On the Scalability of Graph Kernels Applied to Collaborative Recommenders

Jérôme Kunegis, Andreas Lommatzsch, Şahin Albayrak [1] and **Christian Bauckhage** [2]
{kunegis, andreas, sahin}@dai-lab.de, christian.bauckhage@telekom.de

Abstract. We study the scalability of several recent graph kernel-based collaborative recommendation algorithms. We compare the performance of several graph kernel-based recommendation algorithms, focussing on runtime and recommendation accuracy with respect to the reduced rank of the subspace. We inspect the exponential and Laplacian exponential kernels, the resistance distance kernel, the regularized Laplacian kernel, and the stochastic diffusion kernel. Furthermore, we introduce new variants of kernels based on the graph Laplacian which, in contrast to existing kernels, also allow negative edge weights and thus negative ratings. We perform an evaluation on the Netflix Prize rating corpus on prediction and recommendation tasks, showing that dimensionality reduction not only makes prediction faster, but sometimes also more accurate.

1 Introduction

In information retrieval, the task of filtering and recommending items to users is often done in a content-based manner [1]. Collaborative filtering, by contrast, bases item rankings on ratings collected from users [13].

A collaborative filtering system therefore usually consists of a database of users, items (such as text documents or movies), and a collection of ratings that users have assigned to these items.

Collaborative rating databases are modeled as bipartite graphs, where users and items are represented by means of nodes, and the ratings by means of labeled edges. Recently, kernels have been used to tackle the task of recommendation. Graph kernels in particular are based on the rating database's underlying bipartite graph model. Traditionally, this type of kernel only copes with positively rated links. For a recommendation application this is a drawback since users may want to express their dislike and therefore may also assign negative ratings. In this paper, we therefore introduce kernels for collaborative rating prediction which also cope with negative ratings.

In order to scale to the size of current rating databases, collaborative recommendation algorithms must be able to process very large rating corpora. This necessitates some form of dimensionality reduction. Of course, dimensionality reduction will influence prediction accuracy and runtime. In this paper, we study the performance of collaborative recommendation algorithms in combination with dimensionality reduction.

Our contributions are therefore as follows: First, we study the prediction accuracy of graph kernels for recommendation on signed data, as opposed to unsigned data. Second, we introduce signed variants of all graph kernels that are based on the graph Laplacian. Third, we evaluate the influence of dimensionality reduction on the recommendation accuracy and runtime performance for both training and prediction.

Next, we first review related work; we then introduce terms and definitions used in collaborative filtering. The different graph kernels we consider in this paper are presented in the third section and the fourth section describes how to apply dimensionality reduction to each of these kernels. Afterwards, we discuss the application of graph kernels to recommendation algorithms. Finally we present and discuss our experimental evaluation.

2 Related Work

For a general description of collaborative rating prediction, we refer the reader to [2]. Graph kernels have been used for collaborative recommendations in [3]. The authors of [6] apply kernels to link analysis. In this study, the underlying graph is weighted by only positive values.

The matrix exponential has been used outside of computer science for sociometric analysis [8], and has been rediscovered for collaborative filtering recently [12].

Dimensionality reduction for collaborative filtering is discussed in [14]. This previous work however only apply dimensionality reduction to the adjacency matrix of the bipartite rating graph, without using graph kernels. This reference gives values for the optimal reduced rank between 5 and 15. We will refer to this method as *simple dimensionality reduction*.

As computation of dense kernels is too expensive in the case of large rating databases, sparse methods have to be employed, which in the general case scale much better than dense methods [4].

3 Definitions

Throughout this paper, we assume U to be the set of users where $|U| = m$, likewise we assume I to be the set of items where $|I| = n$. The rating database is represented by means of a sparse matrix $R \in \mathbb{R}^{m \times n}$ whose number of nonzero elements is denoted by r.

[1] DAI-Labor, Technische Universität Berlin, Germany
[2] Telekom Laboratories, Berlin, Germany

The sparse matrix R corresponds to a weighted bipartite rating graph $G = (V, E, W)$ where $V = U \cup I$ is the set of vertices and E the set of edges. For every rating $R_{ui} \neq 0$, E contains an edge (u, i), the corresponding edge weights are given by $W_{ui} = R_{ui}$. Since we want to consider positive and negative ratings, we do not restrict W to nonnegative values.

