Hybrid Recommender Systems based on Content Feature Relationship

Ehsan Aslanian, Mohammadreza Radmanesh and Mahdi Jalili, *Senior Member, IEEE*

***Abstract*— Recommendation systems get ever-increasing importance due to their applications in both academia and industry. The most popular type of these systems, known as collaborative filtering algorithms, employ user-item interactions to perform the recommendation tasks. With growth of additional information sources other than the rating (or purchase) history of users on items, such as item descriptions and social media information, further extensions of these systems have been proposed, known as hybrid recommendation algorithms. Hybrid recommenders use both user-item interaction data and their contextual information. In this work, we propose new hybrid recommender algorithms by considering the relationship between content features. This relationship is embedded into the hybrid recommenders to improve their accuracy. We first introduce a novel method to extract the content feature relationship matrix, and then the collaborative filtering recommender is modified such that this relationship matrix can be effectively integrated within the algorithm. The proposed algorithm can better deal with the cold-start problem than the state-of-art algorithms. We also propose a novel content-based hybrid recommender system. Our experiments on a benchmark movie dataset show that the proposed approach significantly improves the accuracy of the system, while resulting in satisfactory performance in terms of novelty and diversity of the recommendation lists.**

***Index Terms*— Social networks, recommender system, content-based recommenders, collaborative filtering, matrix completion.**

I. Introduction

Recommendation is a basic need in online stores, where the customers try to discover appropriate items among the ever-growing number of items provided. These customers not only expect the recommendation system to find good items for them, but also they care about the novelty and diversity of the items in the recommendation list. Therefore, system owners often employ sophisticated algorithms to use the available data (e.g., users rating/purchase history on items and contextual information on the users and items) to provide efficient recommendation list for a target user.

In the last decade, due to high availability of data from online stores and product ratings, Recommendation Systems (RSs) have been widely studied as an academic major [1-8]. RSs implicitly learn the taste of users and take it into account to find the most proper items for them. Indeed, recommendation is a prediction problem which tries to predict the preference of a user to an unobserved item given the user's rating history over some other items. This history can be provided as explicit ratings or implicit user interactions with the items. Such interactions are often modeled as a bipartite network with links from users to items. Additionally, some other information such as users/items contextual information may also be available. Hybrid RSs use both contextual information and users-items interaction data to design efficient recommenders [9-11].

Ehsan Aslanian is with School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran. Mohammadreza Radmanesh is with Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. Mahdi Jalili is with School of Engineering, RMIT University, Melbourne, Australia. Address for correspondence: Mahdi Jalili (mahdi.jalili@rmit.edu.au). Australian Research Council supported this research through project No. DE140100620.

Due to unavailability of contextual information in most cases, the vast majority of RS algorithms are based on users-items interaction data. Collaborative Filtering (CF) algorithms are the most successful methods in both academia and industry, which use these interaction data to perform different recommendation tasks [4, 12-15]. Although CF methods often achieve the highest accuracy on benchmark datasets (with sufficient interaction data), they have some major problems in real systems. The presence of cold-start users and/or items is one of the main issues of CF methods in many systems [16]. Cold-start refers to insufficient amount of ratings for new users/items that makes it difficult (or impossible) to provide reliable rating prediction for such users/items. Using contextual information along with users-items interaction data is one of the ways to handle cold-start problem in RSs. Employing different information sources to create hybrid RSs can improve existing CF algorithms. A well-known example of such algorithms is content-boosted CF [17], which enriches the rating matrix by content-based preferences, leading to improvements in the accuracy of CF algorithms. Different types of hybrid RSs, from simple to complex methods, have been proposed in the literature. Most of them can be categorized into mixed, switched, weighted, feature-augmentation and meta-level hybrid methods [10].

In this paper we introduce the concept of content feature relationship, and design two hybrid methods based on this relationship. The first algorithm uses a strategy to deal with the cold-start problem and the second algorithm provides an efficient approach for top-*N* recommendation. There are a number of works in the literature using concepts similar to the content feature relationship for handling the cold-start problem. Gunawardana and Meek proposed a unified approach to build hybrid RSs [9]. In their approach items and features are uniformly embedded in the model and their relationship is learned. Kouki *et al.* introduced a probabilistic model based on statistical relational learning framework to design hybrid RSs, combining contextual information from multiple sources to improve the quality of recommendations [18]. What discriminates our work from these algorithms is that they are basically probabilistic. Probabilistic models are usually difficult to implement and impose high computational complexity, making them not practical for commercial systems. Our proposed method is a vector-based model with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2631138, IEEE Transactions on Industrial Informatics

simple implementation, and yet satisfactory performance.

It has been shown that pure Singular Value Decomposition (PureSVD) and non-normalized cosine similarity item-item CF (NNCosKNN) show good performance in top-N recommendation [19]. More recently Ning *et al.* studied sparse linear methods [20] that provide similar accuracy and superior computational performance. However, the iterative implementation of PureSVD makes this method to have both good performance and relatively low computational complexity. In this manuscript, we compare the performance of our proposed method with these state-of-the-art methods and show its superiority in terms of both accuracy and novelty measures. Our main contributions in this work are:

- Introducing the concept of content feature relationship and an innovative method for extracting such relationship;
- Introducing a new approach for dealing with the cold start problem (when there are new item items to be recommended) in item-based CF algorithm;
- Proposing an improved content-based RS, which outperforms the state-of-the-art top-N recommendation methods.

