



Personalized Case-Based Explanation of Matrix Factorization Recommendations

Jose Jorro-Aragoneses^(✉), Marta Caro-Martinez^(✉),
Juan Antonio Recio-Garcia^(✉), Belen Diaz-Agudo^(✉),
and Guillermo Jimenez-Diaz^(✉)

Department of Software Engineering and Artificial Intelligence,
Universidad Complutense de Madrid, Madrid, Spain
{jljorro,martcaro,jareciog,belend,guille}@ucm.es
<http://gaia.fdi.ucm.es>

Abstract. Matrix factorization is an advanced recommendation strategy based on characterizing both items and users on a vector of latent factors inferred from rating patterns. These vectors represent, somehow, a characterization of the user preferences in a lower dimensionality space. Although matrix factorization is more accurate than other recommendation strategies, the main problem associated with this approach is that the discovered factors are opaque and difficult to explain to the final user. In this paper we propose a personalized case-based explanation strategy that uses the latent factors to find similar explanatory cases already rated by the user.

Keywords: Case-based explanation · Personalised explanation · Matrix factorization

1 Introduction

Recommender systems are typically based on one of two strategies. The *content filtering* approach creates a profile for each user or product to characterize its contents and recommends a similar product that matches the user profile. For example, a movie profile could include attributes regarding its genre, year, director, actors, and so forth. An alternative approach is *collaborative filtering* that is more flexible and generally more accurate than content-based techniques. Collaborative filtering relies only on user ratings and analyzes relationships between users and items, or between items to identify new user-item associations [1]. Recommendations resulting from content-based strategies are more comprehensible for users, as they are based on the explicit user preferences.

Since its success during the Netflix prize challenge the *matrix factorization* algorithm [2] has become one of the most successful algorithms to generate personalized recommendations. Matrix factorization is an advanced strategy that

Supported by the UCM (Research Group 921330), the Spanish Committee of Economy and Competitiveness (TIN2017-87330-R) and the fundings provided by Banco Santander in UCM (CT17/17-CT17/18) and (CT42/18-CT43/18).

Table 1. Recommended movie using matrix factorization

Movie Id.	Movie title	Year	Director	Stars	Predicted
223	Clerks	1994	Kevin Smith	Jason Mewes Jeff Anderson	4.042

attempts to merge the content and collaborative information in a single model based on characterizing both items and users on a vector of factors inferred from the ratings patterns. Although these vectors represent, somehow, a characterization of the user preferences, they are opaque collections of numeric values computed by the algorithm. In this paper we propose using these vectors to define a personalized similarity metric between items for every user. Case-based explanations focus primarily on finding explanatory cases that are *similar* to the recommended item [20]. Then, we use these cases to interpret the opaque output of the matrix factorization recommendation algorithm.

From the point of view of recommender systems, we propose an item-based explanation, since it uses items to justify a recommendation [16]. The main advantage of this approach is that it allows users to assess the quality of the recommendation by comparing items, that ideally should be *similar* according to the user’s criteria. The main challenge of these case-based explanation strategies is to find a similarity metric that matches the user’s criteria. Current content-based approaches [13] are based on the comparison of item’s features, leaving aside the user’s interpretation of these features. Therefore, in this paper we use the vectors of factors that characterize the user preferences to compute a similarity metric that finds related items in order to explain the recommendation.

Let’s motivate our approach with an example. Given a user that has rated several movies in a dataset, the matrix factorization algorithm recommends “Clerks”. Table 1 shows its features and Table 2 shows the most similar rated movies using as similarity metric the cosine of the vectors of factors extracted from the matrix factorization.

Here, “The usual suspects” is the most similar but there is not a clear intuition about the reasons for this similarity from the point of view of the canonical content-based distance. According to that distance, “The usual suspect” won’t be chosen as an item for comparison as there are no common features between both movies (leaving aside the year). However, our hypothesis is that the vector of factors resulting from the matrix factorization is able to capture relations that make sense from the user’s point of view. For example, the user may like politically incorrect movies and the matrix factorization has captured that factor, and therefore making both movies similar.

Section 2 reviews the related work in explanations in recommender systems. Section 3 explains the matrix factorization method. Section 4 describes how to define a personalized similarity metric between items for every user that is used to retrieve the explanatory cases. Section 5 evaluates the similarity metric associated to our case-based explanation model demonstrating how to get relevant explanatory cases without additional knowledge on the item features. Section 6 concludes the paper and describes the ongoing lines of work.

