal to the average of the coordinates of the items that belong to its neighborhood $N(i)$.

Based on Equation 5, we reformulate the posterior probability of U and V (see Equation 3) as follows:

$$p(U, V | R; \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_A^2) \propto \prod_{u=1}^N \prod_{i=1}^M \left[\mathfrak{N} \left(R_{u,i} \mid U_u^T V_i; \sigma_R^2 \right) \right]^{1_{R(u,i)}} \times \prod_{u=1}^N \left[\mathfrak{N} \left(U_u \mid 0; \sigma_U^2 I \right) \right] \times \prod_{i=1}^M \left[\mathfrak{N} \left(V_i \mid \sum_{j \in N(i)} A_{i,j} V_j; \sigma_A^2 I \right) \right] \times \prod_{i=1}^M \left[\mathfrak{N} \left(V_i \mid 0; \sigma_V^2 I \right) \right] \quad (6)$$

Objective Function and Gradient Descent. Equation 6 provides the basis for learning U and V by exploiting the existence of the affinity graph A . The learning procedure is performed by finding those U and V variables that maximize the posterior probability of Equation 6. Since the natural logarithm function

$\ln\left(p\left(U, V | R; \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_A^2\right)\right)$ is monotonically increasing, we proceed by minimizing its arithmetic-negation function $L(U, V) = -\ln\left(p\left(U, V | R; \sigma_R^2, \sigma_U^2, \sigma_V^2, \sigma_A^2\right)\right)$ instead of maximizing directly Equation 6. This gives the following objective function $L(U, V)$ for which we seek the values of variables U and V that minimize it⁷:

$$L(U, V) = \frac{1}{2} \sum_{u=1}^N \sum_{i=1}^M 1_R(u, i) (R_{u,i} - U_u^T V_i)^2 + \frac{\mathcal{G}_U}{2} \sum_{u=1}^N U_u^T U_u + \frac{\mathcal{G}_V}{2} \sum_{i=1}^M V_i^T V_i + \frac{\mathcal{G}_A}{2} \sum_{i=1}^M \left(\left(V_i - \sum_{j \in N(i)} A_{i,j} V_j \right)^T \left(V_i - \sum_{j \in N(i)} A_{i,j} V_j \right) \right) \quad (7)$$

where $\mathcal{G}_U, \mathcal{G}_V, \mathcal{G}_A$ are the regularization hyper-parameters that are equal to $\sigma_R^2 / \sigma_U^2, \sigma_R^2 / \sigma_V^2, \sigma_R^2 / \sigma_A^2$, respectively, which help in avoiding model overfitting.

To minimize $L(U, V)$, which is a convex function, we can use the gradient descent on $\partial L / \partial U_u$ and $\partial L / \partial V_i$ for each coordinate pair U_u and V_i , and repeatedly update their values. In each repetition, called *epoch*, updating is performed according to the following rules:

$$U_u = U_u - \gamma \frac{\partial L}{\partial U_u} \quad V_i = V_i - \gamma \frac{\partial L}{\partial V_i} \quad (8)$$

where γ is the learning rate, which controls the speed of convergence.

In summary, this section presents a way to take into account the information in the affinity matrix A , by relating with Equation 5 the latent features of each item, given in matrix V , with the latent features of all its related items, given in matrix U . This allows to extend the problem of finding the matrices U and V that maximize Equation 3, by additionally incorporating in Equation 6 the connections between related items are given through Equation 5. Finally, by equivalently restating the problem as minimization in Equation 7, we presented its solution based on the gradient descent method, a fast and effective way to solve optimization problems. These characteristics of the gradient descent method are suitable in the examined case due to the large size of the data. The performance of the aforementioned procedure is examined experimentally in the next section.

4 Experimental Evaluation

In this section, we present the experimental evaluation of the proposed method (CRMF) based on a real dataset. We consider as baseline the state-of-the-art PMF matrix factorization of [13], which does not utilize any content relationships. This way, we can demonstrate the superiority of CRMF against PMF due to the utilization

⁷ Please note that in $L(U, V)$ we keep only the terms that depend on the variables U and V .

of content relationships. In the rest of this section, we first describe in Section 4.1 the formation of the real data set, in Section 4.2 we explain the experimental set up, whereas Section 4.3 gives the experimental results. Finally, in Section 4.4 we present the discussion of these results.

4.1 Data Set

Experimental evaluation of the proposed method was performed based on the Million Song dataset⁸ and last.fm. To the best of our knowledge, this is the only publicly available dataset containing observations expressed through users’ preferences, along with information about relationships between items. In our future work, we intend to examine cases with different types of content, when such data sets become publically available.