II. PRELIMINARIES

For RSs to work efficiently, the rating (or purchase) histories should be provided as explicit or implicit user-item interactions, which can be modeled as a bipartite network. In this paper, we assume that explicit ratings of users on items (i.e., directed links from users to items in the users-items bipartite network) are available. These ratings are used to compute the item-item similarity scores, which is used in the item-based CF algorithm. Having a set of items $I = \{i_1,i_2,...,i_M\}$ and a set of users $U = \{u_1,u_2,...,u_N\}$, R is an N-by-M adjacency matrix (weighted and directed), where $R_{u,j}$ represents the rating of user u to the item i. In this work, we also use items contextual information, represented as matrix F. Having a set of items I and a set of content features $\{f_1,f_2,...,f_K\}$, F is an M-by-K matrix for which entry $F_{i,f} = 1$ if item i contains feature f, and $F_{i,f} = 0$ otherwise. In our experiments, items are movies and their content features include movie genre(s), actors/actresses, director(s), country and production year. For instance, if actor a has showed up in movie m, then $F_{m,a} = 1$. Matrix F appears differently in hybrid RSs. In some works, it has been used along with SVD to reach better accuracy. For example, Content-Boosted matrix factorization uses such strategies by first finding the relation of users and content features, and then using it to predict the ratings [21].

III. ITEM-BASED COLLABORATIVE FILTERING

In this work we use hybrid recommendation in which the main algorithm is an item-based CF. In an item-based CF scenario, for predicting the rating of user u on item i, first the similarity of i with other items is computed. Often, two types of similarity measures are used for this purpose: Cosine similarity measure and Pearson correlation. Cosine similarity is defined as follows:

$$sim(i,j) = \frac{\sum_{u\in U} R_{u,i}R_{u,j}}{\sqrt{\sum_{u\in U} {R_{u,i}}^2}\sqrt{\sum_{u\in U} {R_{u,j}}^2}}. \tag{1}$$

Pearson correlation is computed like the cosine similarity with the difference that the ratings are normalized by the item's mean rating, as:

$$sim(i,j) = \frac{\sum_{u\in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j} - \bar{R}_j)^2}}, \tag{2}$$

where $\bar{R}_i$ is the mean rating given to item i. Once the similarity values between all items and item i are computed, they are sorted in descending order. Then, the top-k items receiving ratings from user u are selected, and the rating of user u to item i is computed as a weighted mean of these values:

$$R_{u,i} = \frac{\sum_{j\in Top-k(i)} sim(i,j) \times (R_{u,j} - \bar{R}_j)}{\sum_{j\in Top-k(i)} sim(i,j)} + \bar{R}_i. \tag{3}$$

IV. HYBRID RECOMMENDER SYSTEMS BASED ON CONTENT FEATURE RELATIONSHIP

A. Extracting content feature relationship

In this work we introduce a novel approach for extracting the relationship between content features. There are two intuitive justifications for the proposed approach. The first is that although an item's content features are the source of preference for a user towards that item, not all of the content features have the same effect. The second is that content features are semantically related to each other. For instance, if a user has watched movies produced in 1941, 1943 and 1944, he/she will probably like movies produced in 1942. The relationship between content features is represented by a K-by-K real valued matrix C, where K is the number of features. Each entry $C_{f1,f2}$ shows the relationship between features f_1 and f_2. In the following we provide the mathematical formulation. As mentioned, the content features are the source of preference (i.e., ratings) of users towards items. Thus, one can state that the rating similarity between items is proportional to their content similarity:

$$S \propto FF^T, \tag{4}$$

where S is an M-by-M matrix such that $S_{i1,i2}$ is the rating similarity between items i_1 and i_2, and F is the feature matrix. The main problem with this formulation is high levels of sparseness for matrix F. While an item might contain thousands of features, only small portion of them is non-zero for many applications. In a movie recommendation application, which is the case study of this paper, features like *year*, *director* and *country* are sparse, since they contain only one or two non-zero entries in their feature vectors. When only one out of hundred features is non-zero, the probability that the similarity estimation of two items being non-zero is 0.01, which is quite small. Thus, the right hand side of equation (4) leads to a highly sparse matrix, which is an inaccurate estimation for matrix S (this issue is illustrated in figure 3 in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2631138, IEEE Transactions on Industrial Informatics

the next subsection).

Let's rewrite equation (4) by inserting the content feature relationship matrix *C*, where each entry of *C* is the relation of two features, as described. We should insert matrix *C* such that the feature relations are taken into account for calculating the similarity of two items. Thus, it will be placed as a mediator between dot product of two feature vectors:

$$S \propto FCF^T. \tag{5}$$

Equation (5) indicates that the relationship between items is influenced by three factors: the relationship between items and their content features, the relationship between content features and the one between content features and items. It is worth mentioning that in a mathematical formulation, every relationship is represented by a matrix, and a combination of two relationships corresponds the product of these matrices, as also performed in equation (5). Since each entry of the feature relationship matrix *C* is non-zero, the similarity estimation for two items with completely different feature vectors will be non-zero. Consequently, the similarity values in matrix *S* can be accurately estimated.

Equation (5) is a linear equation and minimizing the least squares results in a simple pseudo-inverse solution to obtain the unknown values of *C*. Often, a regularization term is added in such cases to guarantee the inverse of the matrix and avoid overfitting. Substituting the proportionality relationship in equation (5) with equality relationship, and applying the regularization factor with parameter λ, one can obtain the optimal *C* by forcing the gradient to zero, as

$$\begin{aligned}
&\nabla_C \frac{1}{2}[(S - FCF^T)^T(S - FCF^T) + \\
&\qquad \lambda((FC)^T(FC) + (CF^T)^T(CF^T) + \lambda C^T C)] = 0 \\
&\Rightarrow -F^T SF + F^T FCF^T F + \lambda(F^T FC + CF^T F + \lambda C) = 0 \\
&\Rightarrow (F^T F + \lambda I)C(F^T F + \lambda I) = F^T SF \\
&\Rightarrow C = (F^T F + \lambda I)^{-1} F^T SF (F^T F + \lambda I)^{-1}\,,
\end{aligned} \tag{6}$$

where ∇_C is the gradient with respect to C, *I* is the identity matrix and λ is a regularization parameter.

