An Application of Link Prediction in Bipartite Graphs: Personalized Blog Feedback Prediction

Krisztian Buza*

Dpt. of Computer Science and Inf. Theory
Budapest University of Techn. and Economics
1117 Budapest, Magyar tudósok körútja 2.,
Hungary
buza@cs.bme.hu

Ilona Galambos

Dpt. of Computer Science and Inf. Theory
Budapest University of Techn. and Economics
1117 Budapest, Magyar tudósok körútja 2.,
Hungary
galambos.ilus@gmail.com

Abstract: The last decade lead to an unbelievable growth of the importance of social media. One of the most interesting challenges associated with social media is predicting which user is expected to comment which blog. In this paper, we formulate the above task as a link prediction problem in bipartite graphs. Recently, sparse matrix factorization became popular for link prediction. We show that the conventional algorithm for the factorization of matrices has great potential to predict new links between blogs and users. However, it fails to capture the true distribution of comments and therefore a straight-forward application of conventional matrix factorization leads to suboptimal predictions that do not even outperform the simple baseline of random predictions in terms of overall precision and recall. In order to alleviate this deficiency, we devise a simple technique which improves the efficiency of the algorithm substantially.

Keywords: blogs, feedback prediction, link prediction, bipartite graphs, matrix factorization

1 Introduction

The last decade lead to an unbelievable growth of the importance of social media. While in the early days of social media, blogs, tweets, facebook, youtube, social tagging systems, etc. served more-less just as an entertainment of a few enthusiastic users, nowadays news spreading over social media may have substantial economical effect (see e.g. opinions about products and services) and impact on the most important changes of our society, such as the revolutions in the Islamic world, or US president elections.

One of the most interesting challenges associated with social media is predicting which user is expected to comment which blog. In this paper, we formulate the above task as a link prediction problem in bipartite graphs. Recently, sparse matrix factorization became popular for link prediction. Therefore, we apply matrix factorization to predict which user is expected to comment which blog.

We show that the conventional algorithm for the factorization of matrices has great potential to predict new links between blogs and users. However, it fails to capture the true distribution of comments and therefore a straight-forward application of conventional matrix factorization leads to suboptimal predictions that do not even outperform the simple baseline of random predictions in terms of overall precision and recall. In order to alleviate this deficiency, we devise a simple technique which improves the efficiency of the algorithm substantially.

The remainder of the paper is organized as follows: Section 2 gives a short overview of the most closely related works, in Section 3 we define the problem of personalized blog feedback prediction and explain

*Krisztian Buza is on research term at the University of Warsaw, Poland. This research was developed in the framework of the project TÁMOP - 4.2.2.C-11/1/KONV-2012-0013 ("Infokommunikációs technológiák és a jövő társadalma"). We acknowledge the DAAD-MÖB Researcher Exchange Program.

matrix factorization in detail. In the empirical study presented in Section 4 we show how to apply matrix factorization for the blog feedback prediction problem. Finally, we conclude in Section 5.

2 Related Work

Data mining techniques for social media have been studied by many researchers, see e.g. [1] and [2]. Our problem is inherently related to many web mining problems, such as opinion mining or topic tracking in blogs. For an excellent survey on opinion mining we refer to [3].

Out of the works related to blogs we point out that Pinto applied topic tracking methods [4], while Mishne exploited special properties of blogs in order to improve retrieval [5]. Despite its relevance, there are just a few works on feedback prediction for blogs. Yano and Smith used Naive Bayes, Linear and Elastic Regression and Topic-Poisson models to predict the *number* of feedbacks in political blogs [6]. In our previous work, we used Neural Networks, RBF Networks, Regression Trees and Nearest Neighbor models for predicting the *number* of feedbacks that a blog is expected to receive [7].

