The Link Prediction Problem
in Bipartite Networks

Jérôme Kunegis, Ernesto W. De Luca, and Sahin Albayrak

DAI Lab, Technische Universität Berlin
Ernst-Reuter-Platz 7
D-10587 Berlin, Germany
{jerome.kunegis,ernesto.deluca,sahin.albayrak}@dai-labor.de

Abstract. We define and study the link prediction problem in bipartite networks, specializing general link prediction algorithms to the bipartite case. In a graph, a link prediction function of two vertices denotes the similarity or proximity of the vertices. Common link prediction functions for general graphs are defined using paths of length two between two nodes. Since in a bipartite graph adjacency vertices can only be connected by paths of odd lengths, these functions do not apply to bipartite graphs. Instead, a certain class of graph kernels (spectral transformation kernels) can be generalized to bipartite graphs when the positive-semidefinite kernel constraint is relaxed. This generalization is realized by the odd component of the underlying spectral transformation. This construction leads to several new link prediction pseudokernels such as the matrix hyperbolic sine, which we examine for rating graphs, authorship graphs, folksonomies, document–feature networks and other types of bipartite networks.

1   Introduction

In networks where edges appear over time, the problem of predicting such edges is called *link prediction* [1, 2]. Common approaches to link prediction can be described as *local* when only the immediate neighborhood of vertices is considered and *latent* when a latent model of the network is used. An example for local link prediction methods is the triangle closing model, and these models are conceptually very simple. Latent link prediction methods are instead derived using algebraic graph theory: The network's adjacency matrix is decomposed and a transformation is applied to the network's spectrum. This approach is predicted by several graph growth models and results in *graph kernels*, positive-semidefinite functions of the adjacency matrix [3].

Many networks contain edges between two types of entities, for instance item rating graphs, authorship graphs and document–feature networks. These graphs are called bipartite [4], and while they are a special case of general graphs, link prediction methods cannot be generalized to them. As we show in Section 2, this is the case for all link prediction functions based on the triangle closing model, as well as all positive-semidefinite graph kernels. Instead, we will see that

their odd components can be used, in Section 3. For each positive-semidefinite graph kernel, we derive the corresponding odd pseudokernel. One example is the exponential graph kernel $\exp(\lambda)$. Its odd component is $\sinh(\lambda)$, the hyperbolic sine. We also introduce the bipartite von Neumann pseudokernel, and study the bipartite versions of polynomials with only odd powers. We show experimentally (in Section 4) how these odd pseudokernels perform on the task of link prediction in bipartite networks in comparison to their positive counterparts, and give an overview of their relative performances . We also sketch their usage for detecting near-bipartite graphs.

2 Bipartite Link Prediction

The link prediction problem is usually defined on unipartite graphs, where common link prediction algorithms make several assumptions [5]:

- *Triangle closing*: New edges tend to form triangles.
- *Clustering*: Nodes tend to form well-connected clusters in the graph.

In bipartite graphs these assumptions are not true, since triangles and larger cliques cannot appear. Other assumptions have therefore to be used. While a unipartite link prediction algorithm technically applies to bipartite graphs, it will not perform well. Methods based on common neighbors of two vertices will for instance not be able to predict anything in bipartite graphs, since two vertices that would be connected (from different clusters) do not have any common neighbors.

Several important classes of networks are bipartite: authorship networks, interaction networks, usage logs, ontologies and many more. Many unipartite networks (such as coauthorship networks) can be reinterpreted as bipartite networks when edges or cliques are modeled as vertices. In these cases, special bipartite link prediction algorithms are necessary. The following two sections will review local and algebraic link prediction methods for bipartite graphs. Examples of specific networks of these types will be given in Section 4.

Definitions Given an undirected graph $G = (V, E)$ with vertex set V and edge set E, its adjacency matrix $A \in \mathbb{R}^{V \times V}$ is defined as $A_{uv} = 1$ when $(u, v) \in E$ and $A_{uv} = 0$ otherwise. For a bipartite graph $G = (V + W, E)$, the adjacency matrix can be written as $A = \begin{bmatrix} 0 & B; B^T & 0 \end{bmatrix}$, where $B \in \mathbb{R}^{V \times W}$ is the biadjacency matrix of G.

2.1 Local Link Prediction Functions

Some link prediction functions only depend on the immediate neighborhood of two nodes; we will call these functions local link prediction functions [1].