One of the main advantages of this method is that there is no need for extra information about the content features, and they can be heterogeneous. In other words, the system works regardless of whether the features are nominal or numerical. Figure 1 presents a part of matrix *C* related to the feature *year*, in which *C* is the feature relationship matrix for a combination of heterogeneous features. The horizontal axis includes the years from 1922 to 2008 and the vertical axis is the value corresponding to the relationship between features (similarity between the movies produced in two corresponding years). As expected, the relationship between years decreases with the increase of their difference, which means that movies produced in the same time periods have more similarity as compared to those produced in different periods. The diagram also indicates that the users have more similar taste about old movies rather than new ones.

We also investigate matrix *C* for relationship of other features. Table 1 shows the relationships of 7 different genres, which shows a rational relationship between them. For example, *children* and *documentary* genres have negative relationship, whereas *action* and *war* genres are positively related. Table 2 shows the feature relationships for 6 countries, which shows interesting patterns for the movies produced in different countries. For example, the movies produced in Japan have significantly higher self-relationship than those produced in other counties. These observations show that our approach to extract the relationship between content features makes sense. We will next show that this approach improves the performance of CF algorithm. Similar approaches can also be used for other studies such as document analysis and information retrieval, as some previous works have done so [22].

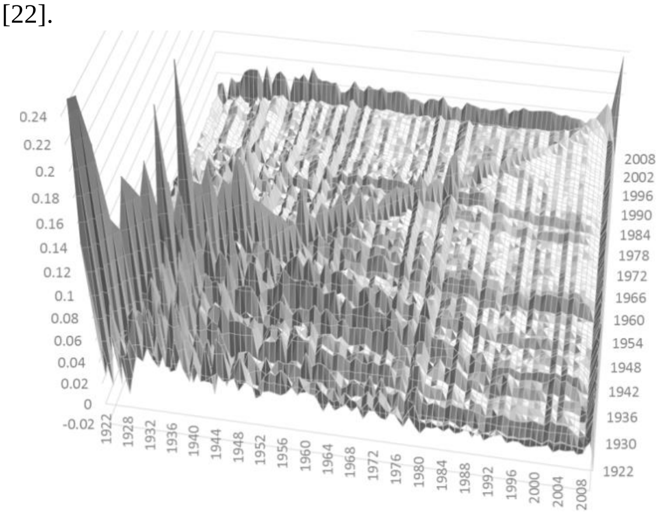

Figure 1: A Sample content feature relationship when *year* is considered as the feature.

TABLE 1: CONTENT FEATURE RELATIONSHIP FOR FEATURE *GENRE*.

	Animation	*Children*	*Comedy*	*Action*	*Documentary*	*War*	*Western*
Animation	0.06	0.04	0.02	0.01	-0.04	0.00	0.00
Children	0.04	0.31	0.01	0.01	-0.09	0.07	0.02
Comedy	0.02	0.01	0.29	-0.03	0.06	-0.04	-0.02
Action	0.01	0.01	-0.03	0.06	-0.06	0.07	0.06
Documentary	-0.04	-0.09	0.06	-0.06	0.72	-0.16	-0.12
War	0.00	0.07	-0.04	0.07	-0.16	0.30	0.16
Western	0.00	0.02	-0.02	0.06	-0.12	0.16	0.51

TABLE 2: CONTENT FEATURE RELATIONSHIP FOR FEATURE *COUNTRY*.

	USA	*France*	*Japan*	*Canada*	*Italy*	*Germany*
USA	0.0307	0.0197	0.0203	0.0261	0.0178	0.0228
France	0.0197	0.0280	0.0194	0.0218	0.0204	0.0189
Japan	0.0203	0.0194	0.0707	0.0220	0.0156	0.0170
Canada	0.0261	0.0218	0.0220	0.0339	0.0188	0.0219
Italy	0.0178	0.0204	0.0156	0.0188	0.0437	0.0183
Germany	0.0228	0.0189	0.0170	0.0219	0.0183	0.0283

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2631138, IEEE Transactions on Industrial Informatics

B. *Dealing with cold-start problem*

The term cold-start is used for predicting the ratings when new items (and/or users) are added to the system [1]; an example is shown in Figure 2. New items may have few ratings (no rating in extreme case), and thus it may not be possible to use CF methods for predicting ratings for them [1, 16]. To deal with this problem, we have to use other information provided about the items such as their contextual information.

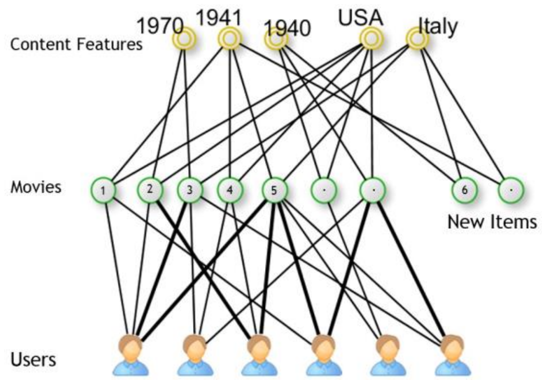

Figure 2: An example of the cold-start problem: the new items have no users' rating history, and thus CF algorithms fail to predict the ratings for them.

