Alternative Similarity Functions for Graph Kernels

Jérôme Kunegis, Andreas Lommatzsch
DAI-Labor
kunegis, andreas@dai-labor.de

Christian Bauckhage
Deutsche Telekom Laboratories
christian.bauckhage@telekom.de

Abstract

Given a bipartite graph of collaborative ratings, the task of recommendation and rating prediction can be modeled with graph kernels. We interpret these graph kernels as the inverted squared Euclidean distance in a space defined by the underlying graph and show that this inverted squared Euclidean similarity function can be replaced by other similarity functions. We evaluate several such similarity functions in the context of collaborative item recommendation and rating prediction, using the exponential diffusion kernel, the von Neumann kernel, and the random forest kernel as a basis. We find that the performance of graph kernels for these tasks can be increased by using these alternative similarity functions.

1. Introduction

In the domain of collaborative filtering, the task of rating prediction consists of predicting missing entries in a database of ratings between users and items. Based on a sparse collection of ratings given by users to items, the prediction of unknown ratings can be achieved by using *kernels*, functions that give a similarity between user pairs or items pairs. Recent work has seen the application of *graph kernels*, which are based on the bipartite graphical model underlying the rating dataset. While the various known graph kernels define similarity functions that can be used for collaborative rating prediction, they also define a new user or item space. In this space, traditional kernels such as the Gaussian function or the inverted squared Euclidean distance can be used.

We show in this paper that the application of such traditional kernels on top of graph kernels can increase the accuracy of collaborative rating prediction. First, we give an overview of several kernels that can be applied on top of graph kernels. Then, we analyse their rating prediction performance in function of their variance parameter and in function of the underlying graph kernel.

The rest of this paper is organized as follows. In Section 2, we present the collaborative filtering task and baseline algorithm. Section 3 defines the three graph kernels we use. The next section defines and motivates the similarity functions. The experimental evaluation of our setup is performed in Section 5, and we conclude in Section 6.

2. Collaborative Filtering

The collaborative recommender systems we consider consist of users, items, and ratings given by users to items. The ratings are collected in a sparse rating matrix.

We use the following definitions. U is the set of users of size m and I the set of items of size n. The sparse rating matrix is denoted by $R \in \mathbb{R}^{m \times n}$. $A \in \mathbb{R}^{(m+n)\times(m+n)}$ is the adjacency matrix of the underlying bipartite graph, and the $D \in \mathbb{R}^{(m+n)\times(m+n)}$ the degree matrix defined by $d_{ii} = \sum_j r_{ij}$ when i is a user, and analoguously when i is an item. The graph Laplacian is given by $L = D - A$.

Collaborative rating prediction for the pair (u, i) is usually implemented by averaging over known ratings for item i by other users, weighting the average by the similarity between the other users and u [4]. This approach is called user-based rating prediction. Equivalently, item-based rating prediction is implemented by averaging of user u's ratings of other items. The flexibility of these approaches lies in the choice of similarity function that is used for weighting. Early algorithms used the Pearson correlation and cosine distance between two user's ratings. Later work has applied various kernels as weights, including graph kernels as presented in the next section.

3. Graph Kernels

Most similarity functions used for collaborative filtering are *kernels* in that they can be interpreted as the inner product in a suitably defined space of users [3, 7].

Specifically, *graph kernels* are kernel functions based on the interpretation of the rating matrix as a bipartite graph, having users and items as vertex classes.

A review of various graph kernels can be found in [3]. In this paper, we use the random forest kernel, the von Neumann kernel, and the exponential kernel. Other graph kernels have proven to give very similar results to these three for our task.

All three kernels can be defined by a dissimilarity matrix $K \in \mathbb{R}^{(m+n)\times(m+n)}$. A kernel function $k(i,j)$ is then defined by $k(i,j) = 1/(K_{ii} + K_{jj} + K_{ij} + K_{ji})$. In the graph kernels we cover, the dissimilarity matrix K is symmteric and positive semi-definite. It can therefore be written as $K = QDQ^T$, where Q is orthogonal and D is non-negative and diagonal, giving the decomposition $K_{ij} = U_i U_i^T$ with $U = QD^{1/2}$. We then find the distance $d(i,j) = k(i,j)^{-1/2}$ using

$$\begin{aligned} d(i,j)^2 &= K_{ii} + K_{jj} - K_{ij} - K_{ji} && (1)\\ &= (U_i - U_j)^2 && (2)\end{aligned}$$

showing that graph kernels are indeed inversed squared Euclidean distances.

