Link Prediction Using Matrix Factorization with Bagging*

Zhifeng Wu
School of Information Technology and Engineering
Tianjin University of Technology and Education
Tianjin, China, 300222
zhifeng.wu@hotmail.com

Yixin Chen
Department of Computer Science and Engineering
Washington University in St. Louis
Saint Louis, MO, USA,63130
chen@cse.wustl.edu

Abstract —**Link prediction aims at estimating the likelihood of the existence of links between nodes. In this paper, we treat link prediction as a collaborative filtering problem, and propose an algorithm to solve this problem using matrix factorization approach. For making better predictions, this paper also explores the use of bagging technique as combination approaches for matrix factorization. We subsample the training set and added random noise to make multiple classifiers, and then combine these classifiers to be the final classifier. Results on several data sets show the efficacy of our approach. Compared with several popular proximity metrics, the accuracy of the algorithm can be increased greatly by use of bagging technique, especially measured by precision.**

Index Terms—link prediction, matrix factorization, bagging, collaborative filtering

I. INTRODUCTION

Link prediction, introduced by Liben-Nowell and Kleinberg [1], is a technique used to predict the missing links within a network or to predict the formation of links within a network in the future. Link prediction problem has become a hot topic in recent years and has been applied in many domains. For example, many problems in social networks and data mining can be modeled as a link prediction problem, such as the friends suggestion problem [2] in social network, the product recommendation problem [3] in online shopping system and the protein-protein interaction problem [4] in biological networks. Many algorithms have been proposed by researcher to address this issue. Most of these algorithms are based on similarity of nodes, statistical models and Markov chains.

In many cases, recommendation system is treated as a special case of link prediction. Some link prediction algorithms are adapted to solve recommendation system problem. For example, Z Huang etc. [5] presented six linkage measures adapted for recommendation and compared link prediction approaches with standard collaborative filtering algorithms. N Chiluka etc. [6] employed adaptations of popular link prediction algorithms for recommending items in User- Generated Content systems (UGCs). Liyan Zhang etc.[7] propose a recommendation algorithm which takes into account user's preference on item categories, and computes rank scores in different categories for each item. Xin Li and Hsinchun Chen[8] converted a recommendation problem into a link prediction problem by mapping transactions to a bipartite user-item interaction graph, where the graph structure captures subtle information on relations between users and items. Catherine A. Bliss etc. [9] provide an approach to predicting future links by applying the Covariance Matrix Adaptation Evolution Strategy(CMA-ES) to optimize weights in a large dynamic social networks. In contrast, link prediction also can be treated as a recommendation system, it is closely related to the problem of collaborative filtering. Recently, sparse matrix factorization also became popular for for link prediction task[10,11,12], which consider link prediction as a matrix completion problem and employ latent matrix factorization framework to learn latent feature factors for each object, and make predictions by taking appropriate inner products.

In this paper, first we treated the link prediction as the problem of collaborative filtering, where users and items are represented by nodes, and edges between nodes are treated like the rating in recommendation system according to the similarity score and then apply matrix factorization to predict which nodes are expected to connect. For collaborative filtering, several matrix factorization techniques have been successfully applied, including singular value decomposition[13], probabilistic matrix factorization [14], maximum margin matrix factorization [15], probabilistic latent semantic analysis [16]. we show that the conventional algorithm for the factorization of matrices has the potential to the link prediction. However, the results also show that it is unstable due to noise, bias and variance. In order to increase the stability, bagging technique is introduced in our approach. We have implemented bagging by using randomly generated subsets of the entire training set. In this work, moreover, we not only subsampled training set randomly, but also generated some noise for each sub-training set to prevent overfitting.

This work was jointly supported by the National Natural Science Foundation of China (No. 11172209), Natural Science Foundation of Tianjin(No. 14JCYBJC15400) and Tianjin University of Technology and Education Fund(No. RC14-51).

ICIS 2016, June 26-29, 2016, Okayama, Japan

The remainder of the paper is organized as follows: Section 2 will give a description of link prediction problem and the evaluation metrics; In Section 3, principle of matrix factorization will be explained in detail; We will show how to apply matrix factorization with bagging technique to predict linkage in Section 4; Seven real networks will be introduced in Section 5, and then comparison of the experimental results for several indices and our method on these real networks will be presented. The last section is a conclusion of the paper.