Several hybrid methods have been proposed to deal with the cold-start problem. Our work is basically similar to SimComb proposed by Mobasher *et al.* [23] that uses content features to compute the item-item similarities. These similarities are then used either in combination with rating similarities to improve performance of RS, or solely to deal with the cold-start problem. SimComb showed that performing SVD on the content matrix can extract items latent semantics. However, our experiments show that SVD does not reach good results in cases where the content feature matrix is sparse (which is the case for many realistic scenarios). For example, in the movie dataset, if we use only the feature *year* (in its raw format and not aggregated over a decade), the latent semantic cannot be extracted using SVD, because the feature matrix corresponds to a many-to-one relation. Therefore, in such a case the results of SVD in its best are as good as using simple content similarity (e.g., Cosine similarity of feature vectors in Figure 3). For features like *year*, SVD method often fails, since it is not possible to find a relationship between two years by only looking at items content matrix.

In this section we use the relationship matrix introduced in the previous section and in section V we show that one can effectively handle the cold-start situation by employing this relationship in RS algorithms. Let's first discuss how the method works for the case of new items. In the absence of users' rating for new items, RS cannot compute the similarity of a new item with others. Let's suppose that a new item i is involved in the prediction task. In order to compute the similarity of this item with an old item j ($S_{i,j}$), one has to compute

$$S_{i,j} = f_i \times C \times f_j^T, \tag{7}$$

where f_i is the new item's content feature vector and f_j is the content feature vector of an old item j. C is the relationship matrix. When predicting the rating for new item i, having its similarity with all old users, we can find the K-most similar items to i and perform the item-based CF to predict user's ratings on this item. Figure 3 shows an example of how content features can be used for estimating the similarity between a new item and existing ones. On the left hand of this figure the calculated similarity scores are presented. These similarities are computed using dot product of content feature vectors, as introduced in equation (4). Figure 4 illustrates how we can improve the similarity estimation having the relationship between content features. Suppose that we have a prior knowledge about content features relationship such that every feature has a relation of 1 with itself and 0 with the others, except for features 1941 and 1940 which have a relation of 0.5 with each other. Using equation (7) we calculate similarity scores, as shown on the left hand of Figure 4. As you see, these similarities provide more discrimination between items and make similarity estimation more accurate.

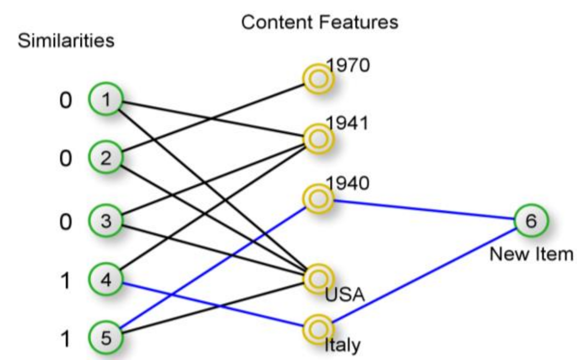

Figure 3: A simple similarity estimation for new items. When a new item is added to the system and it does not contain any user ratings, content feature similarity can be used for estimating its similarity with other items. However, since the content features are highly sparse, these similarities are inaccurate and cannot effectively discriminate items.

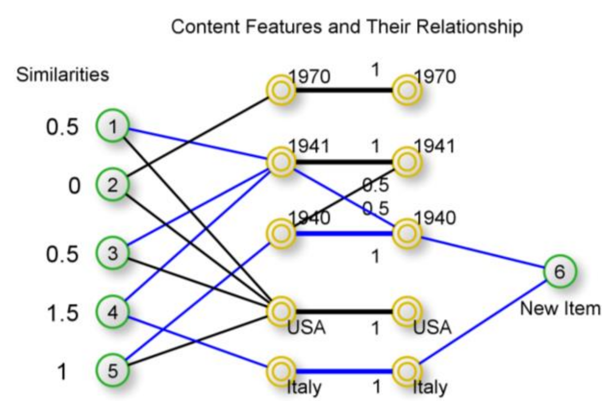

Figure 4: An improved similarity estimation for new items. Embedding content feature relationship matrix in the similarity estimation process enables system to use the underlying semantic relationship between features to improve the performance. For instance, in the example shown in this figure, knowing the relationship between years 1940 and 1941 increases the similarity value of items 1, 3 and 4 leading to a better discrimination between items.

The proposed Feature Relation Improved Similarity Estimation (FRISE) method has some drawbacks when it faces denser feature vectors. The main cause is that this method underestimates the self-relations of the features, which are the diagonal entries of matrix C. To overcome this issue, we add parameter α to the model and calculate C' as:

$C' = C + \alpha I$.

Replacing C with C' in equation (7) we get Extended FRISE (FRISE+) method. Using the above equation for the similarity matrix, one obtains

$$S \propto FC'F^T = F(C + \alpha I)F^T = FCF^T + \alpha FF^T. \quad (8)$$

C. Improving content-based recommendations

In this section we show how to use the content feature relationship matrix to improve content-based RSs. Although content-based RSs are the incipient generation of recommendation lists, they are one of the widely used methods in RSs. The reason for their wide-spread use is that in many cases the rating information is not available (or only some binary users-items interaction data is available) and one has to use items content, either solely or in combination with the interaction data. In some other cases, we may not have access to the whole network. Suppose a situation in which a web user visits different web pages and is not aware of other users. In this case, the only possible recommendation scenario is to use a content-based algorithm. In content-based RSs, first we create a Term Frequency-Inverse Document Frequency (TF-IDF) profile for each user based on the content of the items he/she has already visited. Then, the preference of the user to an arbitrary item is defined as a similarity between his/her profile and the item's content. This similarity can be computed as a simple inner product, and thus content-based methods are fast and scalable for online recommendation.