Let u and v be two nodes in the graph for which a link prediction score is to be computed. Local link prediction functions depend on the common neighbors of u

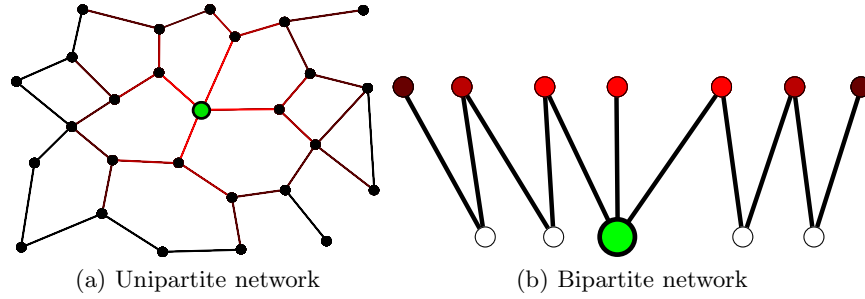(a) Unipartite network   (b) Bipartite network

Fig. 1. Link prediction by spreading activation in unipartite and bipartite networks. In the unipartite case, all paths are used. In the bipartite case, only paths of odd length need to be considered. In both cases, the weight of paths is weighted in inverse proportion to path length.

and v. In the bipartite link prediction problem, u and v are in different clusters, and thus have no common neighbors. The following link prediction functions are therefore not applicable to bipartite graphs: The number of common neighbors [1], the measure of Adamic and Adar [6] and the Jaccard coefficient [1]. These methods are all based on the *triangle closing* model, which is not valid for bipartite graphs.

Preferential Attachment Taking only the degree of u and v into account for link prediction leads to the *preferential attachment* model [7], which can be used as a model for more complex methods such as modularity kernels [8,9].

If $d(u)$ is the number of neighbors of node u, the preferential attachment models gives a prediction between u and v of $d(u)d(v)/(2|E|)$. The factor $1/(2|E|)$ normalizes the sum of predictions for a vertex to its degree.

3 Algebraic Link Prediction Functions

Link prediction algorithms that not only take into account the immediate neighborhood of two nodes but the complete graph can be formulated using algebraic graph theory, whereby a decomposition of the graph's adjacency matrix is computed [10]. By considering transformations of a graph's adjacency matrix, link prediction methods can be defined and learned. Algebraic link prediction methods are motivated by their scalability and their learnability. They are scalable because they rely on a model that is built once and which makes computation of recommendations fast. These models correspond to decomposed matrices and can usually be updated using iterative algorithms [11]. In contrast, local link prediction algorithms are *memory-based*, meaning they access the adjacency data directly during link prediction. Algebraic link prediction methods are learnable because their parameters can be learned in a unified way [12].

In this section, we describe how algebraic link prediction methods apply to bipartite networks. Let $G = (V, E)$ be a (not necessarily bipartite) graph. Algebraic link prediction algorithms are based on the eigenvalue decomposition of its adjacency matrix A:

$$A = U \Lambda U^T$$

To predict links, a *spectral transformation* is usually applied:

$$F(A) = U F(\Lambda) U^T$$

where $F(\Lambda)$ applies a real function $f(\lambda)$ to each eigenvalue λ_i. $F(A)$ then contains link prediction scores that, for each node, give a ranking of all other nodes, which is then used for link prediction. If $f(\lambda_i)$ is positive, F is a graph kernel, otherwise, we will call F a pseudokernel.

Several spectral transformations can be written as polynomials of the adjacency matrix in the following way. The matrix power A^i gives, for each vertex pair (u, v), the number of paths of length i between u and v. Therefore, a polynomial of A gives, for a pair (u, v), the sum of all paths between u and v, weighted by the polynomial coefficients. This fact can be exploited to find link prediction functions that fulfill the two following requirements:

- The link prediction score should be higher when two nodes are connected by *many* paths.
- The link prediction score should be higher when paths are *short*.

These requirements suggest the use of polynomials f with decreasing coefficients.

3.1   Odd Pseudokernels

In bipartite networks, only paths of odd length are significant, since an edge can only appear between two vertices if they are already connected by paths of odd lengths. Therefore, only odd powers are relevant, and we can restrict the spectral transformation to odd polynomials, i.e. polynomials with odd powers.

The resulting spectral transformation is then an odd function and except in the trivial and undesired case of a constant zero function, will be negative at some point. Therefore, all spectral transformations described below are only pseudokernels and not kernels.