In this paper, we focus on personalized blog feedback prediction, i.e., instead of simply predicting the number of feedbacks that a blog is expected to receive, we aim at predicting *who* is expected to comment *which* blog. This task is closely related to the *Who Rated What in 2006* task of the KDD Cup 2007 [8]. Matrix factorization, combined with the prediction of the number of ratings per movie and per user, was shown to be successful for the *Who Rated What in 2006* task [9]. In contrast to [9], we focus on the domain of blogs (instead of the movie recommendations). Furthermore, we use a different matrix decomposition scheme (we use the $M \approx U \times V$ scheme instead of the SVD-scheme, i.e., $M \approx U \times \Sigma \times V$) and we use the number of estimated feedbacks directly (instead of using the number of estimated ratings for the calculation a "base prediction"). Moreover, we evaluate personalized blog feedback prediction in terms of precision and recall (instead of RMSE that was used to evaluate the solutions submitted to the KDD Cup).

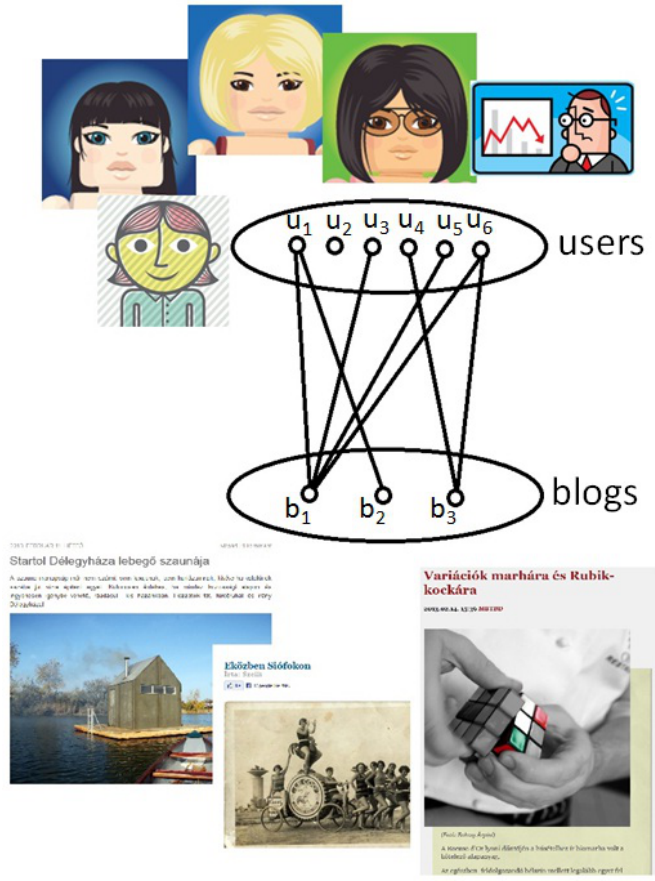
We formulate the problem of personalized feedback prediction as the task of predicting new links in a dynamically changing bipartite graphs. Matrix factorization not only became popular in recommender systems, see e.g. [10], but models based on matrix factorization were shown to be effective for link prediction tasks as well, see e.g. [11, 12], therefore, we base our approach on matrix factorization.

3 Blog Feedback Prediction as a Link Prediction Problem in Bipartite Graphs

On blogs, new documents (opinions, essays, images etc.) appear regularly. These documents are usually of short-term interest, i.e., most of the users read and comment such a document just a few hours or few days after the document was published. Usually, new documents appear after a short time on the same blog. Therefore, throughout this paper, by *blog* we mean the source on which various documents appear regularly. For example, we consider *torokgaborelemez.blog.hu* or *autozz.blog.hu* as a blog. Throughout the paper, we work at the level of *blogs*, not at the level of individual documents.