In this work, we apply a modification to the well-known content-based model using the proposed concept of content feature relationship. Having the content feature relationship matrix, we can embed this relationship in computing the similarity of a user's profile with an arbitrary item (which can be viewed as the preference of the user toward the item). Let's denote the user's TF-IDF profile by vector p_u, the item's content feature vector by f_i and the content feature relationship matrix by C. We define the preference of user u to item i as

$$pref_{u,i} = cosineSimilarity(p_u, Cf_i). \quad (9)$$

In this equation C can be considered as a projection matrix, which projects the sparse feature vector f_i to a denser feature space. As a result, the similarity between the users' profile and this projected feature vector would be computed more precisely that makes the preference estimation more accurate. Using these calculated preference scores we can sort the items for a target user and provide the top-N recommendation (i.e., choosing N items from the top of the ranking vector). The relationship matrix helps the system to have higher precision in the recommendation task, which is mainly because it takes into account the importance and relationship of the features.

D. Computational complexity

RSs usually contain both offline and online processes. Model-based methods such as SVD contain a heavy offline process while K-nearest neighbor needs more online computation. Our method contains both online and offline computation. Content feature relationship matrix is computed offline and the prediction process is performed in an online manner. The online part of the algorithm is simple and similar to the original methods (i.e., item-based CF and content-based RS). Therefore, the main computationally expensive part is the offline part. As it can be seen from equation (6), finding this relationship needs inversing the matrix F^TF, which can be computed in $O(K^3)$. The matrix multiplications can be performed in $O(K^2M+M^2K)$, and thus the overall time complexity of the algorithm is $O(K^3+K^2M+KM^2)$. However, this is an upper bound and more efficient matrix inversion and multiplication methods can be used to reduce the complexity. In our experiments it takes less than 10 minutes for 10K features on a Core-i5 machine.

V. EXPERIMENTS

A. Dataset and content features

We perform our experiments on hetrec-movielens-2k-v2 dataset [24, 25]. This dataset contains 855,598 ratings of 2,113 different users over 10,197 movies. Table 3 represents the feature sets used in our experiments. We filter features by their minimum occurrence in the items' feature vectors.

TABLE 3: FEATURE SETS ACCORDING TO THEIR MINIMUM APPEARANCE IN THE DATASET.

	Minimum Occurrence	Actor	Director	Country	Year	Genre	All
F0	0	95321	4060	72	98	20	99571
F5	5	10673	539	39	87	19	11357
F10	10	3645	162	34	80	19	3940
F20	20	952	20	22	73	19	1086

B. Rate prediction for cold-start items

In Figure 5 we compare Root Mean Square Error (RMSE) of the prediction task for four different methods and six different content feature sets. These methods first calculate the item-item similarities in the cold-start situation using different formulations, and then perform item-based CF. The first method – called Content-Based Similarity Estimation (CBSE) – uses a straightforward content-based similarity introduced in equation (4) to compute the estimated similarities between the items. In this method, the similarity of items i and j by content feature vectors f_i and f_j is computed as follows:

$$sim(i,j) = f_i {f_j}^T. \quad (10)$$

In the second method – called Dimensionality Reduced Feature Similarity Estimation (DRFSE) – we compute the similarities using a dimensionality reduced feature vector introduced in SimComb [23]. This method computes the similarity matrix using the following equation:

$$S = UU^T, \quad (11)$$

where the content feature matrix F is decomposed as F =

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2631138, IEEE Transactions on Industrial Informatics

$U\Sigma V$.

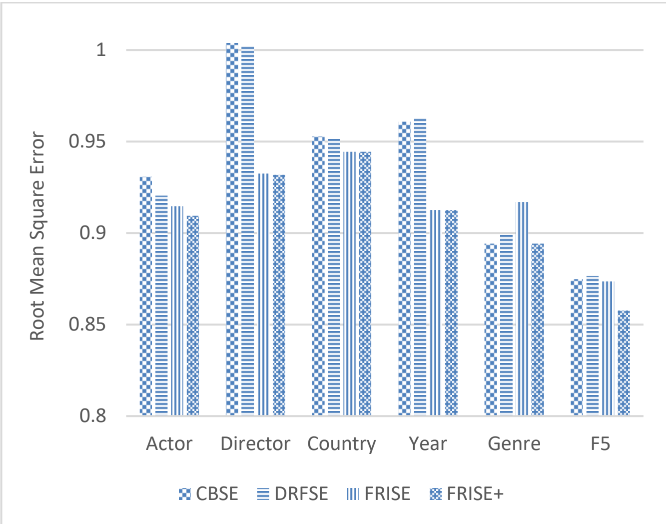

Figure 5: Root Mean Square Error (RMSE) in the cold-start problem (new items) using four different methods and various content features including actors, director(s), country, production year, movie genre and all features (F5). The four similarity estimation methods include content-based similarity estimation (CBSE) which represents the benchmark, dimensionality reduced similarity estimation (DRFSE) which is the state-of-the-art method, feature relation improved similarity estimation (FRISE) and extended feature relation improved similarity estimation (FRISE+) which are our proposed methods.