II. Link Prediction

Assumed G(V,E) denote an undirected network, where V is the set of nodes and E is the set of links, let U denotes a set of all links between all nodes in V. And no multiple links and self-connections in the network. For each pair of nodes, x, y $\in$ V , a score, denotes s_{xy} , is assigned. The score can be viewed as the similarity of nodes x and y. The score s_{xy} is more higher, x is more similar to y. The score can be calculated by different similarity measures. According to the score, all non-existent links in the network can be sorted in a descending order. The links at the top are most likely to exist.

For testing the accuracy of algorithm that used different similarity measures, the set of links, E, is randomly divided into two parts: the training set, ET, and the probe set EP. Training set is treated as known information, and probe set is used for testing and no information can be used for prediction. Clearly, $E = E^T \cup E^P$ and $E^T \cap E^P = \varnothing$. In this paper, the training set contains 90% of links, which is selected randomly from the network data set, and the probe set will be composed of the remaining 10% of links.

There are two standard metrics to quantify the accuracy of prediction algorithms: area under the receiver operating characteristic curve(AUC)[17] and Precision [18,19]. The AUC evaluates the performance of the metric according to the whole list of all non-observed links (U-E). AUC value can be interpreted as the probability of that --- a randomly chosen missing link (a link in E^P) has a higher score than a randomly chosen nonexistent link (a link in U-E).

In algorithmic implement, at each time a missing link and a nonexistent link be picked randomly to compare their score. Among n independent comparisons, if there are n' times that missing links have higher score than nonexistent links and n" times that they have the same score, the AUC can be figured as follow:

$$AUC = \frac{0.5n' + n''}{n} \tag{1}$$

According formula (1), AUC is from 0 to 1. Generally, AUC belongs to [0.5,1.0]. It indicates how much better the similarity measure index than the pure stochastic chosen. More exceeds 0.5, more better. If the AUC value is lower than 0.5, it means that this index is even worse than a random chosen for the missing links.

Precision is defined as the proportion of relevant items selected against the number of items selected. Generally, we select top L predicted links to evaluate this metric. If there are Lr right links, it means these links in the probe set, the precision can be figured as follow:

$$P_r = \frac{L_r}{L} \tag{2}$$

Obviously, higher precision means higher prediction accuracy.

III. Matrix Factorization

Generally, in a recommendation system, collaborative filtering problem can be defined in the following setting. The problem can be modeled by a triplet (U, I, R), where U is the user identifier which taking values from {1, . . . , N}, I is the item identifier which taking values from {1, . . . , M}, R is the rating value which taking values from X $\subset$ R. In practice, we can represent R as a partially specified matrix denoted by R · N · M ,where the matrix contains all the ratings that the users have assigned to the items.

For collaborative filtering problems, matrix factorization is one of the most often applied techniques. The idea behind matrix factorization techniques is very simple, there should be some latent features that determine how a user rates an item. Hence, if these latent features can be discovered, then we should be able to predict a rating with respect to a certain user and a certain item.

Suppose we want to discover K latent features, So we would like to find two matrices P and Q such that their product approximate the matrix R:

$$R \approx P \times Q^T = \hat{R}$$

where P is a N$\times$K and Q is a M$\times$K matrix. This factorization gives a low dimensional numerical representation of both users and items. Let P_{ik} denote the elements of $P \in R^{N \times K}$, and Q_{kj} the elements of $Q \in R^{M \times K}$. So, we can calculate the dot product of the two vectors corresponding to user i and item j:

$$\hat{r}_{ij} = p_i^T q_j = \sum_{k=1}^{K} p_{ik} q_{kj} \tag{3}$$

The squared error between the estimated rating and the real rating, can be calculated by the following equation for each user-item pair:

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 \tag{4}$$

In fact, for avoiding overfitting, regularization usually be applied. This is done by adding a parameter β and modify the squared error as follows:

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^{K} \left(\|P\|^2 + \|Q\|^2 \right) \quad (5)$$

Therefore, we can minimize the sum of the squared error of all rij in the training set to get the optimal matrices P* and Q*:

$$(P^*, Q^*) = \arg\min_{P,Q} \sum_{(i,j) \in T} e_{ij}^2 \quad (6)$$

To minimize the error, we need to know the gradient at the current values, and therefore we differentiate the equation with respect to these two variables separately:

$$\begin{aligned} \frac{\partial}{\partial p_{ik}} e_{ij}^2 &= -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij} q_{kj} + \beta p_{ik} \\ \frac{\partial}{\partial q_{kj}} e_{ij}^2 &= -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij} p_{ik} + \beta q_{kj} \end{aligned} \quad (7)$$