The adjacency matrix $A \in \mathbb{R}^{(m+n)\times(m+n)}$ of G is given by $A = \begin{bmatrix} & R \\ R^T & \end{bmatrix}$ and we also define a diagonal degree matrix D whose entries D_{ii} contain the sum of adjacent edge weights of the corresponding node i. The graph Laplacian is then given by $L = D - A$.

For our extension to the case of negative ratings, we also define the *absolute degree matrix* $\bar{D}$, which is a diagonal matrix, too, and contains the sum of *absolute edge weights* for each node. Analoguously, we define $\bar{L} = \bar{D} - A$.

4 Graph Kernels

In this section we present the kernels evaluated in this paper. For all except one of these, we introduce new variants in order to be able to deal with signed rating data. We also describe dimensionality reduction in itself, which is not a kernel but can be used in place of a kernel.

All kernels are based on the following observation: If $K \in \mathbb{R}^{V\times V}$ is a symmetric matrix, then the following function d is a dissimilarity matrix: $d(i, j) = K_{ii} + K_{jj} - K_{ij} - K_{ji}$. Its square root is an Euclidean metric in the space spanned by the eigenvectors of K, and its inverse is a similarity measure between any two nodes [9]. In the next paragraphs, we give expressions for the matrix K for the various kernels.

Rank reduced adjacency matrix kernel. The adjacency matrix A itself may be interpreted as a kernel, because if two nodes are similar (positively or negatively) they will be connected by an edge. However, in order to derive predictions for items that were not rated so far, this kernel is only useful after a rank reduction has been applied. We simply set

$$K_{\mathrm{DIM}} = A$$

Exponential diffusion kernel. [10] defines the exponential diffusion kernel using the matrix exponential:

$$K_{\mathrm{EXP}} = \exp(\alpha A) = \textstyle\sum_{i=0}^{\infty} \frac{1}{i!}\alpha^i A^i$$

Since A^n contains the number of paths of length n between any two nodes, this kernel represents an average of path counts between nodes, weighted by the inverse factorial of path length. Therefore, longer connections are less influential than shorter ones.

Resistance distance kernel. This kernel is also called the commute time kernel. It results from interpreting the graph G as a network of electrical resistances with resistance values given by the edge weights W. Given a pair of nodes, the total resistance induced by the network is a distance given by the following kernel [16]:

$$K_{\mathrm{RES}} = L^+ = (D - A)^+$$

where L^+ denotes the Moore-Penrose pseudoinverse of the Laplacian matrix.

In order to also account for negative edge weights, we define a signed resistance distance kernel [11]

$$K_{\mathrm{RES-S}} = \bar{L}^+ = (\bar{D} - A)^+$$

where we apply the absolute degree and Laplacian matrix as defined in the previous section.

Stochastic diffusion kernel. This kernel is based on a stochastic diffusion process and hence only applies to positive data [10].

$$K_{\mathrm{STO}} = (1 - \alpha)(I - \alpha D^{-1}A)^{-1}$$

The parameter α denotes the probability in the diffusion process of following a graph edge instead of returning to the starting node. The matrix $D^{-1}A$ is a stochastic diffusion matrix, and this kernel is therefore designed for positive data.

As with the resistance distance kernel, this kernel is a new variant of the stochastic diffusion kernel which also takes into account negative ratings

$$K_{\mathrm{STO-S}} = (1 - \alpha)(I - \alpha \bar{D}^{-1}A)^{-1}$$

Laplacian exponential diffusion kernel. The Laplacian exponential diffusion kernel applies the matrix exponential to the Laplacian [3, 15]:

$$K_{\mathrm{LEX}} = \exp(-\alpha L) = \exp(-\alpha(D + A))$$

We found this kernel to perform poorly in practice. However, a signed version leads to acceptable results.

$$K_{\mathrm{LEX-S}} = \exp(-\alpha \bar{L}) = \exp(-\alpha(\bar{D} + A))$$

In the evaluation, we will only use the signed Laplacian exponential diffusion kernel.

Regularized Laplacian kernel. This kernel is a generalization of the random forest kernel [3].

$$K_{\mathrm{REL}} = (I + \alpha(\gamma D - A))^{-1}$$

The random forest kernel itself is based on random forest models [3]. It arises in the calculation of weighted counts of forests of G in which two nodes belong to the same tree.

$$K_{\mathrm{FOR}} = (I + L)^{-1}$$

We do not show the random forest kernel in the evaluation because it performs similarly to the regularized Laplacian kernel. We also define a signed variant of the regularized Laplacian kernel:

$$K_{\mathrm{REL-S}} = (I + \alpha(\gamma \bar{D} - A))^{-1}$$

We only use the signed regularized Laplacian kernel for evaluation, as it performs much better than the unsigned variant.

