Search for User-related Features in Matrix Factorization-based Recommender Systems

Marharyta Aleksandrova[1,2], Armelle Brun[1], Anne Boyer[1], and Oleg Chertov[2]

[1]Université de Lorraine - LORIA,
Campus Scientifique, 54506 Vandoeuvre les Nancy, France
[2]National Technical University of Ukraine "Kyiv Polytechnic Institute",
37, Prospect Peremohy, 03056, Kyiv, Ukraine
{marharyta.aleksandrova,armelle.brun,anne.boyer}@loria.fr,
chertov@i.ua

Abstract. Matrix factorization (MF) is one of the most powerful approaches used in the frame of recommender systems. It aims to model the preferences of users about items through a reduced set of latent features. One main drawback of MF is the difficulty to interpret the automatically formed features. Following the intuition that the relation between users and items can be expressed through a reduced set of users, referred to as representative users, we propose a simple modification of a traditional MF algorithm, that forms a set of features corresponding to these representative users. On one state of the art dataset, we show that proposed representative users-based non-negative matrix factorization (RU-NMF) discovers interpretable features and does not significantly decrease the accuracy on test with 10 and 15 features.

Keywords: Recommender systems, matrix factorization, features interpretation.

1 Introduction

Recommender systems, that belong to machine learning area, aim to estimate preferences (ratings) of target users on previously non-seen items, in order to recommend them those items, which would probably satisfy these target users [1]. Recommendation algorithms are used in a wide area of real services starting from electronic commerce to considering search engines as a special type of recommender systems.

In order to estimate unknown preferences, recommender systems can use information about the content of the items (content-based methods), preferences of other users (collaborative filtering) or the both sources (hybrid approaches) [1]. In the frame of collaborative filtering we can outline such approaches as neighborhood-based [1] and matrix factorization-based techniques [2]. The first approach searches for neighbor users, who have similar preferences as the target user and recommends items that were highly rated by his neighbors. Thus, if needed, recommendation can be easily explained: a certain item was recommended because it was highly rated by the users having similar tastes to the active one.

Matrix factorization relies on the idea that there is a small number of latent factors (features) that underly the preferences (interactions) between users and items. As these features are defined so as to fit at best the data, no obvious interpretation can be made of them and as a result, unlike neighborhood-based approaches, recommendations have no obvious explanation. However, as it was shown in [3], providing explainable recommendation remains important for the users fidelity.

Based on the assumption that the preferences between users are correlated, we assume that within the entire set of users, there is a small set of users that have a specific role or have specific preferences. These users can be considered as representative of the entire population and we intend to discover features that are associated with these representative users. We think that if the discovered features would represent elements from the real world, they could not only be interpretable, but the recommendations could also be easily explained, similarly to the explanation provided by neighbor-based approaches, where neighbors are replaced by representative users. In order to identify these representative users we propose a representative users-based non-negative matrix factorization approach (RU-NMF), which is a slight modification of traditional non-negative matrix factorization technique based on multiplicative update rules.

This paper is a part of the work in progress about the discovery of features related to reality in matrix factorization-based approaches. In the current research we rise two main questions: (a) can MF algorithms result in features that can be associated with real users? (b) will the quality of recommendations be reduced if such associations are made explicit? We also point out some potential benefits of the resulting model. To the best of our knowledge, this work is the first one that is interested in not only interpreting features in MF-based recommendation approaches, but also in constraining these features so that they correspond to real elements of the system.

2 Related works

Let M be the number of users and N the number of items. The interaction between these entities is usually represented under the form of a matrix R ($\dim(R) = M \times N$) with elements r_{mn} corresponding to the rating assigned by the user m to the item n. Thus the recommendation problem is reduced to the task of estimating the missing values in R.

MF techniques decompose the original rating matrix R into two low-rank matrices U ($\dim(U) = K \times M$) and V ($\dim(V) = K \times N$) in such a way that the product of these matrices approximates the original rating matrix $R \approx R^* = U^T V$ with respect of the condition of minimal loss. This task is usually solved by optimization methods, such as gradient descent or alternating least squares [4]. The set of factors K can be seen as a joint latent space on which a mapping of both users and items spaces is performed [4]. Thus matrices U and V can be considered as transfer matrices to the new feature space from the spaces of users and items respectively. MF techniques have recently attracted more attention than traditional neighborhood-based approaches [1], as they are adequate for large-scale and sparse datasets [5] and have proven to result in models of both low-complexity and good accuracy (see Netflix Prize competition [4, 6]).