We update the weights in the direction opposite to the gradient:

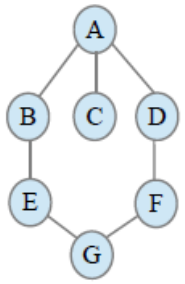
$$\begin{aligned} p'_{ik} &= p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha \left(2e_{ij} q_{kj} - \beta p_{ik} \right) \\ q'_{kj} &= q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha \left(2e_{ij} p_{ik} - \beta q_{kj} \right) \end{aligned} \quad (8)$$

Using the above update rules, we can then iterative perform the operation until the error converges to its minimum. Note, in equation (6) states that the optimal P and Q minimizes the sum of squared errors only on the known elements of R, that means that (i, j) is in the training set T.

IV. Predict Linkage Using Matrix Factorization with Bagging Technique

In this paper, we treat the link prediction as the problem of collaborative filtering, where users and items are represented by nodes, and edges between nodes are treated as the rating according to the similarity score. So, an undirected network G(V,E) can be modeled as a collaborative filtering problem model (U,I,R), which denoted by (V,V,R). Here, V is the set of nodes in the graph G. R also is the rating, which represented by the short path length of a pair of nodes with a max length k called trunc-path in this paper. For example, we consider the following matrix representation of the network of figure 1(a): the columns and rows of the matrix R both correspond nodes A to G.

Assume the trunc-path k is 2, if there is a path between node i to node j, and the length is less than or equal to 2, the corresponding cell has the value of the path length, otherwise it has the value of Infinity. This is illustrated in the Figure 1(b). In the matrix R, ij r represented the similarity of a pair of nodes (i, j), more smaller the value is, more similar a pair of nodes is. As we know, however, in collaborative filtering the rating is more bigger more similar. Therefore, we transform the matrix using the maximal value in the matrix minus each cell value. The new matrix is illustrated in Figure 1(c). For the new matrix, then we can use matrix factorization to estimate the unknown value between the pair of nodes. The pseudo-code of the algorithm is shown in Figure 2. The parameter α controls how large or small the update steps are relative to the current error of the estimation, the parameter β is used to control the magnitudes of the row and column nodes vectors to make P and Q give a good approximation R while containing only small numbers.



(a)

	A	B	C	D	E	F	G
A	0	1	1	1	2	2	-
B	1	0	2	2	1	-	2
C	1	2	0	2	-	-	0
D	1	2	2	0	-	1	2
E	2	1	-	-	0	2	1
F	2	-	-	1	2	0	1
G	-	2	-	2	1	1	0

(b)

	A	B	C	D	E	F	G
A	2	1	1	1	0	0	-
B	1	2	0	0	1	-	0
C	1	0	2	0	-	-	2
D	1	0	0	2	-	1	0
E	0	1	-	-	2	0	1
F	0	-	-	1	0	2	1
G	-	0	-	0	1	1	2

(c)

Fig. 1. An example of network and the corresponding matrix.

MatrixFactorization(R,K, α,β)
Input: R, matrix with n rows and n columns; K, number of features; α,learning rate; β,regularization factor
Output: P,Q*: the rows and columns feature matrices*

create P and Q matrices and initialize them randomly
loop until the ter min *al condition is met (P×Q approximate R well or the* max *imal number of iteration is reached)*
select a known element of R, denote it by r_{ij}

$$r'_{ij} = \sum_{k=1}^{K} p_{ik} \times q_{kj}$$
$$e_{ij} = r_{ij} - r'_{ij}$$
for$(k = 0; k < K; k++)$
$$p_{ik} = p_{ik} + \alpha \times \left(2 \times e_{ij} \times q_{kj} - \beta \times p_{ik} \right)$$
$$q_{kj} = q_{kj} + \alpha \times \left(2 \times e_{ij} \times p_{ik} - \beta \times q_{kj} \right)$$
end
end

Fig. 2.The pseudo-code of matrix factorization algorithm

Moreover, for increasing the accuracy of the algorithm, the bagging technique is applied in our approach. We subsample the training set several times randomly to make multiple different classifiers, then combine these classifiers to a new classifier by bagging technique. For each classifier, we randomly select about 75%-90% samples from the training set as a sub-training set. In addition, to avoid the local optimal, we also add 0.01%-0.1% random noises to the sub-training set. When the matrices of each classifier are calculated by the matrix factorization, the final matrix R can be calculated by average all sub-matrices. The algorithm is shown in figure 3. Finally, as most of link prediction algorithms, all non-existent links in the network be sorted in a descending order according to the value (score). The links at the top are most likely to exist.