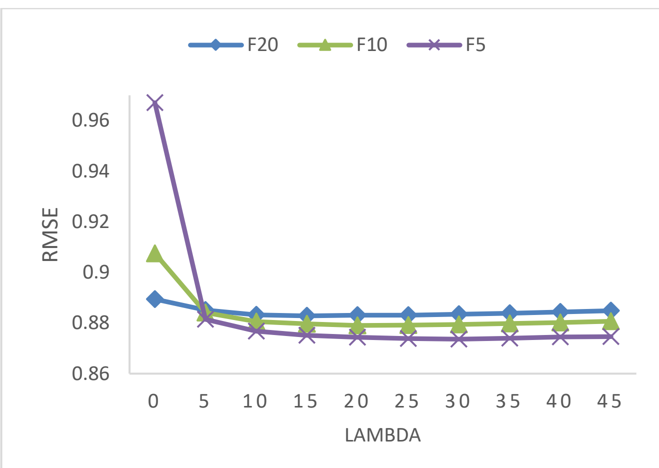

Figure 6: The effect of parameter λ on RMSE in feature relation improved similarity estimation (FRISE) algorithm. As the number of features increases, the influence of λ increases.

In the third method – which is our proposed method, called Feature Relation Improved Similarity Estimation (FRISE) – the content feature relationship is used for computing the content-based item-item similarity matrix, as expressed in equation (5). The similarity of items i and j, when the corresponding content feature vectors are f_i and f_j, is computed using equation (7). The fourth method – Extended Feature Relation Improved Similarity Estimation (FRISE+) – uses equation (8) for the similarity estimation.

For this experiment, the content feature relationship matrix C is computed independently for different feature sets, and the reported results are categorized by these feature sets. It is seen that for sparse item feature types, especially *year* and *director* features, our method significantly improves the prediction error. As described in section IV, FRISE did not work well for *Genre* feature which is a dense feature vector. This is mainly because our model underestimates the self-relation of the features. As it is seen in figure 5, the extended method (FRISE+) outperforms all others.

Figures 6 and 7 show the empirical study of the effect of parameters λ and α on RMSE. As it is seen in figure 6, λ gets more influence as the number of features increases. The reason is that over-fitting usually happens for high dimension feature spaces. When $\alpha = 0$, FRISE+ is equivalent to FRISE and for $\alpha = +\infty$, FRISE+ works the same as CBSE (Figure 7).

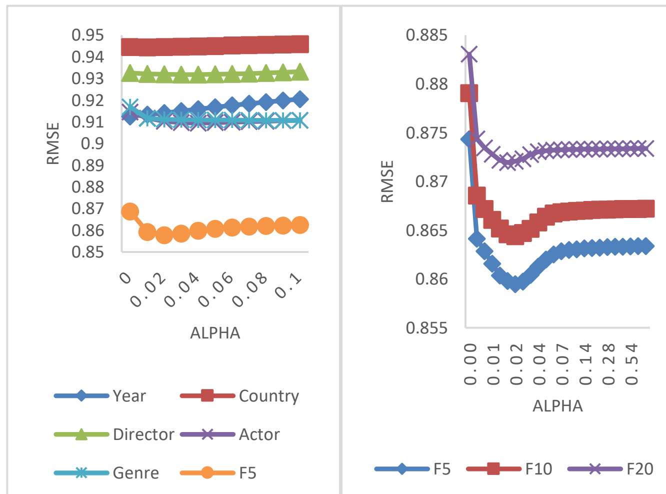

Figure 7: The effect of parameter α on RMSE in FRISE+ model. FRISE underestimates the self-relation of a feature for dense features. FRISE+ combines it with content-based similarity estimation (CBSE) and achieves a better prediction error.

C. Top-N recommendation

Figure 8 shows a comparison of accuracy measures for the proposed content-based method and a number of state-of-the-art RS methods including Rate Normalized SVD (NSVD50), Correlation-based item-item CF (CorKNN), Content-based (CB), Non-normalized Cosine-based item-item CF (NNCosKNN), Content-Boosted CF (CBCF), Non-normalized SVD (PureSVD50) and the proposed Improved Content-based RS (ICB) for two feature sets F5 and F10. In order to obtain these results, first the preferences (or rating values) are predicted, and then the top-*N* recommendation is constructed for each user. Let's define the accuracy metrics used in these experiments. As mentioned, the system provides a set of items (e.g., 10 items) as the recommended list for each user. This list contains both relevant and irrelevant items and the system tries to include more relevant items in the list. Having the confusion matrix shown in Table 4, Precision, Recall and F1 metrics are defined as follows:

$$Precision = \frac{N_{rs}}{N_{rs} + N_{is}}, \tag{11}$$

$$Recall = \frac{N_{rs}}{N_{rs} + N_{rn}}, \tag{12}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2631138, IEEE Transactions on Industrial Informatics

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (13)$$

These metrics measure the quality of recommendations in terms of the number of hits in the recommendation list. The F1 measure is proposed to consider a combination of precision and recall so that the effect of recommendation list size on the balance of these two metrics is removed.

TABLE 4: CONFUSION MATRIX.