The items in the examined case are songs of various genres and their relationships are expressed through similarities provided by last.fm. For each user, the data set contains the number of playcounts, which denotes the number of times the user has listened to a song, where the case of zero playcount is not explicitly provided and represents missing preference (corresponding to null, as described in Section 3.1).

In our experiments, playcounts are considered as a measure of preference of users to songs, since the more times a user listens to a song, the more it is assumed that the user prefers this song. To clean noise from this data set, we filtered out as spurious observations with playcounts lower than 2 and greater than 15, since the former can happen unintentionally and the latter are outliers of the dataset, comprising 2.25% of the total observed ratings, which may have resulted by automatic crawlers. This filtering process resulted into 11,190,628 playcounts for 892,237 users and 296,604 songs. In addition, to map the playcounts into a 5-stars rating scale, which is popular in e-commerce sites, we performed equal-frequency binning of playcounts, using 5 bins as presented in Table 2. Bins in this case correspond to stars, e.g., playcounts that belong to the second bin corresponds to 2 stars.

Table 2. Equal-frequency binning of playcounts into 5 bins/stars

Bin/Stars	Playcounts	Num. of observed ratings	Percentage
1	=3	3,214,271	28.72%
2	=4	1,805,081	16.13%
3	=5	2,250,999	20.12%
4	=6	1,680,384	15.02%
5	≥ 7	2,239,893	20.02%

Finally, we focused on the more dense part of the dataset, in order to examine the collaborative effects between users. Thus, we applied the commonly used technique

⁸ Retrieved from <http://labrosa.ee.columbia.edu/millionsong/>

of p-core filtering [2]. The p-core of level p has the property, that each user and song has/occurs in at least p observations (in our experiments we set p to be 0.001% of the total number of playcounts).

The distributions of users and songs in the evaluation dataset are illustrated in Fig.2(a) and (b), respectively. In Fig.2(a) we notice that only few users (below 0.05%) have many observed ratings; moreover Fig.2(b) shows that the number of observed ratings of the majority of songs is rather low. Therefore, the problem of cold-start for users and songs appears in the examined dataset, as also appears in real-world recommender systems. As we experimentally show, this impacts negatively the performance of the state-of-the-art PFM method [13], which does not consider content relationships.

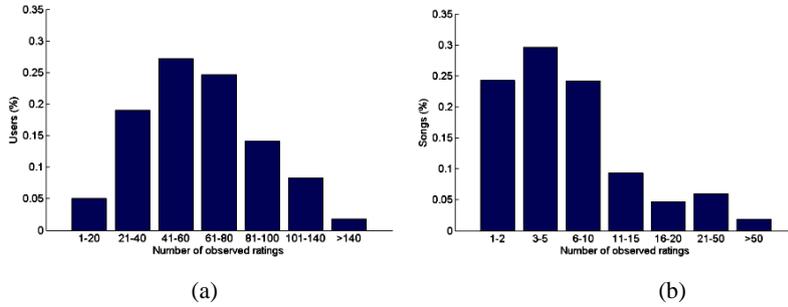


Fig. 2.Distribution of users (a) and songs (b) in the evaluation dataset.

For each song s , up to k related songs (called neighbors of s) are provided by last.fm as those songs that are more similar to s (last.fm determines similarity between songs based on user listening criteria). In our experiments, we set k equal to 5. However, for each song s , the actual number of related songs (denoted as neighborhood degree) varies, as depicted in Fig.3, which indicates that the majority of songs has neighborhood degree lower than 4.

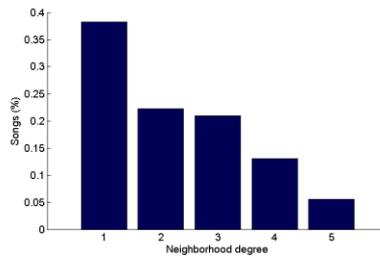


Fig. 3.Neighborhood degree distribution (the vertical axis is in the range [0, 1]).

4.2 Experiment Setup

To measure performance, we split the data set as follows: 50% of data are used each time as the training set and the remaining 50% as the test data. Our reported results are the averages out of 5 executions with different 50%-50% splits. In our experiment we use the evaluation metric of Root Mean Squared Error (RMSE), defined as:

$$RMSE = \sqrt{\frac{\sum_{(u,s) \in R_{test}} (r_{u,s} - \hat{r}_{u,s})^2}{|R_{test}|}}$$

where R_{test} is the set of all pairs (u,s) in the test data.