Features resulting from factorization usually do not have any physical sense, what makes resulting recommendations less explainable. Some authors made attempts to interpret them by using non-negative matrix factorization based on multiplicative update rules (further referred to as NMF). NMF imposes the condition of non-negativity on the values of matrices U and V, to ensure that each user profile can be represented as an additive linear combination of coordinates [7]. [8, 7] assumed that the features formed can be related to behavioral patterns, or to groups of users. However, the interpretation of each feature is not so easy to perform as it has to be discovered manually, by analyzing the content of the matrices. Authors of [9] focused on the explanation of the recommendations with MF techniques. With this aim, MF and neighborhood-based approaches are combined through weighting schemes. Nevertheless, such a method allows only partial explanation of the recommendations.

So we can conclude that relatively few works concern feature interpretation in MF-based recommendation techniques and the proposed approaches either require human post-processing or provide only partial interpretation. Still such an interpretation could be useful not only for understanding and characterizing the relation between users and items but also to make recommendations explainable.

3 The proposed approach: RU-NMF

3.1 Preliminaries

Let us consider 2 linear spaces L_1 and L_2 of dimensionality respectively 6 and 3, with basic vectors in canonical form $\{\boldsymbol{u}_m\}$, $m \in \overline{1,6}$ and $\{\boldsymbol{f}_k\}$, $k \in \overline{1,3}$. Let the transfer matrix from L_1 to L_2 be specified by matrix (1).

$$P = \begin{pmatrix} 0 & 0 & p_{13} & p_{14} & 1 & p_{16} \\ 1 & 0 & p_{23} & p_{24} & 0 & p_{26} \\ 0 & 1 & p_{33} & p_{34} & 0 & p_{36} \end{pmatrix} \tag{1}$$

$\boldsymbol{u}_5$, $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are direct preimages of $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$ respectively. Indeed, $P\boldsymbol{u}_5 = P\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T = \boldsymbol{f}_1$. By analogy, $P\boldsymbol{u}_1 = \boldsymbol{f}_2$, $P\boldsymbol{u}_2 = \boldsymbol{f}_3$. At the same time vectors $\boldsymbol{u}_3$, $\boldsymbol{u}_4$ and $\boldsymbol{u}_6$ will be mapped into linear combinations of basic vectors $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ and $\boldsymbol{f}_3$. For example, $P\boldsymbol{u}_3 = p_{13}\boldsymbol{f}_1 + p_{23}\boldsymbol{f}_2 + p_{33}\boldsymbol{f}_3$ presents the linear combination for $\boldsymbol{u}_3$.

3.2 RU-NMF

As mentioned previously, matrix U can be considered as a transfer matrix from the space of users to the space of features. Analyzing the example considered above, we can say that if matrix U has a form similar to (1), *i.e.* U has exactly K unitary columns with one non-zero and equal to 1 element on different positions, then the users corresponding to these columns are direct preimages of the K features. We say they represent the canonical coding of the features, following [10]. The features can thus be directly interpreted as users. These users will be referred to as representative users.

Obviously, in the general case, one cannot guarantee that the matrix U will be in a form similar to matrix (1). Worse, none of the column-vectors of matrix U may directly represent the canonical form of a feature. However we could design a matrix factorization approach that imposes appropriate constraints. In our case, the constraints would be the following: K columns in U have to represent the canonical coding of K different features. In order to solve this problem we propose the RU-NMF approach, that forms both matrices U and V, with features corresponding to representative users. The whole process consists of 6 steps, further detailed below.

Step 1. A traditional matrix factorization is performed, resulting in both matrices U and V with K features. As in [8, 7], that were also focusing on the interpretation of features, we used non-negative matrix factorization based on multiplicative update rules.