Table 2. Most similar movies according to the vectors of factors resulting from the matrix factorization

Movie Id.	Movie title	Year	Director	Stars	Rating	Similarity
50	The Usual Suspects	1995	Bryan Singer	Kevin Space Kevin Pollak	5.0	0.796
163	Desperado	1995	Robert Rodriguez	Quentin Tarantino Salma Hayek	5.0	0.750
596	Pinocchio	1940	Norman Ferguson	Mel Blanc Cliff Edwards	5.0	0.665
151	Rob Roy	1995	Michael Caton-Jones	Liam Nesson Eric Stoltz	5.0	0.646
101	Bottle Rocket	1996	Wes Anderson	Andrew Wilson Lumi Cavazos	5.0	0.465

2 Related Work

Using explanations in recommendation systems is an important area of research in this type of systems. One of the main problems with recommendation systems is that users do not know why a product has been recommended to them. Recommender systems that use explanations improve user confidence in those recommendations [20]. In addition, users consume more products resulting from an explainable recommendation process [7].

Nowadays there are many works that apply explanations in recommender systems. In a previous work [3], we carried out an in-depth study of the explanation systems applied to recommendation systems. As a result of this study, we developed a theoretical model to classify the explanation systems according to their characteristics. According to this model, explanation systems employ different methods to obtain the knowledge needed to generate explanations. The model we present in this paper is knowledge-light and the only knowledge container employed is the algorithm, and more precisely, the similarity between items and the user's experiences.

In [12] we find explanation system for movie recommendation systems based on the similarity between plots. Movie similarity is based on the characteristics that are in common between the characters and the interactions of the characters in the plot. The IMVEX system [5] is a rule-based system that personalizes the explanations for different types of users. The knowledge base used is the user profile. The system developed by [11] shows an explanation system for a recommendation system for groups, based on the similarity of preferences among the members of the group. In [17], we found a system that displays the recommendations along with the characteristics that have been involved in the selection of the best candidates for the recommendation. Another example of a

system that takes into account similarities between user preferences and item characteristics is the framework presented in [23].

We are particularly interested in experience-based explanations, which use the past actions of the user and her history of interactions as a source of knowledge to generate explanations. CBR-based explanations are an example of experience-based explanations. There are different works based on CBR. The work in [4] reviews classic systems that use CBR as a way to find similar cases that are used as an explanation of recommendations. In [19], the attribute with the highest weight in the similarity metric is selected in order to find the similar cases that may be of interest to the user as an explanation of the recommendation. In [8] we found a case-based system to explain the detection of healthcare-associated infections. The work in [15] describes a case-based recommender system for hotels, where cases are obtained from users' reviews. The explanations of the recommendations are based on features obtained from this information. The PSIE (Personalized Social Individual Explanation) approach [18] includes explanations to group recommender systems and social explanations with the aim of inducing a positive reaction to users in order to improve their perception of the recommendations. In [14] we found a CBR system that uses the difference between the query and the case descriptions to explain all recommendations.

Finally, there are some works to explain recommendations provided by systems based on latent factors. This is due to the fact that these systems work very well, but they are difficult to explain. In [10] the authors describe the TriRank system, which extracts information from the reviews to improve the transparency of the recommender system. Another work that tries to explain the recommendations obtained from matrix factorization is [24]. The explanation model consists of determining which movies have influenced the rating predicted by the matrix factorization algorithm. In [21], authors propose a method called Tree-enhanced Embedding Method (TEM) that uses embedding-based and tree-based models to extract explanations of recommenders systems based on collaborative filtering and latent factors.

In the following section we explain how a recommendation system based on matrix factorization works. In addition, we explain what information we will be able to use from this algorithm to generate the explanations.

3 Recommendation Using Matrix Factorization

Matrix factorization is one of the most commonly used methods for creating a latent factor model applied to recommendation systems. To create the model, the algorithm uses a $R \in \mathbb{R}^{U \times I}$ matrix that contains the ratings that users (U) have made on a set of items (I). The main problem with the R matrix is that it is very sparse, that is, it only contains a small part of the ratings. The goal of matrix factorization is to complete the R matrix by relating users to items through latent factors of N dimensionality.