In all our experiments, we set hyper-parameters $\mathcal{G}_U = \mathcal{G}_V = 1$, since we found that for these settings the lowest RMSE for the PMF and CRMF methods are achieved. We tuned hyper-parameters \mathcal{G}_A based on grid searching over the training set.

4.3 Experiment Results

First, we evaluate the convergence of PMF and CRMF in terms of RMSE as function of the number of epochs. Fig. 4 presents the results, where $D=20$ was the number of dimensions of the latent-feature space. CRMF has lower RMSE than the original PMF. The improved accuracy of CRMF can be explained by the positive effect of the content relationships that bring additional information about items with fewer ratings, a problem which appears in our evaluation dataset, as presented in Section 4.1.

Additionally, CRMF converges after a smaller number of epochs (in this case, in 40 epochs) compared to PMF (in this case, at least 80 epochs are required). This means that CRMF requires less run time for building the prediction model.

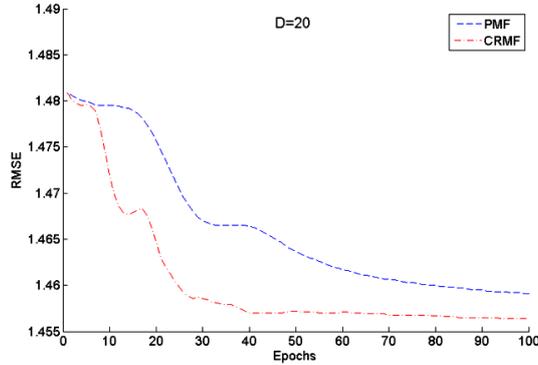


Fig. 4. RMSE of CRMF and PMF for $D=20$ dimensions of the latent feature vectors, by varying the number of epochs.

Next, in Fig. 5 we compare CRMF with PMF in terms of RMSE, by varying the dimensionality D of the latent-feature space. We can observe that CRMF has lower

RMSE than PMF for all different number of examined dimensions. By considering the high impact of the dimensionality of the latent-feature space on the model complexity, as described in [13], the lower D value for the proposed method results into lower computational complexity compared to the original PMF method.

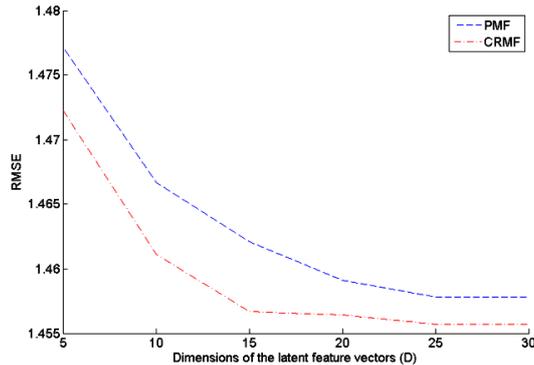


Fig. 5. RMSE of PMF and CRMF by varying dimensionality of the latent feature vectors (D).

4.4 Discussion

Assessment of experimental results. Our experimental results showed that CRMF compares favorably to PMF in terms of achieving a lower RMSE. Based on the comparison of approaches in Section 2.4, PMF is a state-of-the-art model-based method, which has been demonstrated to be more accurate (i.e., offering lower RMSE) than the rest approaches. Therefore, CRMF provides a clear advancement of the state-of-the-art. The source of this advancement by CRMF, in comparison to PMF, is the extension of the optimization problem considered by PFM (see Equation 3) in order to take into account content relationships that are described in Section 3.3 (see Equation 5). The reason that content relationships provide an improvement is the fact that they comprise an additional source of information that is exploited to overcome the problem of sparsity which incurs in ratings data. Since sparsity exists in all real-world recommender systems, our results demonstrate that exploiting additional sources of information, such as the content relationships, can be beneficial.

Application in real-world recommender systems. The presented experimental results demonstrate the suitability of the proposed CRMF method for a wide range of real-world applications, including media personalization for mobile devices (e.g., formation of playlists), home entertainment (e.g., suggestions about movies or music), or e-commerce involving media (e.g., sales of DVDs), as well as other types of goods when relationships can be determined between them. CRMF can be easily integrated in existing recommender systems for the aforementioned applications, as it does not add significant complexity in terms of its implementation and due to its run-time performance, compared to state-of-the-art model-based approaches that are becoming increasingly popular in real-world recommender systems. Regarding the additional

source of information, i.e., content relationships, that is exploited by CRMF, a large number of databases already exists for this purpose. Our experimental investigation showed this for publically available data of prominent media provider in the music domain. A similar approach can be followed for other media types, too; for instance, movies or images. In all these cases, content relationships can be formed based on several options (see also the related discussion in Introduction), such as the similarity between items computed based on the content or/and on other features, such as textual annotations; their usage data (e.g., how often are items consumed together); or user-provided social input about them (e.g., social tags, geo-tags, etc.). These kind of data needed to form content relationships, are nowadays available in most applications of online media and, therefore, they can be easily integrated in the proposed method.