All these kernels are based on matrix inversion or exponentiation and cannot be computed directly.

5 Dimensionality Reduction

Given the huge but sparse adjacency matrix A, any computation of graph kernels will require dimensionality reduction.

If $A = Q\Lambda Q^T$ denotes the eigenvalue decomposition of the symmetric matrix A, a rank-k approximation of A is given by $\tilde{A} = Q_k\Lambda_k Q_k^T$, where $k \ll m + n$ is the desired rank and Q_k and Λ_k denote the corresponding truncations of Q and Λ.

This kind of dimensionality reduction is also known as latent semantic analysis and is frequently used for projecting high-dimensional data into lower dimensional spaces.

In order to apply this reduction to a kernel K, we observe that all kernels which we presented in the previous section can be expressed as $K = Qf(\Lambda)Q^T$, where Q and Λ are given by the eigenvalue decomposition of the a linear combination of the matrices A and D. The function $f(\Lambda)$ depends on the individual characteristics of the kernel. Note that $f(\Lambda)$ can be computed efficiently because it only only has to be applied to the diagonal matrix Λ. In the kernels we consider in this paper, three different types of $f(\Lambda)$ occur: Matrix inversion, the Moore-Penrose pseudoinverse, and the matrix exponential.

The rank reduced kernels are then computed as in the following example: If $\alpha A = Q\Lambda Q^T$ then $\tilde{K}_{\mathrm{EXP}} = Q_k \exp(\Lambda_k)Q_k^T$.

We also note that the truncation mode depends on the operation performed. For inversion and pseudo-inversion, we must retain the eigenvalues closest to zero, excluding zero eigenvalues for the pseudoinverse. For the matrix exponential, we retain the biggest eigenvalues and corresponding eigenvectors. For simple dimensionality reduction, we retain the eigenvalues with biggest absolute value.

6 Recommendation and Prediction

In this section, we describe the basic rating prediction algorithm, and the rating prediction algorithms based on graph kernels.

A common preprocessing step in collaborative filtering is to normalize the rating data. Normalization can be user-based or item-based [5]. For user-based normalization, each user's nonzero ratings are scaled to zero mean and unit variance, but zero entries of A remain unchanged.

In our implementation, we use a hybrid of user-based and item-based normalization. Given a rating r, the normalized rating $\hat{r}$ is computed using the user's and item's mean rating and rating standard deviation:

$$\hat{r} = (2r - \mu_u - \mu_i)/(\sigma_u + \sigma_i) \quad (1)$$

Once a rating has been predicted based on the normalized rating matrix, it has to be scaled back to the user's original range of ratings by inverting Equation (1).

Given a user u and item i and ignoring normalization, the baseline user-based rating prediction algorithm [13] proceeds as follows:

1. Retrieve all ratings of item i according to other users.
2. Compute the average over these ratings, weighted by the correlation between the other users and user u.

To predict ratings using a kernel, the correlation step in this algorithm is replaced by computing the similarity measure induced by the kernel. Since collaborative filtering considers user-user or item-item similarities, we consider distinct kernels for the sets of users and items respectively.

Recommendation is implemented by predicting ratings for all possible items, and choosing the items with the highest rating prediction, in function of the number of items searched.

7 Evaluation

We use the Netflix Prize corpus of ratings[3] for evaluation. Out of the whole corpus, we use a subset of 3,216 users, 1,307 items and 57,507 ratings. The corresponding rating matrix is filled to 1.37%. The test and training samples were drawn at random from the complete rating set.

We measure the accuracy of rating prediction using the root mean squared error (RMSE) which is the square root of the average over all squared differences between the actual and the predicted rating. This precedure is standard in the collaborative filtering literature [2].

the accuracy of recommendation is given by the normalized discounted cumulated gain (nDCG), as defined in [7].