To do this, we apply the Simons Funk’s model [6]. We define $P \in \mathbb{R}^{U \times N}$ matrix, which relates each user from U to the factor dimensions (N), and $Q \in \mathbb{R}^{I \times N}$ matrix, which relates the set of items I to each factor dimension (N). This way, a user $u \in U$ is associated with a vector $p_u \in P$ that measures the preferences of the user on items according to the corresponding latent factors. On the other hand, an item $i \in I$ is associated with a vector $q_i \in Q$ that measures how the item is reflected according to the latent factors. The dot product of both vectors will give us the user’s u rating prediction (r'_{ui}) of item i , as illustrated in Fig. 1:

$$r'_{ui} = p_u q_i^T \tag{1}$$

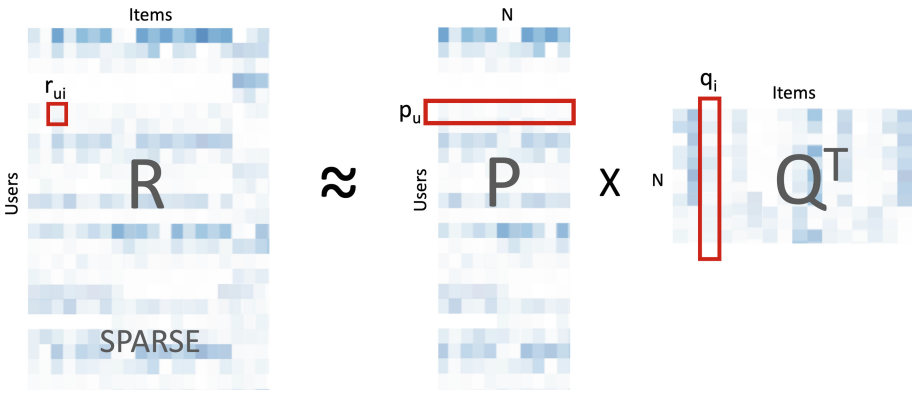


Fig. 1. Matrix factorization general schema.

A recommender system uses a $R^{U \times I}$ matrix, which contains the estimations for each user and each item. This matrix is the result of multiplying P and Q^T matrices.

$$R' = PQ^T \tag{2}$$

From this matrix we will obtain the items that will be recommended to a specific user. To learn the values of P and Q the system minimizes the error between the rating prediction and the known ratings. In our learning process we use the stochastic gradient descent method. In this process, the algorithm runs through the known rating set ($r_{ui} \in R$). For each rating, the system computes the error between this rating and its prediction.

$$e_{ui} = r_{ui} - p_u q_i^T \tag{3}$$

Once the error is known, the values of p_i and q_u are modified by a magnitude proportional to γ in the opposite direction of the gradient. The new values will be:

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \quad (4)$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \quad (5)$$

Once we have described the general schema of the matrix factorization recommendation technique, following sections will depict our proposal for using the Q matrix to find explanatory cases, because this matrix captures user preferences through the factor vectors.

4 Retrieval of Explanatory Cases Using the Q Space

Case-based explanation requires a set of similar items that will be presented as explanatory examples. These items must be similar to the item recommended by the system according to the user preferences. As we described in the previous section, P matrix describes users as factor vectors, meanwhile, Q matrix contains factor vector representations for every item, both of them using a N dimensional space. The dot product of user and item vectors, $p_u q_i^T$ computes the estimated rating for a user u and item i . This way, p_u contains the description of the user, and q_i a general description of the item according to the preferences of all the users in the dataset. As the goal of the explanation process is to obtain explanation items in a personalized way for each user, we need initially to transform the Q matrix to represent the items according to the concrete user u . To do so, we transform the Q matrix into a collection of vectors where each N -dimensional vector represents the description of an item q_i multiplied by the user preferences p_u :

$$Q^u = \{q_1^u, \dots, q_M^u\} \quad (6)$$

where $q_i^u = p_u q_i$

Here, $q_i \in \mathbb{R}^N$ and $M = |I|$ is the number of items in the dataset. This collection of vectors summarizes the user u preferences, where several factor vectors are more discriminant than others. The example in Fig. 2 shows that the vectors represented in columns 1, 12 and 14 are the most discriminant in order to compute the predicted rating of an item. It is important to note that these vectors are personalised for every user as it is the result of multiplying Q by p_u . Therefore, the Q^u matrix is completely different for every user.

This fact is illustrated by Fig. 3, which shows the factor value distribution of the Q^u vectors for two different users given the same set of movies. We can clearly observe that the characterization of both users is different, allowing us to use Q^u as a description of the user's profile. However, the characteristics of the matrix factorization algorithm does not provide a symbolic description of