We formulate the personalized blog feedback prediction problem as a link prediction problem in dynamically changing bipartite graphs. Consider a bipartite graph G. The vertices in the first vertex class correspond users, while the vertices in the second vertex class corresponds blogs. Therefore, we call this graph *user-blog-graph*. A user-blog-graph is illustrated in the left of Figure 1. An edge means that the corresponding user comments on the corresponding blog. As the users may change their behavior, i.e., a user may begin to comment on a blog that she/he has not commented before, the graph is changing dynamically. Therefore, personalized blog feedback prediction can be seen as predicting the occurrence of new edges in the aforementioned graph.

We consider the following matrix representation of the user-blog-graph: the columns of the user-blog-matrix M correspond users, and the rows of the matrix correspond blogs. If there is an edge between a



	u_1	u_2	u_3	u_4	u_5	u_6
b_1	**1**	**0**	**1**	**0**	**1**	**1**
b_2	1	0	0	0	0	0
b_3	0	0	0	1	0	1

user-blog-matrix

	u_1	u_2	u_3	u_4	u_5	u_6
b_1	**1**	**0**	**1**	**0**	**1**	**1**
b_2	1	-	0	-	0	0
b_3	0	-	-	1	0	1

sampled user-blog-matrix

Figure 1: Personalized feedback prediction problem as link prediction problem in bipartite graphs (left). The corresponding user-blog-matrix is shown in the top right of the figure, while the sampled user-blog-matrix is shown in the bottom right of the figure.

user u and a blog b, the corresponding cell has the value of 1, otherwise it has the value of 0. This is illustrated in the top right of Figure 1.

With factorization of a matrix M, we mean to find matrices U and V so that $M \approx U \times V$. Let $m_{i,j}$ denote the value in the i-th row and j-th column of M. Furthermore, let $u_{i,j}$ (and $v_{i,j}$ respectively) denote the value in the i-th row and j-th column of U (and V respectively). Using this notation, we aim at finding U and V so that

$$\sum_{i,j} \left(m_{i,j} - \sum_{k=0}^{K} u_{i,k} v_{k,j}\right)^2 + \lambda\left(\sum_{i,j} u_{i,j}^2 + \sum_{i,j} v_{i,j}^2\right) \tag{1}$$

is minimized. Minimization of the above formula means to minimize the sum of squared errors plus λ-times a regularization term that avoids the model to become too complex. The parameter λ controls the relative importance of the regularization term compared to the sum of squared errors. We use K to denote the number of columns of U which is equal to the number of rows of V too. K is often called the number of latent factors.

We use a gradient descent algorithm for the minimization of (1). This is an iterative algorithm: matrices U and V are initialized with random values, then a sequence of update steps follows, in each of them we adjust the values of U and V. One of these update steps is illustrated in Figure 2. The pseudocode of the algorithm is shown in Figure 3. The parameter ϵ controls how large or small the

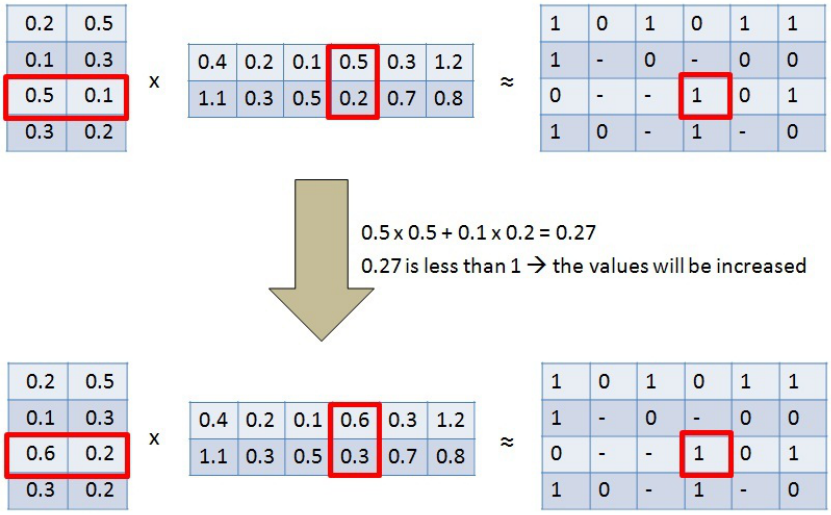


Figure 2: Illustration of one of the update steps of the matrix factorization algorithm.

update steps are relative to the current error of the estimation. This algorithm optimizes Formula (1), see also [13].