The Hyperbolic Sine In unipartite networks, a basic link prediction function is given by the matrix exponential of the adjacency matrix [13–15]. The matrix exponential can be derived by considering the sum

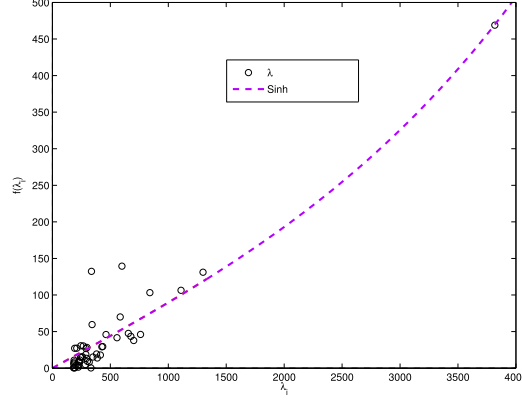$$\exp(\alpha A) = \sum_{i=0}^{\infty} \frac{\alpha^i}{i!} A^i$$

Fig. 2. In this curve fitting plot of the Slovak Wikipedia, the hyperbolic sine is a good match, indicating that the hyperbolic sine pseudokernel performs well.

where coefficients are decreasing with path length. Keeping only the odd component, we arrive at the matrix hyperbolic sine [16].

$$\sinh(\alpha A) = \sum_{i=0}^{\infty} \frac{\alpha^{1+2i}}{(1+2i)!} A^{1+2i}$$

Figure 2 shows the hyperbolic sine applied to the (positive) spectrum of the bipartite Slovak Wikipedia user–article edit network.

The Odd von Neumann Pseudokernel The von Neumann kernel for unipartite graphs is given by the following expression [13].

$$K_{\mathrm{NEU}}(A) = (I - \alpha A)^{-1} = \sum_{i=0}^{\infty} \alpha^i A^i$$

We call its odd component the odd von Neumann pseudokernel:

$$K_{\mathrm{NEU}}^{\mathrm{odd}}(A) = \alpha A (I - \alpha^2 A^2)^{-1} = \sum_{i=0}^{\infty} \alpha^{1+2i} A^{1+2i}$$

The hyperbolic sine and von Neumann pseudokernels are compared in Figure 3, based on the path weights they produce.
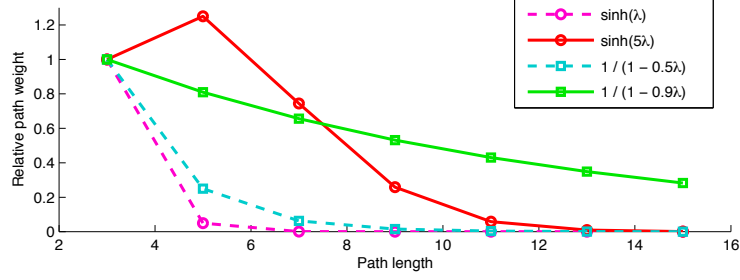
Fig. 3. Comparison of several odd pseudokernels: the hyperbolic sine and the odd von Neumann pseudokernel. The relative path weight is proportional to the corresponding coefficient in the Taylor series expansion of the spectral transformation.

Rank Reduction Similarly, rank reduction of the matrix A can be described as a pseudokernel. Let λ_k be the eigenvalue with k-th largest absolute value, then rank reduction is defined by

$$f(\lambda) = \begin{cases} \lambda & \text{if } |\lambda| \geq |\lambda_k| \\ 0 & \text{otherwise} \end{cases}$$

This function is odd, but does not have an (odd) Taylor series expansion.

3.2 Computing Latent Graph Models

Bipartite graphs have adjacency matrices of the form

$$A = \begin{pmatrix} & B \\ B^T & \end{pmatrix}$$

where B is the biadjacency matrix of the graph. This form can be exploited to reduce the eigenvalue decomposition of A to the equivalent singular value decomposition $B = \tilde{U}\Sigma\tilde{V}$.

$$A = \begin{pmatrix} U & U \\ V & -V \end{pmatrix} \begin{pmatrix} +\Sigma & \\ & -\Sigma \end{pmatrix} \begin{pmatrix} U & U \\ V & -V \end{pmatrix}^T$$

with $U = \tilde{U}/\sqrt{2}$, $V = \tilde{V}/\sqrt{2}$ and each singular value σ corresponds to the eigenvalue pair $\{\pm\sigma\}$.