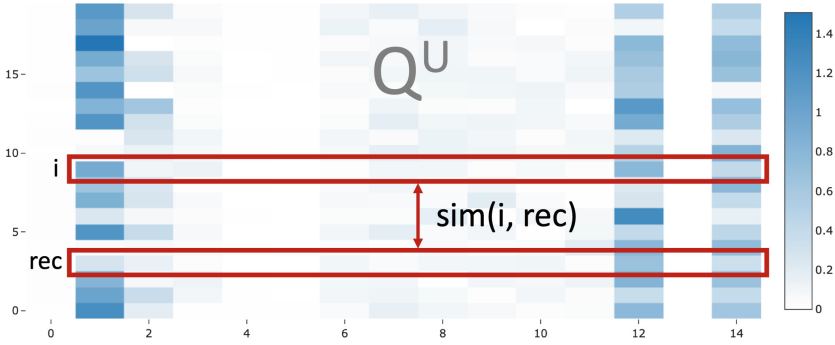


Fig. 2. Visualization of the Q^u matrix capturing the user preferences in a collection of M vectors (number of items) with dimension N (in this example $M = 20 \times N = 14$).

these factors that could be used to explain the results. We cannot even have an intuition of what these vectors exactly mean for each user, as they are numeric values computed by the algorithm. But we can exploit this Q^u matrix to define a personalized similarity metric between items for every user.

Matrix Q^u describes the items according to the user rating patterns. But, to generate the explanations using a case-based approach, the system will only use the items that the user has previously rated. That is, the system filters the items that the user has not rated yet from the Q^u matrix. The result is a new matrix $Q^{u'}$:

$$Q^{u'} = \{q_i^u \in Q^u : r_{ui} \neq \emptyset\} \tag{7}$$

Now, we can define a similarity metric over this space to calculate the similarity between two items according to the user’s perception. We propose using the cosine similarity function to compare q_i^u vectors of each item. The benefit of using this similarity function is that it does not take into account vector magnitudes, which allows item comparison without having to obtain a prior knowledge about the latent factors for each user:

$$sim^{Q^u}(i, rec) = \cos(q_i^u, q_{rec}^u) = \frac{q_i^u \cdot q_{rec}^u}{|q_i^u| \cdot |q_{rec}^u|} \tag{8}$$

Once the similarity metric is defined over the \mathbb{R}^N vector factors space, the set of explanatory cases is obtained by selecting the most similar rated items. The explanatory case set (*Exp*) includes the k items of $Q^{u'}$ that are more similar to the recommended item (*rec*) as described in Algorithm 1.

5 Evaluation

We have described a case-based explanation model where the explanatory examples are retrieved from the Q^u matrix that captures the user preferences in a

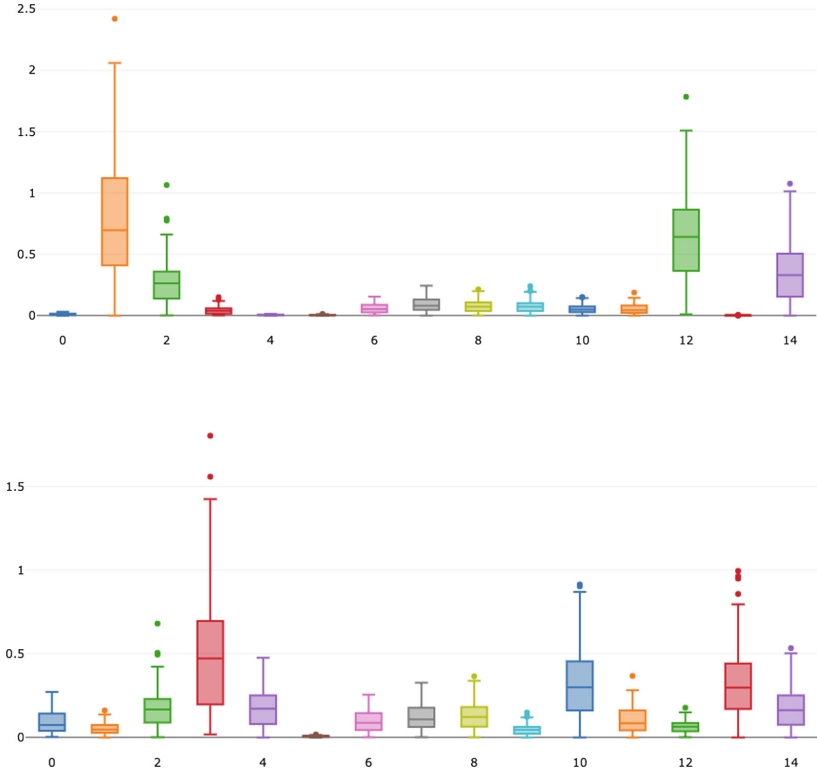


Fig. 3. Factor value distribution of the Q^u vectors that summarizes two users’ profiles (preferences) with dimension $N = 14$. Top figure corresponds to the matrix shown in Fig. 2.