Similarly to the case of recommender systems, the matrix is highly sparse. In our case, this means that there are orders of magnitudes more 0-s in the matrix than 1-s. Such dominance of 0-s could "mislead" the factorization algorithm: roughly speaking, in terms of machine learning, this could result in learning that the matrix contains zeros everywhere. As such a model is useless in any practical applications, we downsample the matrix, i.e., we take take all the 1-s and randomly select the same amount of 0-s. This sampled user-blog-matrix is illustrated in the bottom right of Figure 1. Then the factorization algorithm is performed on the sampled user-blog-matrix. This corresponds to calculating the sum of squared error in (1) only for the positions contained in the sample and minimizing this modified formula.

4 Techniques of Using Matrix Factorization and Their Empirical Evaluation

In order to illustrate the potential of matrix factorization for the personalized feedback prediction problem, we empirically evaluated various prediction techniques based on matrix factorization. For the evaluation we used real-world data collected from the webpage of a blog provider. From all the data that we collected, we selected the most active 10% of the users and the blogs. We measured the activity of both the users and blogs in terms of their comments. Therefore, our dataset contained around 5000 users and 70 blogs.

For collecting the data, for the matrix factorization and for all the experiments described in this section, we used our own software that we implemented in Java.

We aimed at simulating the real-world scenario in which we want to make predictions for the future from the data collected in the past. Therefore, after preprocessing the data, we generated two user-blog-matrices. The first one, M_1, for period between May 2010 and April 2011, the second one, M_2, for the period between May 2011 and April 2012. We used the M_1 as training data, i.e., in order to determine the matrices U and V. The second matrix, M_2, was reserved as test data: we used it as gold standard in order to evaluate the predictions. That is, based on the matrices U and V that were determined using solely M_1, we predicted which users are likely to comment which blogs and we used M_2 to check whether these users really commented those blogs in the subsequent year.

We used precision and recall as evaluation metrics. We observed that users mostly comment blogs

```
Input: matrix M with n rows and m columns, integer K,
       real number eps, real number lambda
1.  Create U and V matrices and initialize their values randomly
    (U has n rows, K columns; V has K rows, m columns)
2.  While U x V does not approximate M well enough
    (or the maximal number of iterations is not reached)
3.     Select a known element of M, denote it by x
       let i and j denote the row and column of x
4.     Let x' be the dot product of the corresponding
       row of U and column of V
5.     err = x' - x
6.     for k=0; k < K; k++)
7.         u_{i,k} ← u_{i,k} - eps*err*v_{k,j} - lambda*u_{i,k}
8.         v_{k,j} ← v_{k,j} - eps*err*u_{i,k} - lambda*v_{k,j}
           // simultaneous update!
9.     end for
10. end while
```

Figure 3: The pseudocode of the matrix factorization algorithm.

which they already commented in the past. As this is a trivial trend, in our evaluation, we focus on how the approach can predict *new* links between users and blogs. Therefore, we calculated both precision and recall regarding the *new user-blog pairs*, i.e., we are interested in predicting that a user will comment a blog which he/she has not commented in the previous year. Therefore, we calculate precision (P) and recall (R) as follows:

$$P = \frac{\text{TP}}{\text{AllPred}} \qquad\qquad R = \frac{\text{TP}}{\text{AllNewLinks}}, \tag{2}$$

where TP denotes the true predictions, i.e., the number of predicted user-blog links that were among the true new user-blog links; AllPred denotes the number of all the predicted new links and AllNewLinks denotes the number of all the true new user-blog links. For both precision and recall, higher values indicate better quality.