Step 2. A normalization of each of the M column vectors of the matrix U is performed so as to result in unitary columns. The resulting normalized matrix is denoted by U_{norm} and the set of normalization coefficients by C.

Step 3. This step is dedicated to the identification of the representative users in the U_{norm} matrix. We will consider user u_m as the best preimage candidate for the feature f_k if the vector in matrix U_{norm} corresponding to the user u_m will be the closest to the corresponding canonical vector (a vector with the only one non-zero and equal to 1 value on position k). The notion of closeness between vectors is expressed in Euclidean distance. That is the task of finding representative user u_m is reduced to solving the optimization problem (2).

$$dist(f_k, u_m^{norm}) \to \min \qquad (2)$$

where u_m^{norm} is the vector from matrix U_{norm} corresponding to the user u_m. Let us consider the following example. Assume that we have vector α of the form $\left(\alpha_1\ \alpha_2\ \ldots\ \alpha_K\right)^T$ with unique norm ($\sqrt{\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_K^2} = 1$). Then the distance between α and the first canonic vector $f_1 = \left(1\ 0\ \ldots\ 0\right)^T$ is expressed by $dist^2(f_1, \alpha) = (1 - \alpha_1)^2 + \alpha_2^2 + \ldots + \alpha_K^2$. Performing simple mathematical transformations we can obtain equation (3).

$$dist^2(f_1, \alpha) = 2(1 - \alpha_1) \qquad (3)$$

This means that the minimum of the distance is obtained under the condition $\alpha_1 \to \max$. Taking into account this reasoning, we consider a user u_m as a preimage candidate for the feature f_k if the maximum value of appropriate vector u_m^{norm} is situated on the position k; and the best preimage candidate is the one among all candidates with the highest maximum.

Let us analyze what can be the highest value of distance (2) between a canonic vector f_k and a preimage candidate vector u_m^{norm}. We have already noted that maximum value of the candidate vector u_m^{norm} is situated on the position k, otherwise u_m will be considered as a preimage candidate for another feature. Considering the formula (3) we can say that the maximum of distance is reached when the maximum value of the vector u_m^{norm} is as small, as possible. Obviously this condition holds only for the vector u_m^{norm} with all equal values $\left(\frac{1}{\sqrt{K}}\ \frac{1}{\sqrt{K}} \cdots \frac{1}{\sqrt{K}}\right)^T$, where K is the dimensionality of u_m^{norm}. In this case distance will be equal to $dist^{max} = \sqrt{2(1 - \frac{1}{\sqrt{K}})}$.

As we can see maximal value of distance depends on the dimensionality of the feature space. So, in order to unify characteristics of representative users considering different number of features and for simplicity of analysis, we propose to use the quality score (4) for the identification of representative users and their characterization.

$$q(u_m) = 1 - \frac{dist(f_k, u_m)}{dist^{max}(K)} \tag{4}$$

The highest quality score, namely 1, is assigned to a candidate with a vector equal to the canonical coding as the appropriate distance is equal to 0. The lowest quality score, namely 0, is assigned to candidates with no influencing coordinates (all values are equal).

Thus on the third step all users are divided into subgroups of preimage candidates for each feature (according to the position of maximal value in u_m^{norm}). After this, the user with the highest quality score among all candidates is considered as the representative one for the current feature.

Once all preimages are identified, the matrix U_{norm} is modified so as to obtain a matrix in a form of (1): in each column that corresponds to a representative user, a 1 is assigned to the coordinate at the position of the maximum value and a 0 is assigned to all others. The resulting modified matrix is the matrix U_{norm}^{mod}.

In some cases, a feature, say feature f_k, may have no candidate preimage. In this case we can either decrease the number of features considered for factorization or search for a vector with the second maximum situated on that specific position.

Step 4. Each column of the matrix U_{norm}^{mod} is multiplied to the appropriate normalization factor from the set C resulting in matrix U^{mod}. After this, representative users will remain preimages of the features but with scaling coefficients.

Step 5. In order to obtain the best model we also have to modify the matrix V under the condition of minimal loss. The modification of V can be performed using optimization methods with the starting value obtained during the first step. As the objective of this paper is to determine the relevance of finding preimages of the features and to quantify the decrease of the quality of the recommendations, we did not consider this step.