N-dimensional space. To evaluate our model we will prove that the explanatory examples retrieved using sim^{Q^u} are more relevant to the user than the items that we would retrieve using sim^I , that is, a content-based approach that can compute the similarity for every pair of items. A benefit of our approach is that it is knowledge-light, in opposition to the classical content-based approach using I .

Our experiments demonstrate that our model provides personalized results without requiring any knowledge about the items’ description. It overcomes one of the main problems associated with content-based approaches, namely that they require gathering external information that might not be available. Our model does not need the I description matrix, but only the R matrix that includes the users’ ratings.

Algorithm 1. Computation of the explanatory cases set

Input: R, P, Q, u, rec, k**Output:** *Exp*

```

1  $p_u \leftarrow P[u]$ 
2  $r_{ui} \leftarrow R[u, i]$ 
3  $Q^u \leftarrow \{p_u q_1, \dots, p_u q_m : q_i \in Q\}$ 
4  $Q^{u'} \leftarrow \{q_i^u \in Q^u : r_{ui} \neq \emptyset\}$ 
5  $Exp \leftarrow \{\}$ 
6 while  $k > 0$  do
7    $Exp \leftarrow Exp \cup \{i : \operatorname{argmax}_{q_i^u \in Q^{u'}} \operatorname{sim}(q_i^u, q_{rec}^u)\}$ 
8    $Q^{u'} \leftarrow Q^{u'} \setminus q_i^u$ 
9    $k \leftarrow k - 1$ 
10 end
11 return Exp

```

5.1 Datasets

To test our hypothesis we have used the popular movie domain. In this evaluation we used two public datasets. The first one is the 100k MovieLens dataset [9], which contains 100,000 ratings made by users in the MovieLens recommendation system. This dataset will be used by the matrix factorization algorithm. The second dataset contains the features of 5,000 movies [22]. These descriptions have been extracted from IMDB¹. More concretely, the movie features that we used in the evaluation are: genres, directors, actors, screenwriters and the decade in which the movies were released. This second dataset let us to compare the quality of our examples compared to a classical content-based approach.

In the evaluation we selected the movies that both datasets have in common. The final dataset used for the evaluation contains 11,477 ratings made by 587 users on 164 movies. 90% of the dataset has been used to train the P and Q matrices of the recommender system. Regarding the sparsity of the training matrix, it represents the 11% of the complete matrix. The remaining 10% of the dataset has been used to perform the evaluation. Moreover, in order to perform a stratified evaluation according to the rating values, we have created another dataset where each fold has the same rating value. To create the stratified dataset, we have selected 34 items for each rating value². We have made this selection randomly, and we have repeated it 100 times. Then, we have got 3400 items for each rating value. This second evaluation set will verify that the system works better by eliminating the bias of the most popular ratings.

¹ <https://www.imdb.com/>.

² This is the highest possible value as the dataset only contains 34 items rated with 2.5 as shown in Table 5.

5.2 Methodology

As we have explained before, in this evaluation we try to demonstrate that the items we recover using the sim^{Q^u} metric are more relevant as a personalized explanation of a recommendation. To prove it, we have to compare the retrieved examples to those cases we would retrieve using a classical content-based approach.

In order to define a content-based similarity metric using the I matrix we need a binary representation of the item description, where each vector position represents if the item has that description feature or not. To build these descriptions we have converted the multivalued features of the film descriptions (genres, directors, actors, ...) into binary values. This way we avoid the bias of knowledge-rich approaches that use more elaborated metrics to compute these multivalued features. Another advantage is that we could use the same cosine metric to compare both item descriptions in Q^u and I :

$$sim^{Q^u}(i, rec) = \cos(q_i^u, q_{rec}^u) \quad (9)$$

$$sim^I(i, rec) = \cos(I[i], I[rec]) \quad (10)$$

To estimate the quality of the recovered explanatory cases, we are going to compare them with the recommended item. Our evaluation metric will compute the Root Mean Square Error (RMSE) between the estimated rating r'_{ui} for the recommended item and the average of the actual user ratings for the k explanatory cases, either retrieved using sim^{Q^u} or sim^I . In the evaluation we used different k values, with $k \in \{1, 2, 3, 5, 10\}$.

The intuition behind this evaluation is that, given a recommended item to be explained, the explanatory cases should have a real rating given by the user similar to the estimated rating provided by the recommender system for the recommended item. As we are using a stratified evaluation, this approach let us validate if the proposed method could be useful to explain both positive (high estimated rating) and negative recommendations (low estimated rating).