[3] `http://www.netflixprize.com/`

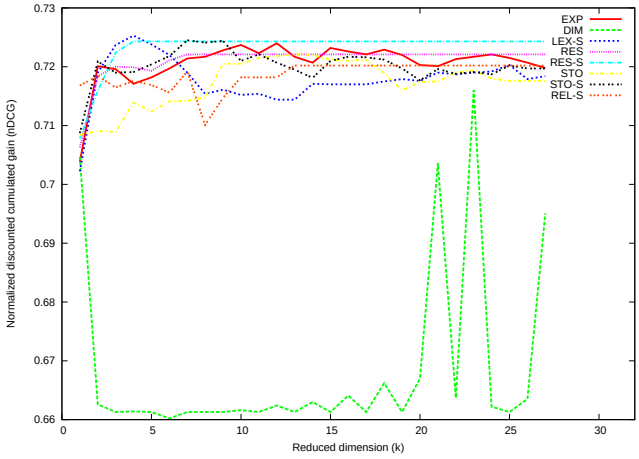

Figure 1. Comparison of recommendation accuracy in function of the reduced rank k for all kernels. This figure shows the normalized discounted cumulated gain (nDCG). Higher values denote better recommendation.

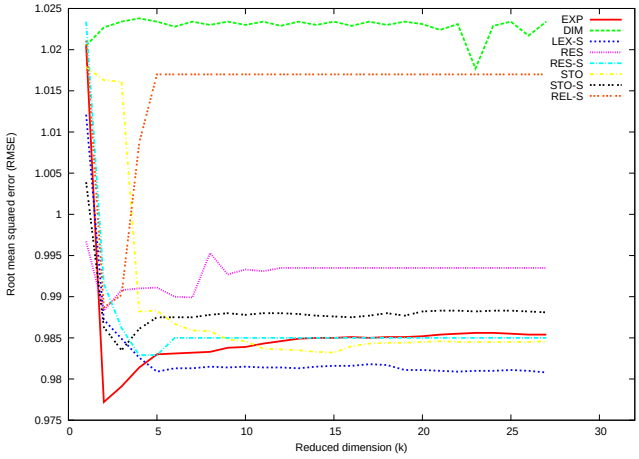

Figure 2. Comparison of rating prediction error in function of the reduced rank k for all kernels. This figure shows the root mean squared error (RMSE). Lower values denote more accurate rating prediction.

For testing the scalability of the different kernel methods, we varied the parameter k from 1 to 27 for each kernel. Figure 1 shows the nDCG in function of k for all kernels in the recommendation task. Higher nDCG values denote more accurate recommendations. Figure 2 shows the RMSE in function of k for all kernels in the prediction task. Lower RMSE values denote more precise predictions.

Asymptotic behavior. We observe two different patterns of asymptotic behavior. Some kernels attain their best performance asymptotically for big k, while others reach an optimum for a specific value of k. Table 1 summarizes these findings. We note that most kernels (EXP, REL-S, RES, RES-S, STO-S) show an inverted behavior on the recommendation task than on the prediction task. Although simple dimension-

ality reduction shows isolated peaks for specific values of k, there is no recognizable pattern for this kernel.

Table 1. Classification of graph kernels by their asymptotic behavior. The left column groups the kernels attaining their best performance at a specific, small value of k. The right column contains the kernels having asymptotic optimal behavior for big k. Kernels showing neither behavior are omitted.

	$k_{\text{best}} = k_0$	$k_{\text{best}} = +\infty$
Recommender	DIM	EXP, RES, RES-S STO, STO-S, REL-S
Prediction	EXP, RES, RES-S STO-S, REL-S	LEX, STO

Choice of k. Algorithms with asymptotically optimal performance reach their almost-optimum for $k = 5$. The other kernels peak between $k = 2$ and $k = 5$. Both observations suggest that a value $k > 5$ is not needed for this size of corpus.

Stability. At the task of rating prediciton, all kernels perform smoothly in function of k. At recommendation, only the resistance distance and regularized Laplacian kernels perform smoothly. The other kernels' performances vary much more with changing k. We must therefore recommend the resistance distance and regularized Laplacian kernels as their results are more predictable and consistent.

Good recommender but bad predictor. We observe that the regularized Laplacian kernel shows acceptable recommendation accuracy, but bad prediction accuracy except for a small peak at $k = 2, 3$. We interpret this performance as a correctly ranked prediction which however does not match the actual values.

Simple dimensionality reduction. Dimensionality reduction itself performs worse than all proper kernels as expected. Also, the accuracy of simple dimensionality reduction seems to oscillate between better performance for even k and worse performance for odd k. We explain this by the fact that the spectrum of A contains pairs of eigenvalues $\pm\lambda$ because the rating graph is bipartite. Apparently, using only one of these two eigenvalues and its respective eigenvector leads to lower accuracy.