Step 6. The resulting recommendation model is made up of matrices U^{mod} and V according to the formula $R^* = \left(U^{mod}\right)^T V$.

4 Experimental results

4.1 Datasets and Evaluation

In order to evaluate RU-NMF, we perform experiments on the 100k MovieLens dataset[1], which contains 100k ratings, ranging from 1 to 5, assigned by 943 users to 1682 movies (items). In all experiments, 80% of the ratings are used for learning the model and 20% for testing it. We prepare 30 different pairs of learning and test sets with the first one randomly chosen from the original 100k MovieLens dataset and the second one made up of the remaining part. The accuracy of the models are evaluated with two classical measures: mean absolute error (MAE) and root mean square error (RMSE) [11].

4.2 Quality of the Representative Users

In this section we answer the question (a), namely if MF algorithms can result in features that can be associated with real users. For each of the 30 datasets, we perform NMF with 10, 15 and 20 features. After that, for each run we order the representative users by their quality score and compute the mean quality at each rank (considering the number of features used for factorization). The corresponding values are presented in figure 1. When the number of features is equal to 10, the quality score of the representative users is particularly high: 90% of them have a quality score higher than 0.8 and 60% have a score higher than 0.9. With 15 and 20 features the quality of representative users decreases. For example, when the number of features is equal to 15, only half of the vectors corresponding to representative users have a quality score above 0.8 and only 20% above 0.9. With 20 features only 30% of the representative users have a quality above 0.8 and 10% above 0.9. As a result we can say that when the number of features is equal to 10 NMF naturally results in features, which are very close to the searched canonic form, that means in features that can be interpreted as real users.

4.3 Traditional NMF versus RU-NMF

In this subsection we seek an answer to the question (b) will RU-NMF have a considerable impact on the accuracy of the recommendations. The left part of the table 1 presents the resulting mean and standard deviation values of both errors (MAE and RMSE) on the 30 datasets for NMF with 10, 15 and 20 features. The mean error value, for both MAE and RMSE, goes down on the learning set when the number of features grows up. In contrast, on the test set the errors increase with the number of features. This fact confirms the overfitting problem mentioned in many works [12] and partially supports the conclusion of [9] that the more adequate number of features on the MovieLens dataset is close to 10. We can mention that on the test set, standard deviation seems to decrease as the number of features increases, confirming the consistent increase in the error.

[1] http://grouplens.org/datasets/movielens/

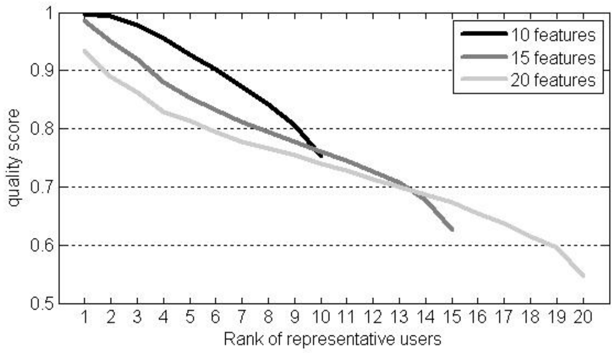

Fig. 1. Quality score for 10, 15 and 20 features.

Table 1. NMF vs RU-NMF: mean and standard deviation values of errors with 10, 15 and 20 features on learning and test sets.

	NMF				RU-NMF			
	Learning set		Test set		Learning set		Test set	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
	10 features				10 features			
mean	0.5482	0.7154	0.8014	1.0507	0.5491	0.7168	0.8018	1.0512
std	0.0019	0.0019	0.0067	0.0096	0.0021	0.0022	0.0067	0.0096
	15 features				15 features			
mean	0.4933	0.6529	0.8393	1.1035	0.4982	0.6613	0.8417	1.1071
std	0.0017	0.0020	0.0046	0.0068	0.0025	0.0039	0.0047	0.0071
	20 features				20 features			
mean	0.4461	0.5988	0.8689	1.1412	0.4585	0.6232	0.8749	1.1505
std	0.0011	0.0012	0.0057	0.0063	0.0029	0.0062	0.0060	0.0069

The right part of the table 1 presents the mean and standard deviation values of the two error measures, computed on the same datasets and the same number of features for RU-NMF. We can see similar dependences as for the traditional NMF: both errors decrease on the learning set while the number of features grows, and both errors increase on the test set. As on NMF, standard deviations seem to decrease on the test set. We can conclude that RU-NMF preserves almost the same characteristics as traditional NMF.