Input: T, the training set;M,the number of iteration (classifiers)
Output: R. : the approximation of R (the similarity score of all non-existent links in the network

for each classifier:
create c R by sampling from T randomly and add a little random noise
$P_c, Q_c = MatrixFactorization(R_c, K, \alpha, \beta)$
$\hat{R}_c = P_c \times Q_c$
end
$\hat{R} = \frac{1}{M} \times \sum_{c=1}^{M} \hat{R}_c$

Fig. 3.The pseudo-code of matrix factorization algorithm with bagging technique

V. EXPERIMENTS

For testing our algorithm, we do some experiments on six real networks, and compare the result with those algorithms that calculate the similarity score by several proximity metrics. In this section, we will give a short description of the proximity metrics and the six real networks at beginning, and then show the results of these indices on the data sets.

A. Proximity Metrics and Data sets

We first provide some definitions and notation which will be useful to understand the descriptions below. Let $\Gamma(x)$ denote the set of neighbors of node x. Let |Q| be the cardinality of the set Q. Let kx be the degree of node x.

- Common Neighbors (CN)[20]. In common sense, if two nodes, x and y, have many common neighbors, they are more likely to form a link. The simplest measure of the neighborhood overlap is the directed count:
$$s_{xy} = \left|\Gamma(x) \cap \Gamma(y)\right|$$
- Salton Index [21]. This index is also called the cosine similarity . It is defined as
$$s_{xy} = \frac{\left|\Gamma(x) \cap \Gamma(y)\right|}{\sqrt{k_x \times k_y}}$$
- Jaccard Index [22]. This index was proposed by Jaccard over a hundred years ago, and is defined as
$$s_{xy} = \frac{\left|\Gamma(x) \cap \Gamma(y)\right|}{\left|\Gamma(x) \cup \Gamma(y)\right|}$$
- Leicht – Holme – Newman Index (LHN) [23]. It is defined as
$$s_{xy} = \frac{\left|\Gamma(x) \cap \Gamma(y)\right|}{k_x \times k_y}$$
- Preferential Attachment Index (PA)[24]. This index is defined as
$$s_{xy} = k_x \times k_y$$
- Adamic – Adar Index (AA) [25]. This index refines the simple counting of common neighbors by assigning the lessconnected neighbors more weights, and is defined as
$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z}$$

Additionally, we describe six representative public network data used in our experiments.

- US Air —the US air transportation network, which contains 332 airports and 2126 airlines[26] .
- Network Science(NS)—A network of publishing on the topic of networks [27].
- Power Grid (PG) — An electrical power grid of western US [28].
- Political Blogs (PB)— A network of the US political blogs [29].
- Neural network of Elegans(Cel) — This data set contains the neural network of the Caenorhabditis elegans worm (C.elegans)[30].
- Facebook-like Social Network (FB)—This data set is an online social network created from a virtual community for students at University of California, Irvine, in the period between April to October 2004[31].

The basic topological features of these networks are summarized in Table 1. N and M are the total numbers of nodes and links. C and r are the clustering coefficient [32] and assortative coefficient [33]. K is the average degree. L is the mean path length of a network. D is the diameter of a network.

TABLE I: THE BASE TOPOLOGICAL FEATURES

Nets	N	M	C	r	K	L	D
air	332	2126	0.5342	-0.2047	11.4578	2.842	8
cel	453	2025	0.5716	-0.2242	7.9956	2.664	8
fb	1899	13838	0.0939	-0.1880	13.0511	3.104	8
ns	1589	2742	0.5156	0.4513	3.0887	6.442	17
pb	1490	16715	0.2268	-0.2169	20.0725	2.781	8
pg	4941	6594	0.0630	-0.0116	2.3898	18.989	46

B. *Experimental Results*

In order to illustrate the potential of our algorithm for the link prediction problem, we empirically evaluated various prediction techniques based on six proximity metrics which be described in section V. Part A.

In these experiments, the trunc-path is set to 2 (for Power Grid data set it be set to 3). For the factorization algorithm, we set α =0.0002, β =0.02, the number of latent factors K=3 and we set the number of iterations to 2000. According to our observations, this amount of iterations were sufficient for the convergence of the algorithm. For bagging algorithm, the number of iterations (classifiers) is set to 15.