5.3 Results

Table 3 shows the RMSE values that we have obtained using both similarity metrics. We observe that the use of the sim^{Q^u} metric to retrieve the explanatory cases decreases the RMSE value. In other words, the rating given by the user to the cases that are recovered with the descriptions of q_i^u are more similar to the rating estimated for the recommended item than using binary descriptions in a content-based style. The third column shows the improvement percentage using the methodology proposed in this paper. In the table we see that the best result is with the value of $k = 1$ where the improvement is 5.5%.

The corresponding results of the stratified evaluation are shown in Table 4. We observe again the best results with low k values. The explanation for this behaviour, both in the complete and the stratified dataset, is the highest performance of the sim^{Q^u} metric when presenting few explanatory cases. On the other

Table 3. RMSE using the complete evaluation set.

k	sim^I	sim^{Q^u}	Improvement (%)
1	1.211	1.144	5.53
2	1.194	1.144	4.16
3	1.189	1.136	4.47
5	1.190	1.143	3.94
10	1.179	1.148	2.67

Table 4. RMSE using the stratified dataset.

k	sim^I	sim^{Q^u}	Improvement (%)
1	1.229	1.125	8.47
2	1.208	1.139	5.71
3	1.211	1.134	6.34
5	1.199	1.146	4.42
10	1.180	1.148	2.71

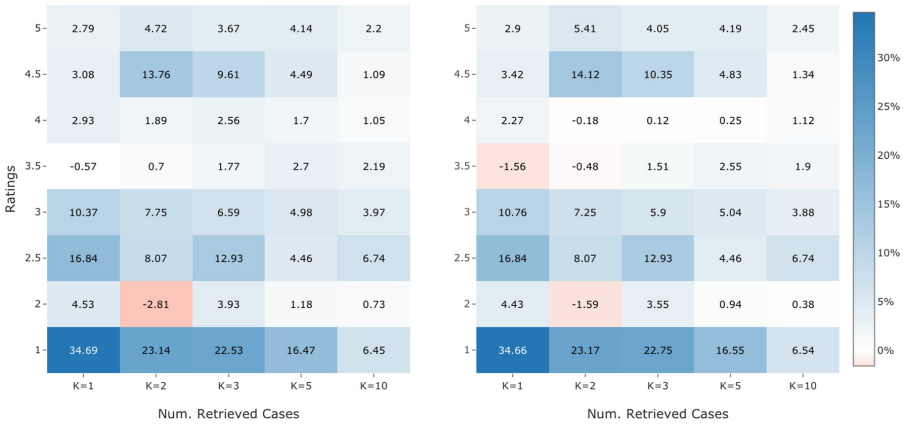


Fig. 4. Graphical representation of the improvement percentage of the sim^{Q^u} metric with respect to the content-based approach sim^I . Left heatmap corresponds to the results of the complete dataset shown in Table 5, whereas heatmap on the right corresponds to the stratified dataset detailed in Table 6. Red cells represent negative improvement (content-based approach is better than latent factors metric). (Color figure online)

hand, as the number of explanatory cases increases, the content-based approach is able to leverage its worse performance.

Next, Tables 5 and 6 show the results segmented by the rating value³. As a general result, we can conclude that the latent factors obtain better results than the content-based approach. The corresponding improvements (in percentage) are illustrated by Fig. 4. This figure let us observe that the similarity metric based on the vector of factors is not only able to explain a movie that the user may like (high predicted ratings), but also to explain why the user won't like a move. It is specially remarkable for those movies with a very low rating, where our approach achieves the highest performance. This figure also illustrates the behaviour of the proposed similarity metric when presenting to the user few explanatory cases, that was summarized by Tables 3 and 4.

Table 5. Detailed RMSE using the complete dataset segmented by the rating value.

Rating	Size	k = 1		k = 2		k = 3		k = 5		k = 10	
		sim^I	sim^{Q^u}	sim^I	sim^{Q^u}	sim^I	sim^{Q^u}	sim^I	sim^{Q^u}	sim^I	sim^{Q^u}
1	37	1.18	0.77	1.08	0.83	1.10	0.85	1.07	0.89	0.99	0.93
2	85	1.04	1.09	0.99	1.02	1.02	0.98	0.98	0.97	0.98	0.97
2.5	34	1.09	0.91	1.03	0.95	1.06	0.93	0.98	0.94	0.99	0.92
3	295	0.98	0.89	0.95	0.87	0.93	0.87	0.94	0.89	0.93	0.89
3.5	95	0.84	0.84	0.81	0.81	0.82	0.81	0.86	0.84	0.86	0.84
4	304	0.90	0.88	0.93	0.91	0.91	0.89	0.92	0.90	0.91	0.90
4.5	66	0.97	0.94	1.05	0.91	1.02	0.92	1.00	0.96	0.94	0.93
5	219	1.06	1.03	1.03	0.98	1.02	0.99	1.01	0.97	1.00	0.98

Table 6. Detailed RMSE using the stratified dataset segmented by the rating value.