For the factorization algorithm, we set $\lambda = 0.001$, $\epsilon = 0.01$, the number of latent factors $K = 10$ and we set the number of iterations to 5000. According to our observations, this amount of iterations were sufficient for the convergence of the algorithm.

Let $\hat{M} = U \times V$. In all of the techniques we evaluated, the predictions are based on $\hat{M}$. We consider the value in the i-th column and j-th row of $\hat{M}$ as a likelihood that user i will comment blog j.

4.1 Top n_b blogs

In the first experiment, for each user u, we selected n_b *new** blogs that are most likely to be commented by the user u according to $\hat{M}$. Technically, this means that for each user u, we selected those blogs which contains the highest values in the corresponding column of $\hat{M}$, and has not been commented by user u in according to the matrix M_1. Precision and recall depend on how we select the number n_b, and we expect a trade-off between precision and recall, i.e., while precision is relatively high, recall is expected to be relatively low and vice versa. Varying n_b, we observe precision-recall pairs and therefore we can plot precision as function of recall as shown in the top left of Figure 4. We compare the performance of this approach to the performance of random predictions, i.e., simply selecting for each user n_b blogs by chance. As one can see in the top left of Figure 4, our approach systematically outperforms random predictions.

*with new blogs we mean blogs that have not been commented by u in the train period