The algorithmic accuracies, measured by the AUC value and by Precision, on the six real networks are presented in Table II and Table III. Each result is obtained by averaging over 10 implementations with independently random partitions of testing set (90%) and probe set (10%). In the experiments, for AUC, n is 20 times size of probe set (we have tried to set n as 30 or 50 times size of probe set, and not found more impact on the results), for Precision, L is set to 100. The result of matrix factorization algorithm is denoted by MF. The result of matrix factorization algorithm with bagging technique is denoted by MFB. The entries corresponding to the highest accuracies among these indices are emphasized in black.

TABLE II: ACCURACIES OF ALOGRITHMS MEASURED BY AUC

index	Air	Cel	FB	NS	PB	PG
CN	0.94	0.93	0.77	0.94	0.93	0.60
SALT	0.91	0.81	0.73	0.94	0.90	0.60
JAC	0.90	0.78	0.74	**0.95**	0.90	0.60
LHN	0.77	0.73	0.72	0.94	0.83	0.56
PA	0.91	0.84	**0.90**	0.67	0.93	0.45
AA	**0.95**	**0.96**	0.78	0.94	0.93	0.60
MF	**0.95**	0.93	0.79	0.94	**0.94**	**0.64**
MFB	**0.95**	0.94	0.79	**0.95**	**0.94**	**0.64**

Compared with the other six measures introduced in section V. Part A, Table II shows that both MF and MFB indices have a better performance on most of given networks. Even though on Cel and FB networks where MF and MFB are not the best, the result is also better than other five indices, according to the AUC. From Table III, it can be seen that MFB index perform the best on all given networks by Precision measured except the NS data. On this network, MFB also shows well performance than the other five indices. MF index shows better than given six indices on Air, FB and PB networks by Precision measured. It is observed that, whether measured by AUC or Precision, the MFB index improved the accuracy of link prediction algorithms, and MF index performs well on most of networks.

Figure 4. shows the comparison of MF and MFB indices by precision measure on given data directly. From Table III. and Figure 4., it can be seen that the bagging technique have a great influence on the results. It improved the performance of algorithm greatly, especially on the precision measured.

TABLE III: ACCURACIES OF ALOGRITHMS MEASURED BY PRECISION

index	Air	Cel	FB	NS	PB	PG
CN	0.62	0.15	0.03	0.75	0.41	0.1
SALT	0.03	0.11	0	0.72	0	0
JAC	0.03	0.1	0	0.71	0	0
LHN	0	0.09	0	0.28	0	0.01
PA	0.5	0.11	0.08	0.01	0.09	0
AA	0.62	0.22	0.04	**0.97**	0.36	0.08
MF	0.65	0.12	0.15	0.49	0.51	0.05
MFB	**0.69**	**0.25**	**0.17**	0.79	**0.52**	**0.12**

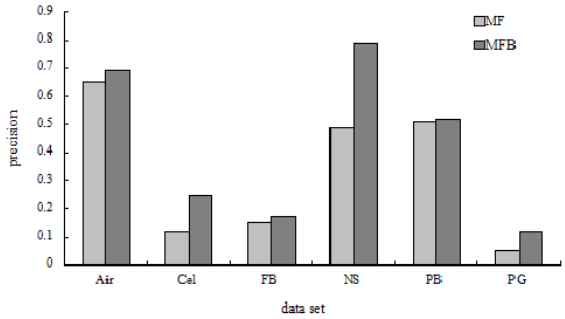


Fig. 4.MF vs MFB on Precision measure

VI. CONCLUSION

Link prediction problem is a long-standing challenge in modern information science, and a lot of algorithms have been presented. In this paper, we treated link prediction problem as a collaborative filtering, in which users and items instead by nodes and the trunc-path be treated as ratings. Then we introduced matrix factorization to address this problem. For improving the performance of the algorithm, bagging technique have been applied to our approach. We subsampled the training set and added random noise to make the classifiers, then combined the multiple classifiers to the final classifier. We proved the efficacy of our proposed algorithm through the link prediction task on several real world data sets. The experiments shown that our algorithm improved the quality of predictions, especially on the precision.

We only have considered unweighted networks in this paper. In the future, we will focus on the problem of weighted networks and on adapting the proposed method to handle weighted networks. At the same time, we also put our attention on link prediction for large scale network.