3.3 Learning Pseudokernels

The hyperbolic sine and the von Neumann pseudokernel are parametrized by α, and rank reduction has the parameter k, or equivalently λ_k. These parameters can be learned by reducing the spectral transformation problem to a one-dimensional curve fitting problem, as described in [12]. In the bipartite case, we can apply the curve fitting method to only the graph's singular value, since odd spectral transformations fit the negative eigenvalue in a similar way they fit the positive eigenvalues. This kernel learning method is shown in Figure 4.
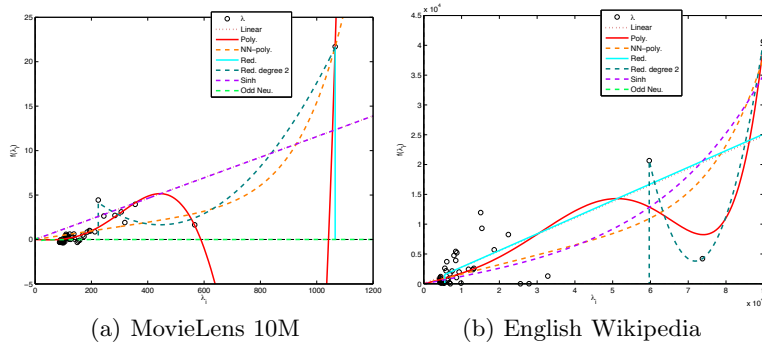
(a) MovieLens 10M　　　　　　　　(b) English Wikipedia

Fig. 4. Learning a pseudokernel that matches an observed spectral transformation in the MovieLens 10M rating network and English Wikipedia edit history.

4　Experiments

As experiments, we show the performance of bipartite link prediction functions on several large datasets, and present a simple method for detecting bipartite or near-bipartite datasets.

4.1　Performance on Large Bipartite Networks

We evaluate all bipartite link prediction functions on the following bipartite network datasets. BibSonomy is a folksonomy of scientific publications [17]. BookCrossing is a bipartite user–book interaction network [18]. CiteULike is a network of tagged scientific papers [19]. DBpedia is the semantic network of relations extracted from Wikipedia, of which we study the five largest bipartite relations [20]. Epinions is the rating network from the product review site Epinions.com [21]. Jester is a user–joke network [22]. MovieLens is a user–movie rating dataset, and a folksonomy of tags attached to these movies [23]. Netflix is the large user–item rating network associated with the Netflix Prize [24]. The Wikipedia edit graphs are the bipartite user–article graphs of edits on various language Wikipedias. The Wikipedia categories are represented by the bipartite article–category network [25]. All datasets are bipartite and unweighted. In rating datasets, we only consider the presence of a rating, not the rating itself. Table 1 gives the number of nodes and edges in each dataset.

In the experiments, we withhold 30% of each network's edges as the test set to predict. For datasets in which edges are labeled by timestamps, the test set consists of the newest edges. The remaining training set is used to compute link prediction scores using the preferential attachment model and the pseudokernel learning methods described in the previous sections. For the pseudokernel learning methods, the training set is again split into 70% / 30% subsets for training. Link prediction accuracy is measured by the mean average precision (MAP), averaged over all users present in the test set [26]. The evaluation results are summarized in Table 1.

Table 1. Overview of datasets and experiment results. See the text for a description of the datasets and link prediction methods. Link prediction methods: Poly: odd polynomials, NN-poly: odd nonnegative polynomials, Sinh: hyperbolic sine, Red: rank reduction, Odd Neu: odd von Neumann pseudokernel, Pref: preferential attachment.