Rating	Size	k = 1		k = 2		k = 3		k = 5		k = 10	
		sim^I	sim^{Q^u}	sim^I	sim^{Q^u}	sim^I	sim^{Q^u}	sim^I	sim^{Q^u}	sim^I	sim^{Q^u}
1	3400	1.18	0.77	1.08	0.83	1.10	0.85	1.07	0.89	0.99	0.93
2	3400	1.04	0.99	1.00	1.01	1.02	0.98	0.97	0.97	0.98	0.97
2.5	3400	1.09	0.91	1.03	0.95	1.06	0.93	0.98	0.94	0.99	0.92
3	3400	0.98	0.88	0.94	0.87	0.93	0.87	0.93	0.89	0.93	0.89
3.5	3400	0.83	0.85	0.81	0.81	0.82	0.80	0.86	0.84	0.86	0.84
4	3400	0.92	0.90	0.93	0.93	0.91	0.91	0.92	0.92	0.92	0.91
4.5	3400	0.99	0.95	1.06	0.91	1.02	0.92	1.01	0.96	0.94	0.93
5	3400	1.05	1.02	1.02	0.97	1.02	0.98	1.01	0.97	1.00	0.97

³ Ratings 0.5 and 1.5 have been removed due to the low number of items.

6 Conclusions and Future Work

Case-based explanation approaches in recommender systems typically use previous items rated by the user in order to explain a recommendation. The novelty of the approach described in this paper is that it infers a similarity measure from the vector of factors obtained by the matrix factorization algorithm and uses this similarity measure to capture the preferences of the user from previous rating patterns (cases). We have proposed a case-based explanation model where the explanatory cases are retrieved from the Q^u matrix, computed by the matrix factorization algorithm, instead of using an item description space, as the Q^u captures the user preferences in a N -dimensional space.

Matrix factorization decomposes the user-item interaction matrix into the dot product of two lower dimensionality matrices, P and Q , using N latent features. This way, each row in P represents the strength of the associations between a user and the latent features. Similarly, each row in Q represents the strength of the associations between an item and the latent features. In this paper we propose combining both P and Q matrices to get a personalized representation of the items in a lower dimensional space according to the user preferences. Although these vectors of latent features are not easy to understand and they cannot be directly exploited to explain the recommendation, we propose to use them in a case-based explanation style. We use the personalized Q^u matrix in order to find those past items rated by the user that are related to the current recommendation in that latent factor space.

The empirical evaluation that we have conducted compares the quality of the explanatory cases obtained by our proposal with a canonical content-based approach. Results reveal a clear improvement specially remarkable for explanatory cases with low ratings. This way, we can provide explanations with a positive or negative perspective, showing examples of why an item is interesting to the user or not. The similarity metric based on the latent factors also achieves good results when proposing explanation based on very few items.

As future work we would like to evaluate this approach with real users and to compare our similarity metric to retrieve the explanatory cases with other similarity distances. We also plan to evaluate this approach using an external recommender system acting as a black box. This way, we do not need to know the underlying recommender algorithm, and compute the Q matrix to obtain explanatory examples instead of using this matrix to provide recommendations.

We would also explore the possibility of associating a semantic description to the latent feature vectors that capture the user preferences. If we could correlate these vectors to a semantic description of the items or the user's profile we could provide a more detailed explanation about the recommended item. However, as explained in this paper, this correlation is not intuitive and very difficult to obtain.