	Recommended	Not Recommended
Relevant	N_r	N_{rn}
Irrelevant	N_{is}	N_{in}

As Figure 8 shows, the proposed improved CB recommender using feature set F5 is the top-performer in hit ratio. We also compare the algorithms in terms of novelty and diversity of the recommendation lists. In many practical RSs, users would like to be recommended items that not only they likely provide high ratings, but also are novel and diverse [6, 26-30]. Therefore, in recent years the research community has put significant efforts in designing RS with high levels of novelty and diversity [5, 6, 26-28, 30, 31]. Coverage index indicates how much the items are recommended fairly and with similar probabilities (equation (14)). Intra-diversity measures the diversity of items recommended for a user. According to equation (15), the more different are the items in a list, the higher the intra-diversity. On the other hand, inter-diversity guarantees that the recommended lists are finely personalized and different users get different lists. These metrics are computed as follows:

$$Coverage = -\sum_{k=1}^{|I|} P(i_k) log P(i_k), \quad (14)$$

$$Intra - Diversity = \frac{\sum_{k=1}^{N}\sum_{l=1}^{N}\left(1 - sim\left(i_{k,}\, i_l\right)\right)}{N^2}, \quad (15)$$

$$Inter - Diversity = \frac{\sum_{k=1}^{|U|}\sum_{l=1}^{|U|}\left(1 - \frac{List(u_k) \cap List(u_l)}{N}\right)}{|U|^2}. \quad (16)$$

Novelty and accuracy do not often go hand in hand. In other words, as novelty increases, accuracy often decreases and vice versa. To deal with the accuracy-novelty dilemma, we use effective novelty measure [5] which sums up novelty over items that are hit, as follows.

$$Effective\ Novelty = {-\sum_{u=1}^{|U|}\sum_{i\epsilon Hit(u)} logP(i)}\Big/{|U|}. \quad (17)$$

Effective novelty is a combination of both novelty and precision metrics and is the metric to judge algorithms. The mean effective novelty metric calculates the average novelty of hit items.

Figure 9 compares the algorithms in terms of coverage, diversity and novelty metrics (all of the metrics are normalized by their max value). The proposed method not only shows higher accuracy than other algorithms, but also its effective novelty is the highest. Furthermore, the proposed method has comparable inter- and intra-diversity with other methods. Indeed, the improved CB recommender considerably enhances the accuracy of the recommendations, without significantly affecting the diversity of the recommendation lists.

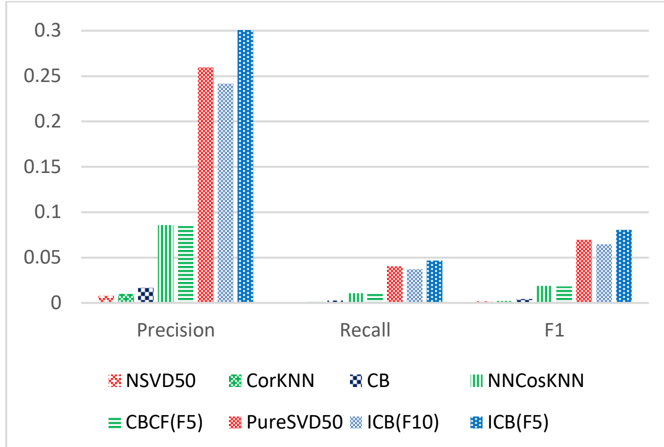

Figure 8: Accuracy metrics (Precision, Recall and F1) for eight RS algorithms including Rate Normalized SVD (NSVD50), Correlation-based Item-Item RS (CorKNN), Content-based RS (CB), Non-normalized Cosine-based Item-Item RS (NNCosKNN), Content-boosted CF (CBCF), Non-normalized SVD (PureSVD50) and the proposed Improved Content-based RS (ICB) for two feature sets F5 and F10.

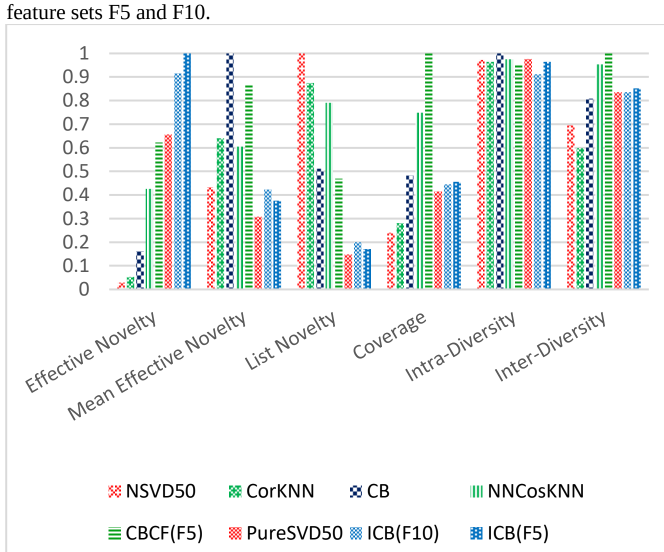

Figure 9: Novelty and diversity metrics including Effective Novelty, Mean Effective Novelty, Recommendation List Novelty, Coverage, Intra-Diversity and Inter-Diversity for six recommender algorithms.

The hybrid RSs are often proposed to deal with the problems of CF and content-based algorithms. However, here the proposed ICB algorithm focuses on improving the general task of RSs (i.e., top-*n* recommendation). To compare its performance with state-of-the-art hybrid algorithms, we implement hybrid content-boosted CF (CBCF) [17] and apply it to the same recommendation task. The implemented CBCF is an item-item CF that its similarities are calculated on a rating matrix enriched by a content-based algorithm. As shown in figures 8 and 9, incorporating content information in CBCF does not improve the hit ratio. It just improves some novelty-diversity metrics in comparison with NNCosKNN algorithm. The key point about the high performance of proposed ICB algorithm is that it chooses top-*n* items similar to the procedure that a user would naturally handpick his/her purchase list. In the application to movie recommendations (the task considered in this paper), the users separate their purchase list according to some key features of the movies, e.g., their directors and/or superstars. Embedding content relationship matrix simulates such a process.

The results also indicate that removing item biases via rate normalization (e.g., NSVD50 and CorKNN methods) leads to low precision, because the items popularity is not considered in this way. Therefore, despite the observation that rate normalization improves prediction error, it is not helpful for the top-*N* recommendation task.