Dataset	Nodes	Edges	Poly.	NN-poly.	Sinh	Red.	Odd Neu.	Pref.
BibSonomy tag-item	975,963	2,555,080	0.921	**0.925**	**0.925**	0.782	0.917	0.924
BibSonomy user-item	777,084	2,555,080	0.748	0.771	0.771	0.645	0.750	**0.821**
BibSonomy user-tag	210,467	2,555,080	0.801	0.820	0.820	0.777	0.295	**0.878**
CiteULike tag-item	885,046	2,411,819	0.593	0.608	0.608	0.510	0.635	**0.698**
CiteULike user-item	754,484	2,411,819	0.853	**0.856**	**0.856**	0.735	0.855	0.838
CiteULike user-tag	175,992	2,411,819	0.812	0.836	0.836	0.782	0.202	**0.881**
DBpedia artist-genre	47,293	94,861	0.824	**0.971**	0.833	0.736	0.841	0.961
DBpedia birthplace	191,652	273,695	0.952	0.977	**0.978**	0.733	0.813	0.968
DBpedia football club	41,846	131,084	**0.685**	0.678	0.674	0.505	0.159	0.680
DBpedia starring	83,252	141,942	0.908	0.916	**0.924**	0.731	0.570	0.897
DBpedia work-genre	156,145	222,517	0.879	0.941	0.908	0.746	0.867	**0.966**
Epinions	876,252	13,668,320	0.644	**0.690**	0.546	0.501	0.061	**0.690**
French Wikipedia	3,989,678	41,392,490	0.667	0.744	0.744	0.654	0.108	**0.803**
German Wikipedia	3,357,353	51,830,110	0.673	0.699	0.699	0.651	0.156	**0.799**
Japanese Wikipedia	1,892,869	18,270,562	0.740	0.752	0.755	0.618	0.076	**0.776**
Jester	25,038	616,912	0.575	0.571	**0.581**	0.461	0.579	0.501
MovieLens 100k	2,625	100,000	**0.822**	0.774	0.738	0.718	0.631	0.812
MovieLens 10M	136,700	10,000,054	**0.683**	0.682	0.663	0.500	0.298	0.680
MovieLens 1M	9,746	1,000,209	0.640	**0.662**	0.538	0.500	0.221	**0.662**
MovieLens tag-item	24,129	95,580	0.860	0.860	0.860	0.737	**0.865**	0.863
MovieLens user-item	11,610	95,580	0.755	0.741	0.728	0.659	0.674	**0.812**
MovieLens user-tag	20,537	95,580	0.782	0.798	0.798	0.672	0.663	**0.915**
Netflix	497,959	100,480,507	**0.674**	0.671	0.670	0.500	0.322	0.672
Spanish Wikipedia	2,684,231	23,392,353	0.634	0.750	0.750	0.655	0.094	**0.799**
Wikipedia categories	2,036,440	3,795,796	0.591	0.659	0.663	0.500	0.589	**0.675**

4.2   Detecting Near-bipartite Networks

Some networks are not bipartite, but nearly so. An example would be a network of "fan" relationships between persons where there are clear "hubs" and "authorities", i.e. popular persons and persons being fan of many people. While these networks are not strictly bipartite, they are mostly bipartite in a sense that has to be made precise. Measures for the level of bipartivity exist in several forms [4, 27], and spectral transformations offer another method. Using the link prediction method described in Section 3.3, nearly bipartite graphs can be recognized by the *odd* shape of the learned curve fitting function.

Figure 5 shows the method applied to two uniparite networks: the Advogato trust network [28] and the hyperlink network in the English Wikipedia [25]. The curves indicate that the Advogato trust network is *not* bipartite, while the Wikipedia link network is nearly so.

5   Discussion

While technically the link prediction problem in bipartite graphs is a subproblem of the general link prediction problem, the special structure of bipartite graphs makes common link prediction algorithms ineffective. In particular, all methods based on the triangle closing model cannot work in the bipartite case. Out of
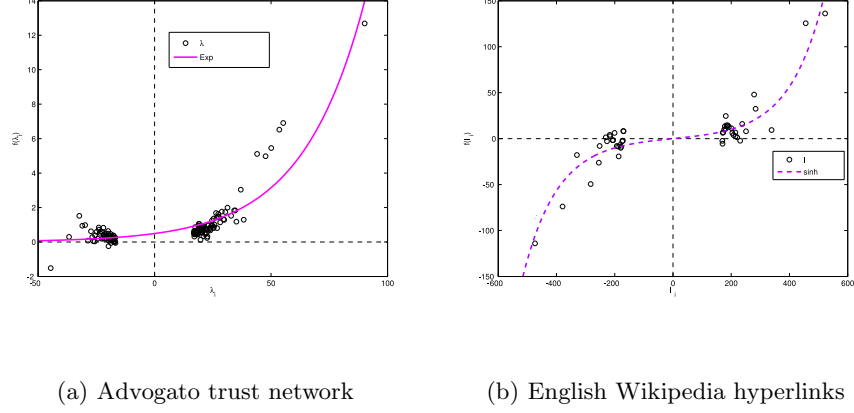
(a) Advogato trust network (b) English Wikipedia hyperlinks

Fig. 5. Detecting near-bipartite and non-bipartite networks: If the hyperbolic sine fits, the network is nearly bipartite; if the exponential fits, the network is not nearly bipartite. (a) the Advogato trust network, (b) the English Wikipedia hyperlink network. These graphs show the learned transformation of a graph's eigenvalues; see the text for a detailed description.

the simple local link prediction methods, only the preferential attachment model can be used in bipartite networks.