Economic value. Regarding the economic value due to the improvement in recommendation accuracy achieved by the proposed method, experience in applications of recommender systems has shown that even small improvement in RMSE translates into improvement that is very important for the quality of recommendations. The reason is that such improvement can make a big positive difference in the identity of the most recommended items for a user [6]. Based on additional studies about online consumer-generated reviews, improvement in accuracy of recommendations has been shown to have a positive impact on purchase behavior, since consumers report being willing to pay from 20% to 99% more for a 5-star-rated item than a 4-star-rated item.⁹ Therefore, the application of the proposed CRFM in real-world recommender systems has the potential of improving customer attraction, satisfaction and retention. Finally, although we focused our investigation on personalization of media, it is possible to extend the use of CRMF for other types of goods, e.g., in the case of e-commerce. In such cases, content relationships can be formed based on available information sources, e.g., information about co-purchase of items in the same basket, which are commonly maintained in such settings. This fact shows the wide range of possible applications that the proposed method can find.

5 Conclusions

Personalized access to content has been identified as a key consumer technology for users that ubiquitously access media through social networks. Recommender systems enable media personalization, by suggesting relevant content to such users and helping them to address the “information overload” problem.

In this paper, our aim was to improve personalization by increasing the accuracy of recommendations. We proposed a novel method, called Content Relationships Matrix Factorization (CRMF), which exploits an additional source of information in the form of content relationships that express relevance between items. We performed an experimental evaluation of the accuracy of CRMF compared to a state-of-the-art matrix

⁹www.comscore.com/Press_Events/Press_Releases/2007/11/Online_Consumer_Reviews_Impact_Offline_Purchasing_Behavior

factorization that does not take into account content relationships. Our experimental results with a real data set showed that CRMF compares favorably to the baseline method, demonstrating the usefulness of considering content relationships.

In our future work, we will extend our approach in order to examine both content and social relationships (i.e., relationships between users), whenever they are both concurrently available.

References

1. G. Adomavicius, A. Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734–749, 2005.
2. V. Batagelj, M. Zaversnik. "Generalized cores", retrieved from arxiv.org/abs/cs/0202039, 2002.
3. T. Bertin-Mahieux, D.P.W. Ellis, B. Whitman, P. Lamere. "The Million Song Dataset". *International Conference on Music Information Retrieval (ISMIR)*, pp.591-596, 2011.
4. R. D. Burke. "Hybrid Web Recommender Systems". *The Adaptive Web*, Springer, 2007.
5. M. Jamali, M. Ester. "A matrix factorization technique with trust propagation for recommendation in social networks". *ACM conference on Recommender systems (RecSys '10)*, pp.135-142, 2010.
6. Y. Koren. "How useful is a lower RMSE?". *Netflix Prize Forum* (<http://www.netflixprize.com/community/viewtopic.php?id=828>), 2007.
7. Y. Koren, R. Bell, C. Volinsky. "Matrix Factorization Techniques for Recommender Systems". *IEEE Computer*, Vol. 42, No. 8, 2009.
8. G. Linden, B. Smith, and J. York. "Amazon.com Recommendations: Item-to-Item Collaborative Filtering". *IEEE Internet Computing*, 7(1), 2003.
9. L. B. Marinho, A. Hotho, R. Jaeschke, A. Nanopoulos, S. Rendle, L. Schmidt-Thieme, G. Stumme, P. Symeonidis. "*Recommender Systems for Social Tagging Systems*". Springer, 2012.
10. P. Melville, R. J. Mooney, R. Nagarajan. "Content-Boosted Collaborative Filtering for Recommendations". *Conference on Artificial Intelligence(AAAI)*, pp.187–192, 2002.
11. M. J. Pazzani, D. Billsus. "Content-Based Recommendation Systems". *The Adaptive Web*, Springer, 2007.
12. F. Ricci, L. Rokach, and B. Shapira. "*Recommender Systems Handbook*". Springer, pp. 1-35, 2011.
13. R. Salakhutdinov, A. Mnih. "Probabilistic matrix factorization". *Neural Information Processing Systems Conference (NIPS)*, 2008.
14. J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. "Collaborative Filtering Recommender Systems". *The Adaptive Web*, Springer, 2007.
15. X. Su, T. M. Khoshgoftaar. "A survey of collaborative filtering techniques". *Advances in Artificial Intelligence*, 2009.