Next we compare the accuracies of RU-NMF and NMF. The accuracy loss ρ, defined by formula (5), computes the relative difference between the error obtained with RU-NMF ($err\,(\mathit{RU\text{-}NMF})$) and the error obtained with traditional NMF ($err\,(\mathit{NMF})$). A positive loss value means that NMF performs better than RU-NMF. Table 2 reports the accuracy loss ρ, computed on the same 30 datasets.

$$\rho = \frac{err\,(\mathit{RU\text{-}NMF}) - err\,(\mathit{NMF})}{err\,(\mathit{NMF})} 100\% \qquad (5)$$

Table 2. Accuracy loss ρ between RU-NMF and the traditional NMF, for 10, 15 and 20 features, %.

	10 features				15 features				20 features			
	mean	std	min	max	mean	std	min	max	mean	std	min	max
Learning set												
MAE	0.17	0.09	0.03	0.38	0.98	0.34	0.49	1.71	2.78	0.67	1.38	4.27
RMSE	0.19	0.10	0.03	0.46	1.29	0.49	0.61	2.38	4.08	1.08	1.94	6.64
Test set												
MAE	0.05	0.06	**-0.06**	0.18	0.29	0.19	**-0.06**	0.77	0.70	0.27	0.13	1.43
RMSE	0.05	0.07	**-0.07**	0.20	0.33	0.20	**-0.04**	0.79	0.82	0.31	0.12	1.53

The first conclusion that we can make when analyzing table 2 is that the accuracy loss increases with the number of features, on both learning and test sets, and for both error measures. In the worst case, the accuracy loss equals to 6.64%, for RMSE with 20 features, which is quite small. The lowest accuracy loss (0.05%) is obtained with 10 features for both errors. Standard deviation holds the same dependence: on test and learning sets, the accuracy loss is the least dispersive with 10 features. When comparing the accuracy loss between test and learning sets, we can note that the average loss is 3 times lower on test than on learning, for both errors and for all the number of features: thus we can say that RU-NMF has a lower relative loss between learn and test compared to NMF. A thorough analysis of the losses obtained on the 30 sets has shown that the accuracy loss on the test set is lower than the one on the learning set, in all cases, whatever is the error and the number of features. In some runs, RU-NMF has even a higher accuracy than NMF (see values in bold in table 2). This holds for 23% and 3% of the runs with 10 and 15 features respectively.

In order to estimate if the loss in accuracy between RU-NMF and NMF is statistically significant, we perform a statistical test. The null hypothesis H_0 denotes "The loss in error between NMF and RU-NMF is null". Student's tests with 99% confidence ($\alpha = 0.01$) have been performed and the results are presented in Table 3. In this table "H_0" represents the acceptance of the hypothesis and "-" its rejection. Considering 10 features on both learning and test sets and 15 features on the test set, both MAE and RMSE are not increased by RU-NMF (for example, the p-values with 10 features on MAE is equal to 0.8072). The null hypothesis is thus accepted for these numbers of features. In other cases, the errors on RU-NMF and traditional NMF models can not be considered as equal.

Considering this, we can conclude that a number of features equal to 10 provides not only the smallest values of errors on the test set, but also results in the representative users of the highest quality. Thus the quality of representative users can be considered as one of the potential indicators of the optimal number of features (the number of features, that results in the smallest error on the test set and that, thus, must be used in the factorization process). Also it may mean that an inverse logical conclusion takes place, notably representative users will be of a high quality only if the number of features is close to the optimal one.

Table 3. Results of Student's test with hypothesis H_0: "The loss in error between NMF and RU-NMF is null" for $\alpha = 0.01$.