ACKNOWLEDGMENTS

This work was jointly supported by the National Natural Science Foundation of China (No. 11172209), Natural Science Foundation of Tianjin(No. 14JCYBJC15400) and Tianjin University of Technology and Education Fund(No. RC14-51).

REFERENCES

[1] David Liben-Nowell, Jon M. Kleinberg: The link prediction problem for social networks. CIKM 2003

[2] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. KDD’10, 2010.

[3] M. Li, B. M. Dias, I. Jarman, W. El-Deredy, and P. J. Lisboa. Grocery shopping recommendations based on basket-sensitive random walk. KDD’09.

[4] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic block models for relational data with application to protein-protein interactions,” Proceedings of Ineterational Biometric Society-ENAR Annual Meetings, 2006.

[5] Z. Huang, X. Li, and H. Chen, “Link prediction approach to collaborative filtering,” in Proc. JCDL’05, 2005, pp. 141–142

[6] N. Chiluka, N. Andrade and J. Pouwelse. A Link Prediction Approach to Recommendations in Large-Scale User-Generated Content Systems. Lecture Notes in Computer Science, 6611 (2011) 189.

[7] Liyan Zhang, Jie Xu, Chunping Li. A random-walk based recommendation algorithm considering item categories. Neurocomputing, 120 (2013) 391–396.

[8] Xin Li, Hsinchun Chen. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. Decision Support Systems 54 (2013) 880–890.

[9] Catherine A. Bliss, Morgan R. Frank, Christopher M. Danforth, etc. An evolutionary algorithm approach to link prediction in dynamic social networks. Journal of Computational Science 5 (2014) 750-764.

[10] A. Menon and C. Elkan. Link prediction via matrix factorization. In European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, 2011

[11] P. Hoff. Multiplicative latent factor models for description and prediction of socialnetworks. Computational & Mathematical Organization Theory, pages 261–272, 2009.

[12] Sheng Gao, Ludovic Denoyer, Patrick Gallinari: Link prediction via latent factor BlockModel. WWW (Companion Volume) 2012: 507-508

[13] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system—a case study. In Proc. Of WebKDD’00: Web Mining for E-Commerce Workshop, at 6th ACM SIGKDD Int. Conf. On Knowledge Discovery and Data Mining, Boston, MA,USA, 2000

[14] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, Advances in Neural Information Processing Systems 20, Cambridge, MA, 2008. MIT Press

[15] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. Advances in Neural Information Processing Systems, 17, 2005.

[16] T. Hofmann. Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst., 22(1):89–115, 2004.

[17] J.A. Hanely, B.J. McNeil, The meaning and use of the area under a receiver operating characteristic (ROC) curve, Radiology 143 (1982) 29.

[18] S. Geisser, Predictive Inference: An Introduction, Chapman and Hall, New York, 1993.

[19] J.L. Herlocker, J.A. Konstann, K. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (2004) 5.

[20] Newman, M. E., Clustering and Preferential Attachment in Growing Networks, Physical Review Letters E, Vol.64 (025102), 2001.

[21] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Auckland, 1983.

[22] P. Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, Bull. Soc. Vaud. Sci. Nat. 37 (1901) 547.

[23] E.A. Leicht, P. Holme, M.E.J. Newman, Vertex similarity in networks, Phys. Rev. E 73 (2006) 026120.

[24] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509.

[25] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Networks 25 (2003) 211.

[26] V. Batageli, A. Mrvar, Pajek Datasets, available at http://vlado.fmf.uni-lj.si/pub/networks/data/default.htm

[27] M. E. J. Newman, Phys. Rev. E 74, 036104 (2006)

[28] D. J. Watts, S. H. Strogatz, Nature 393, 440 (1998).

[29] R. Ackland, Mapping the US political blogosphere: Are conservative bloggers more prominent, Presentation to BlogTalk Downunder, Sydney, 2005

[30] Duch and A. Arenas, Community identification using Extremal Optimization J. Physical Review E , vol. 72, 027104, (2005).

[31] Opsahl, T., Panzarasa, P., 2009. Clustering in weighted networks. Social Networks 31 (2), 155-163, doi: 10.1016/j.socnet.2009.02.002

[32] D.J. Watts, S.H. Strogatz. Collective dynamics of 'small-world' networks. Nature 393, 440 (1998)

[33] M.E.J. Newman. Assortative mixing in networks. Phys. Rev. Lett. 89, 208701 (2002)