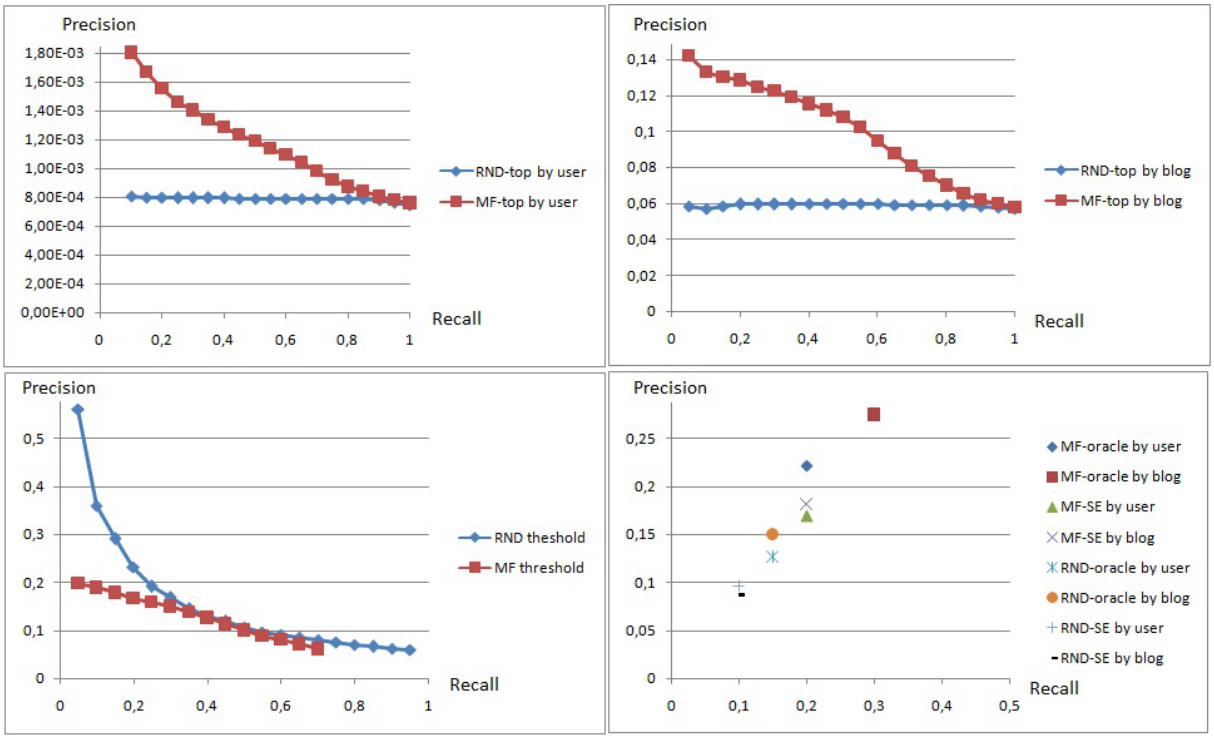


Figure 4: Experimental results

4.2 Top n_u users

We repeated the previous experiment from the perspective of the blogs, i.e., for each blog, we selected n_u users that were most likely to begin to comment those blogs according to $\hat{M}$. The results are shown in the top right of Figure 4. Again, the predictions based on matrix factorization systematically outperform random predictions.

4.3 Predictions based on thresholding $\hat{M}$

Predicting the same number of new blogs for all the users may be relevant to commercial applications, for example, in the case if we want to advertise some blogs to the users, and we can advertise each user the same amount of blogs. However, if we want to predict *who will comment which blog*, we can not assume that all the users will begin commenting the same amount of blogs in the next year. The low precision values in the top left of Figure 4 are explained by the fact that such an assumption is not valid in reality.

In order to take one step further and to allow to predict different number of blogs for different users, the straight forward solution is to set a threshold θ and to assume that all the values in $\hat{M}$ that are above θ correspond to predicting that the corresponding user will comment the corresponding blog. Similarly to the previous two experiments, we only consider *new* user-blog links, i.e., user-blog links that are not included in M_1. We compare such predictions to the predictions resulting from thresholding a random matrix M^{RND}. With varying the values of the threshold, we again observe the trade-off between precision and recall. In this case, however, the predictions based on matrix factorization do not outperform the random predictions as shown in the bottom left of Figure 4.

4.4 Estimation of the number of new links

The reason for this low performance of the previous experiment is that the distribution of comments are not taken into account properly by the matrix factorization algorithm. Therefore, we examined how the prediction quality changes if the take into account the number of expected new user-blog links additionally. Denote $n_b(u)$ the number of expected new user-blog links of user u, i.e., the number of new blogs that user u is expected to comment. Analogously, let $n_u(b)$ denote the number of expected new user-blog links of blog b, i.e., the number of new users who are expected to comment blog b. As simple estimations for $n_b(u)$ and $n_u(b)$, we use the number of *new* user-blog links per user and per blog in the *train* period, i.e., between May 2010 and April 2011. Such estimation of the number of new links is denoted as "SE" (simple estimation) in the bottom right of Figure 4.

We might also assume that we would have a perfect estimator, an Oracle, for the number of new user-blog links per user and per blog. Experimentally, we can simulate such a situation by setting $n_b(u)$ and $n_u(b)$ to the true number of new user-blog links that the user u and blog b received. This "perfect" estimator of the number of new user-blog links per user and per blog is denoted as "Oracle" in the bottom right of Figure 4.

Using the above estimators and $\hat{M}$, for each user u, we select $n_b(u)$ *new* blogs that are most likely to be commented by the user u. The results entitled with "... by user" results refer to this selection in the bottom right of Figure 4.

Similarly, for each blog b, we can select $n_u(b)$ users that are most likely to comment that blog b. The results entitled with "... by blog" refer to this selection in the bottom right of Figure 4.

Again, we compare the quality of predictions resulting from matrix factorization – denoted as MF – to the quality of predictions resulting from a random matrix which is denoted as RND in the Figure. The bottom right of Figure 4 shows that predictions based on matrix factorization clearly outperform random predictions both in terms of precision and recall for both cases of the previously described estimation of the number of new user-blog links, i.e., both for the realistic simple estimator (SE) and for the idealistic perfect estimator (Oracle). Furthermore, we note that the results achieved by using matrix factorization with the simple estimator are above the precision-recall curves of the *"Top n_u users"* and *"Top n_b blogs"* approaches shown in the top of Figure 4. This indicates that taking the number of estimated new user-blog links into account improves over the simple cases of (i) predicting the same number of new users for each blog and (ii) predicting the same number of new blogs for each user.