VI. CONCLUSIONS

In this work, we introduced the content feature relationship concept and proposed an innovative method for extracting such relationship. This relationship was embedded into the algorithms of two well-known recommendation methods. We first modified item-based collaborative filtering such that it could better deal with cold-start problem. Then, we introduced a hybrid content-based method which has the privileges of a content-based method. The proposed methods are based on content features relationship, which is estimated from the data. We used a 2D matrix to model content features relationship. To train the content features relationship matrix from the rating data, we used the least squares method; however other models and learning methods can be applied to extract and use this kind of relationship. Our experiments showed superior performance of the proposed algorithms over a number of state-of-the-art methods including the content-based recommender, item-based collaborative filtering, SVD, and the hybrid SimComb method. The proposed hybrid content-based algorithm showed the highest performance in terms of accuracy and effective novelty.

REFERENCES

[1] G. Adomavicius and 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,* vol. 17, pp. 734-749, 2005.

[2] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," presented at the National Conference on Artificial Intelligence, 1998.

[3] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM,* vol. 40, pp. 56-58, 1997.

[4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," presented at the 10th international conference on World Wide Web, 2001.

[5] A. Javari and M. Jalili, "Accurate and novel recommendations: an algorithm based on popularity forecasting," *ACM Transactions on Intelligent Systems and Technology,* vol. 5, p. 56, 2014.

[6] A. Javari and M. Jalili, "A probabilistic model to resolve diversity-accuracy challenge of recommendation systems," *Knowledge and Information Systems,* vol. 44, pp. 609-627, 2015.

[7] J. L. de la Rosa, N. Hormazabal, G. A. Lopardo, A. Trias, and M. Montaner, "A negotiation-style recommender based on computational ecology in open negotiation environments," *IEEE Transactions on Industrial Electronics,* vol. 58, 2011.

[8] Y. Li, B. Cao, L. Xu, J. Yin, S. Deng, Y. Yin*, et al.*, "An efficient recommendation method for improving business process modeling," *IEEE Transactions on Industrial Informatics,* vol. 10, pp. 502-513, 2014.

[9] A. Gunawardana and C. Meek, "A unified approach to building hybrid recommender systems.," presented at the ACM Recommender Systems, 2009.

[10] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adaptive Interaction,* vol. 12, pp. 331-370, 2002.

[11] H. Zhang, Y. Ji, J. Li, and Y. Ye, "A triple wing harmonium model for movie recommendation," *IEEE Transactions on Industrial Informatics,* vol. 12, pp. 231-239, 2016.

[12] A. Javari, J. Gharibshah, and M. Jalili, "Recommender systems based on collaborative filtering and resource allocation," *Social Network Analysis and Mining,* vol. 4, p. 234, 2014.

[13] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering in usenet news," *Communications of the ACM,* vol. 40, pp. 77-87, 1997.

[14] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," *Engineering Applications of Artificial Intelligence,* vol. 46, pp. 58-66, 2015.

[15] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Transactions on Industrial Informatics,* vol. 10, pp. 1273-1284, 2014.

[16] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," presented at the ACM SIGIR Conference on Research and Development in Information Retrieval, 2002.

[17] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," presented at the AAAI/IAAI, 2002.

[18] P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor, "HyPER: A flexible and extensible probabilistic framework for hybrid recommender systems," presented at the ACM Recommender Systems, 2015.

[19] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," presented at the ACM Conference on Recommender Systems, 2010.

[20] X. Ning and G. Karypis, "Sparse linear methods with side information for top-n recommendations," presented at the ACM Recommender Systems, 2012.

[21] P. Forbes and M. Zhu, "Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation," presented at the ACM Recommender Systems, 2011.

[22] H. Zhang, J. K. L. Ho, Q. M. J. Wu, and Y. Ye, "Multidimensional latent semantic analysis using term spatial information," *IEEE Transactions on Cybernetics,* vol. 43, pp. 1625-1640, 2013.

[23] B. Mobasher, X. Jin, and Y. Zhou, "Semantically enhanced collaborative filtering on the web," in *Web Mining: From Web to Semantic Web*, ed: Springer, 2004, pp. 57-76.

[24] I. Cantador, P. Brusilovsky, and T. Kuflik, "Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011)," presented at the ACM Recommender Systems, 2011.

[25] J. Riedl and J. Konstan, "Movielens dataset," ed, 1998.

[26] G. Adomavicius and Y. Kwon, "Improving aggregate recommendation diversity using ranking-based techniques," *IEEE Transactions on Knowledge and Data Engineering,* vol. 24, pp. 896-911, 2012.

[27] C. L. A. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, *et al.*, "Novelty and diversity in information retrieval evaluation," in *SIG Information Retrieval*, Singapore, 2008, pp. 659-666.

[28] A. Javari, M. Izadi, and M. Jalili, "Recommender systems for social networks analysis and mining: precision versus diversity," in *Complex Systems and Networks*, J. Lü, X. Yu, G. Chen, and W. Yu, Eds., ed: Springer, 2016, pp. 423-438.

[29] N. Lathia, S. Hailes, L. Capra, and X. Amatriain, "Temporal diversity in recommender systems," in *SIG Information Retrieval*, Geneva, Switzerland, 2010, pp. 210-217.

[30] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Science of the United States of America,* vol. 107, pp. 4511-4515, 2010.

[31] P. Castells, J. Wang, R. Lara, and D. Zhang, "Introduction to the special issue on diversity and discovery in recommender systems," *ACM Transactions on Intelligent Systems and Technology,* vol. 5, p. 52, 2015.