Algebraic link prediction methods can be used instead, by restricting spectral transformations to odd functions, leading to the matrix hyperbolic sine as a link prediction function, and an odd variant of the von Neumann kernel. As in the unipartite case, no single link prediction method is best for all datasets.

References

1. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proc. Int. Conf. on Information and Knowledge Management. (2003) 556–559
2. Taskar, B., Wong, M.F., Abbeel, P., Koller, D.: Link prediction in relational data. In: Advances in Neural Information Processing Systems. (2003)
3. Gärtner, T., Horváth, T., Le, Q.V., Smola, A., Wrobel, S.: Kernel Methods for Graphs. In: Mining Graph Data. John Wiley & Sons (2006)
4. Holme, P., Liljeros, F., Edling, C.R., Kim, B.J.: On network bipartivity. Phys. Rev. E **68** (2003) 6653–6673
5. Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic evolution of social networks. In: Proc. Int. Conf. on Knowledge Discovery and Data Mining. (2008) 462–470
6. Adamic, L., Adar, E.: Friends and neighbors on the web. Social Networks **25** (2001) 211–230
7. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439) (1999) 509–512

8. Zhang, D., Mao, R.: Classifying networked entities with modularity kernels. In: Proc. Conf. on Information and Knowledge Management. (2008) 113–122
9. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74** (2006)
10. Chung, F.: Spectral Graph Theory. American Mathematical Society (1997)
11. Rendle, S., Schmidt-Thieme, L.: Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In: Proc. Int. Conf. on Recommender Systems. (2008) 251–258
12. Kunegis, J., Lommatzsch, A.: Learning spectral graph transformations for link prediction. In: Proc. Int. Conf. on Machine Learning. (2009) 561–568
13. Ito, T., Shimbo, M., Kudo, T., Matsumoto, Y.: Application of kernels to link analysis. In: Proc. Int. Conf. on Knowledge Discovery in Data Mining. (2005) 586–592
14. Wu, Y., Chang, E.Y.: Distance-function design and fusion for sequence data. In: Proc. Int. Conf. on Information and Knowledge Management. (2004) 324–333
15. Kandola, J., Shawe-Taylor, J., Cristianini, N.: Learning semantic similarity. In: Advances in Neural Information Processing Systems. (2002) 657–664
16. Cardoso, J.R., Leite, F.S.: Computing the inverse matrix hyperbolic sine. In: Proc. Int. Conf. on Numerical Analysis and its Applications. (2001) 160–169
17. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: BibSonomy: A social bookmark and publication sharing system. In: Proc. Workshop on Conceptual Structure Tool Interoperability. (2006) 87–102
18. Ziegler, C.N., McNee, S.M., Konstan, J.A., Lausen, G.: Improving recommendation lists through topic diversification. In: Proc. Int. World Wide Web Conf. (2005) 22–32
19. Emamy, K., Cameron, R.: CiteULike: A researcher's social bookmarking service. Ariadne (51) (2007)
20. Bizer, C., Cyganiak, R., Auer, S., Kobilarov, G.: DBpedia.org–querying Wikipedia like a database. In: Proc. Int. World Wide Web Conf. (2007)
21. Massa, P., Avesani, P.: Controversial users demand local trust metrics: an experimental study on epinions.com community. In: Proc. American Association for Artificial Intelligence Conf. (2005) 121–126
22. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval **4**(2) (2001) 133–151
23. GroupLens Research: MovieLens data sets. http://www.grouplens.org/node/73 (October 2006)
24. Bennett, J., Lanning, S.: The Netflix prize. In: Proc. KDD Cup. (2007) 3–6
25. Wikimedia Foundation: Wikimedia downloads. http://download.wikimedia.org/ (January 2010)
26. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
27. Estrada, E., Rodríguez-Velázquez, J.A.: Spectral measures of bipartivity in complex networks. Phys. Rev. E **72** (2005)
28. Stewart, D.: Social status in an open-source community. American Sociological Review **70**(5) (2005) 823–842