	Learning set		Test set	
	MAE	RMSE	MAE	RMSE
10 features	H_0	H_0	H_0	H_0
15 features	-	-	H_0	H_0
20 features	-	-	-	-

5 Discussions and future work

This paper proposes a simple modification of the traditional matrix factorization approach (RU-NMF), that aims at forming not only interpretable features, but also features that represent elements from the reality (users). This work is a preliminary one and its main goal is to show that such features can be formed.

We have shown that the features resulting from a traditional approach (NMF) when the number of features is close to the optimal are naturally close to the canonic form. Thus the model can be slightly modified so as to correspond to real users, resulting in a small loss in the accuracy. When the number of features is equal to 10 and 15, this loss is even not statistically significant on the test set. Also it was shown that RU-NMF mainly preserves the same characteristics as traditional NMF. Thus the both questions raised in this paper were answered. The analysis of the accuracy loss has shown that the features formed by RU-NMF consistently disturb the accuracy on the test set less than on the learning set. This can be considered as a potential ability of factorization techniques with features related to reality to limit overfitting problem faced by many others.

From our point of view, the proposed interpretation has several important positive impacts on the way the model can be exploited. First, if such an interpretation can be made, the recommendations can be easily explained. Indeed, if each feature corresponds to a representative user, then the matrix V expresses preferences of representative users on items. Meanwhile, each line in the matrix U reveals interactions between the user, corresponding to this line, and a set of representative users. Thus each user of the population is linearly mapped on the basis related to representative users and the preferences of the latter ones are used to estimate the ratings of the whole population. That makes the recommendation process ideologically close to the neighborhood-based approaches with representative users used instead of neighbors. Second, as the estimated ratings of all users of the population are computed through the representative users, the latter can be viewed as mentor users in the population: the users who represent the preferences of entire population. They can also be viewed as the users to choose in poll studies. They can thus also be considered as those to be tracked, so as to follow the evolution of the preferences of the population. Finally, the approach we propose for this interpretation is automatic, it does not require any human expertise, unlike of other works focused on features interpretation.

In a future work, we would like to focus first of all on the verification of the hypothesis that users associated with the features can be really considered as representative ones. We consider that it can be done while solving the new item cold-start problem. Indeed, knowing preferences of identified users on a new item and their relations with all other users of the population (what is represented by matrix U) we can try to predict ratings of other users on this item. Accuracy of the resulting predictions will indicate if feature-related users can actually represent the whole population. At the opposite of many state of the art approaches that aim at tackling the cold-start problem, this one also requires no information about the content of the items. Second revealed properties of RU-NMF should be verified on other datasets. We would also like to investigate if other factorization techniques (such as those, based on gradient descent and alternating least squires) will result in features, that can be interpreted as real users.

References

1. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art," *IEEE tran. on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.
2. M. Wu, "Collaborative filtering via ensembles of matrix factorizations," in *Proceedings of KDD Cup and Workshop*, vol. 2007, 2007.
3. R. Sinha and K. Swearingen, "The role of transparency in recommender systems," in *CHI'02 extended abstracts on Human factors in computing systems*. ACM, 2002, pp. 830–831.
4. Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
5. G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems," *Journal of Machine Learning Research*, vol. 10, pp. 623–656, 2009.
6. ——, "On the gravity recommendation system," in *Proc. of KDD cup and workshop*, 2007.
7. S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization." in *Proc. of the 6th SIAM Conf. on Data Mining*, 2006.
8. J.-F. Pessiot, V. Truong, N. Usunier, M. Amini, and P. Gallinari, "Factorisation en matrices non-négatives pour le filtrage collaboratif," in *3rd Conf. en Recherche d'Information et Applications (CORIA 2006)*, 2006, pp. 315–326.
9. A. Moin and C. Ignat, "Hybrid weighting schemes for collaborative filtering," INRIA, Tech. Rep., 2014.
10. Y. Guermeur, A. Lifchitz, and R. Vert, *Kernel Methods in Computational Biology*. MIT Press, 2004, no. 9, ch. A kernel for protein secondary structure prediction, pp. 193–206.
11. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.
12. M. Graus, "Understanding the latent features of matrix factorization algorithms in movie recommender systems," Master's thesis, University of Eindhoven, Master of Science in Human-Technology Interaction, 2011.