5 Conclusion and Future Work

In this paper, we focused on personalized feedback prediction for blogs, i.e., we aimed at predicting which user will comment on which blogs. We formulated this problem as a link-prediction problem in bipartite graphs and we used an approach based on matrix factorization to solve it. We selected matrix factorization because approaches based on matrix factorization were shown to be very successful for recommender systems and link prediction and therefore they become very popular in the last decade. We have shown that straight forward re-use of simple techniques from the recommender systems domain lead to suboptimal performance in case of our problem, therefore, we introduced a simple technique of count estimation which improved the quality of predictions substantially.

As future work one can consider more sophisticated estimation of the number of new links as well as the usage of other matrix factorization algorithms such as the ones introduced in [14, 15].

References

[1] T. Reuter, P. Cimiano, L. Drumond, K. Buza, L. Schmidt-Thieme, Scalable Event-Based Clustering of Social Media Via Record Linkage Techniques, 5th International AAAI Conference on Weblogs and Social Media (2011)

[2] L.B. Marinho, K. Buza, L. Schmidt-Thieme, Folksonomy-Based Collabulary Learning, The Semantic Web - ISWC 2008, Lecture Notes Computer Science (2008) **5318**, pp. 261-276

[3] B. Pang, L. Lee, Opinion Mining and Sentiment Analysis, Journal Foundations and Trends in Information Retrieval (2008) **2**, pp. 1-135

[4] J.P.G.S. Pinto, Detection Methods for Blog Trends. Report of Dissertation Master in Informatics and Computing Engineering, Faculdade de Engenharia da Universidade do Porto (2008)

[5] G. Mishne, Using Blog Properties to Improve Retrieval, International Conference on Weblogs and Social Media (2007)

[6] T. Yano, N.A. Smith, Whats Worthy of Comment? Content and Comment Volume in Political Blogs, 4th International AAAI Conference on Weblogs and Social Media (2010), pp. 359–362

[7] K. Buza, Feedback Predicition for Blogs, The 36th Annual Conference of the German Classification Society on Data Analysis, Machine Learning and Knowledge Discovery (2012)

[8] J. Bennett, C. Elkan, B. Liu, P. Smyth, D. Tikk, KDD Cup and workshop 2007. SIGKDD Explor. Newsl. (2007) **9**, 2 (December 2007), pp. 51–52

[9] M. Kurucz, A. A. Benczúr, T. Kiss, I. Nagy, A. Szab, B. Torma, Who Rated What: a combination of SVD, correlation and frequent sequence mining, Proc. KDD Cup and Workshop (2007) **23**

[10] R. Karimi, C. Freudenthaler, A. Nanopoulos, L. Schmidt-Thieme, Exploiting the Characteristics of Matrix Factorization for Active Learning in Recommender Systems, Doctoral Symposium of the 6th Annual ACM Conference on Recommender Systems (2012), Dublin, Irelan, pp. 317-320.

[11] E. Acar, D.M. Dunlavy, T.G. Kolda, Link prediction on evolving data using matrix and tensor factorizations, IEEE International Conference on Data Mining Workshops (2009), pp. 262–269

[12] A. Menon, E. Charles, Link prediction via matrix factorization, Machine Learning and Knowledge Discovery in Databases (2011), pp. 437–452

[13] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer (2009), **42**, pp. 30–37

[14] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, Advances in neural information processing systems (2008) **20**, pp. 1257–1264

[15] P. Symeonidis, A. Nanopoulos, Y. Manolopoulos, A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis, IEEE Transactions on Knowledge and Data Engineering (2010) **22**, pp. 179–192