Laplacian vs adjacency matrix. We observe that kernels based on the graph Laplacian perform better than kernels based on the adjacency matrix. The resistance distance kernel, which corresponds to the inverted Laplacian, is definitely better than simple dimensionality reduction, and Laplacian exponential kernel, while not more accurate than the exponential kernel, has more stable behavior for changing k.

Signed kernels better than unsigned. The signed variants all performed better than the unsigned counterparts. The near-exception are the stochastic diffusion kernels on the prediction task, where the unsigned variant is more accurate asymptotically. However, the overall peak is reached by the signed variant at $k = 3$.

Overall best kernel. For the choice of overall best kernel, we select the signed resistance distance and signed Laplacian exponential kernel. The exponential kernel comes close but has worse stability for varying k, making it difficult to recommend in practice.

8 Conclusion and Future Work

In this paper, we studied the prediction accuracy of collaborative recommender and rating prediction algorithms based on graph kernels. We considered eight different kernels, including three novel, signed variants.

We found that small reduced ranks are acceptable in most cases depending on the rating corpus and that kernels based on the graph Laplacian are usually better than kernels based on the adjacency matrix. Also, we showed that dimensionality reduction not only reduction the runtime but also makes collaborative recommenders more accurate. We also showed that most kernels can be used in the context of signed rating data, when new *signed* kernel variants are used.

REFERENCES

[1] Justin Basilico and Thomas Hofmann, 'Unifying collaborative and content-based filtering', in *Proc. Int. Conf. on Machine learning*, p. 9. ACM Press, (2004).

[2] John S. Breese, David Heckerman, and Carl Kadie, 'Empirical analysis of predictive algorithms for collaborative filtering', in *Proc. Conf. Uncertainty in Artificial Intelligence*, pp. 43–52, (1998).

[3] François Fouss, Luh Yen, Alain Pirotte, and Marco Saerens, 'An experimental investigation of graph kernels on a collaborative recommendation task', in *Proc. Int. Conf. on Data Mining*, pp. 863–868, (2006).

[4] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, October 1996.

[5] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl, 'An algorithmic framework for performing collaborative filtering', in *Proc. Int. Conf. on Research and Development in Information Retrieval*, pp. 230–237, (1999).

[6] Takahiko Ito, Masashi Shimbo, Taku Kudo, and Yuji Matsumoto, 'Application of kernels to link analysis', in *Proc. Int. Conf. on Knowledge Discovery in Data Mining*, pp. 586–592, (2005).

[7] Kalervo Järvelin and Jaana Kekäläinen, 'Cumulated gain-based evaluation of ir techniques', *ACM Trans. Inf. Syst.*, **20**(4), 422–446, (2002).

[8] Leo Katz, 'A new status index derived from sociometric analysis', *Psychometrika*, **18**(1), 39–43, (March 1953).

[9] D. J. Klein and M. Randić, 'Resistance distance', *Journal of Mathematical Chemistry*, **12**(1), 81–95, (1993).

[10] R. Kondor and J. Lafferty, 'Diffusion kernels on graphs and other discrete structures', in *Proc. Int. Conf. on Machine Learning*, pp. 315–322, (2002).

[11] Jérôme Kunegis and Stephan Schmidt, 'Collaborative filtering using electrical resistance network models with negative edges', in *Proc. Industrial Conf. on Data Mining*, pp. 269–282. Springer-Verlag, (2007).

[12] Joel C. Miller, Gregory Rae, Fred Schaefer, Lesley A. Ward, Thomas LoFaro, and Ayman Farahat, 'Modifications of kleinberg's hits algorithm using matrix exponentiation and web log records', in *Proc. Int. Conf. on Research and Development in Information Retrieval*, pp. 444–445, (2001).

[13] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, 'GroupLens: An Open Architecture for Collaborative Filtering of Netnews', in *Proc. Conf. on Computer Supported Cooperative Work*, pp. 175–186, (1994).

[14] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, 'Incremental svd-based algorithms for highly scalable recommender systems', in *Proc. Int. Conf. on Computer and Information Technology*, pp. 399–404, (2002).

[15] A. Smola and R. Kondor, 'Kernels and regularization on graphs', in *Proc. Conf. on Learning Theory and Kernel Machines*, pp. 144–158, (2003).

[16] F. Y. Wu, 'Theory of resistor networks: The two-point resistance', *Journal of Physics A*, **37**, 6653–6673, (2004).