References

1. Bell, R., Koren, Y., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* **42**(08), 30–37 (2009). <https://doi.org/10.1109/MC.2009.263>
2. Bennett, J., Lanning, S., et al.: The netflix prize. In: *Proceedings of KDD Cup and Workshop*, New York, NY, USA, vol. 2007, p. 35 (2007)
3. Caro-Martinez, M., Jimenez-Diaz, G., Recio-Garcia, J.A.: A theoretical model of explanations in recommender systems. In: *ICCBR 2018*, p. 52 (2018)
4. Doyle, D., Tsymbal, A., Cunningham, P.: A review of explanation and explanation in case-based reasoning. D2003, TCD CS report (2003)
5. Finch, I.: Knowledge-based systems, viewpoints and the world wide web. In: *IEE Colloquium on Web-Based Knowledge Servers (Digest No. 1998/307)*, pp. 8/1–8/4, June 1998
6. Funk, S.: Netflix update: Try this at home (2006). <https://sifter.org/simon/journal/20061211.html>
7. Gedikli, F., Jannach, D., Ge, M.: How should I explain? A comparison of different explanation types for recommender systems. *Int. J. Hum.-Comput. Stud.* **72**(4), 367–382 (2014)
8. Gómez, H., et al.: A case-based reasoning system for aiding detection and classification of nosocomial infections. *Decis. Support Syst.* **84** (2016). <https://doi.org/10.1016/j.dss.2016.02.005>
9. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. *ACM Trans. Interact. Intell. Syst.* **5**(4), 19:1–19:19 (2015). <https://doi.org/10.1145/2827872>
10. He, X., Chen, T., Kan, M.Y., Chen, X.: Trirank: review-aware explainable recommendation by modeling aspects. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM 2015*, pp. 1661–1670. ACM, New York (2015). <https://doi.org/10.1145/2806416.2806504>
11. Jameson, A.: More than the sum of its members: challenges for group recommender systems. In: *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI 2004*, 25–28 May 2004, Gallipoli, Italy, pp. 48–54 (2004). <https://doi.org/10.1145/989863.989869>
12. Lee, O., Jung, J.J.: Explainable movie recommendation systems by using story-based similarity. In: *Joint Proceedings of the ACM IUI 2018 Workshops Co-located with the 23rd ACM Conference on Intelligent User Interfaces (ACM IUI 2018)*, 11 March 2018, Tokyo, Japan (2018). <http://ceur-ws.org/Vol-2068/exss5.pdf>
13. Lops, P., de Gemmis, M., Semeraro, G.: Content-based recommender systems: state of the art and trends. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) *Recommender Systems Handbook*, pp. 73–105. Springer, Boston, MA (2011). https://doi.org/10.1007/978-0-387-85820-3_3
14. Mcsberry, D.: Explanation in recommender systems. *Artif. Intell. Rev.* **24**(2), 179–197 (2005). <https://doi.org/10.1007/s10462-005-4612-x>
15. Muhammad, K., Lawlor, A., Rafter, R., Smyth, B.: Great explanations: opinionated explanations for recommendations. In: Hüllermeier, E., Minor, M. (eds.) *ICCBR 2015. LNCS (LNAI)*, vol. 9343, pp. 244–258. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24586-7_17
16. Papadimitriou, A., Symeonidis, P., Manolopoulos, Y.: A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining Knowl. Discov.* **24**(3), 555–583 (2012). <https://doi.org/10.1007/s10618-011-0215-0>

17. Pu, P., Chen, L.: Trust-inspiring explanation interfaces for recommender systems. *Knowl.-Based Syst.* **20**(6), 542–556 (2007). <https://doi.org/10.1016/j.knosys.2007.04.004>
18. Quijano-Sanchez, L., Sauer, C., Recio-Garcia, J.A., Diaz-Agudo, B.: Make it personal: a social explanation system applied to group recommendations. *Expert Syst. Appl.* **76**, 36–48 (2017)
19. Ray, S., Sharma, R.: Explanations in recommender systems: an overview. *Int. J. Bus. Inf. Syst.* **23**, 248 (2016). <https://doi.org/10.1504/IJBIS.2016.10000276>
20. Tintarev, N., Masthoff, J.: Explaining recommendations: design and evaluation. In: Ricci, F., Rokach, L., Shapira, B. (eds.) *Recommender Systems Handbook*, pp. 353–382. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_10
21. Wang, X., He, X., Feng, F., Nie, L., Chua, T.S.: TEM: tree-enhanced embedding model for explainable recommendation. In: *Proceedings of the 2018 World Wide Web Conference, WWW 2018*, pp. 1543–1552. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2018). <https://doi.org/10.1145/3178876.3186066>
22. Yueming: Imdb 5000 movie dataset (2018). <https://www.kaggle.com/carolzhangdc/imdb-5000-movie-dataset>
23. Zanker, M., Ninaus, D.: Knowledgeable explanations for recommender systems. In: *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010*, 31 August–3 September 2010, Toronto, Canada, Main Conference Proceedings, pp. 657–660 (2010). <https://doi.org/10.1109/WI-IAT.2010.131>
24. Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., Ma, S.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2014*, 06–11 July 2014, Gold Coast, QLD, Australia, pp. 83–92 (2014). <https://doi.org/10.1145/2600428.2609579>