We now give the corresponding K for the three graph kernels we review. The random forest kernel is based on random forest models [5]. It arises in the calculation of weighted counts of forests of the rating graph in which two nodes belong to the same tree [2].

$$K_{\mathrm{FOR}} = (I + L)^{-1} \qquad (3)$$

The exponential diffusion kernel [7] is based on the matrix exponential.

$$K_{\mathrm{EXP}} = \exp(\alpha A) = \sum_{i=0}^{\infty} \frac{1}{i!}\alpha^i A^i \qquad (4)$$

This kernels represents an average of path counts between nodes, weighted by the inverse factorial of path length.

The von Neumann diffusion kernel is similar to the exponential diffusion kernel in that it represents a weighted sums of powers of the adjacency matrix A; it differs in that it uses exponentially decreasing weights [3].

$$K_{\mathrm{NEU}} = (I - \alpha A)^{-1} = \sum_{i=0}^{\infty} \alpha^k A^k \qquad (5)$$

For the sake of completeness, we also mention the variant $K = (I - \alpha A)^{-1} - I$ which is discussed in [6], but which we do not evaluate in this paper, because we found it to perform very similarly to the original Von Neumann kernel.

The next section will show how inversion of the squared Euclidean distance can be replaced by other functions that give a similarity.

4. Similarity Functions

In the previous section, graph kernels were shown to correspond to the inverted squared Euclidean distance in a specific space defined by the corresponding kernel.

To simplify the notation we will omit the indices i and j for the functions d and k.

The similarity based on graph kernels can be described as the inverted squared Euclidean distance defined by K. The inverted squared Euclidean distance $k = 1/d^2$ has two features: It is unbounded for small d, and it decreases as a rational function for large d. We propose the following similarity functions in place of the inverted squared Euclidean:

$$\begin{aligned} k_{\mathrm{Eu}} &= \frac{\sigma^2}{d^2} && (6)\\ k_{\mathrm{EuA}} &= 1/\left(1 + \frac{d^2}{\sigma^2}\right) && (7)\\ k_{\mathrm{Ga}} &= \exp\left\{-\frac{1}{2}\frac{d^2}{\sigma^2}\right\} && (8)\end{aligned}$$

The parameter σ is a scale parameter. For convenience, we recast the inverted squared Euclidean using this parameter σ as k_{Eu}. The two other similarity functions are motivated as follows.

k_{EuA} is similar to the inverted squared Euclidean distance. Instead of being unbounded for small ds, it has a global maximum at zero. We will call it the adjusted Euclidean similarity function.

k_{Ga} is the Gaussian function, and is motivated by the frequent use of Gaussian kernels in machine learning. As k_{EuA}, is has a maximum at zero. Unlike k_{Eu} and k_{EuA}, it decreases exponentially for large d, giving more weight to very similar users.

We note that these similarity functions can be interpreted as *kernels* themselves. However, to avoid confusion, we will restrict the term *kernel* to graph kernels.

Figures 1 and 2 show all three similarity functions in function of the distance d and in function of the inverted squared Euclidean distance respectively.

It can be seen that the Gaussian function corresponds to a double sigmoid function applied to the inverted squared Euclidean distance. For comparison, we also show the logistic sigmoid function $k_S = \tanh(\sigma^2/d^2)$.

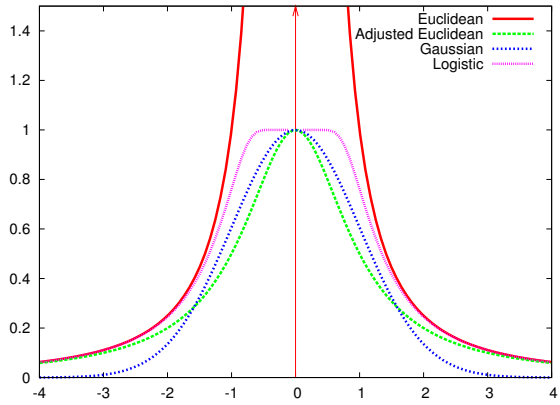

Figure 1. Similarities in function of the Euclidean distance d

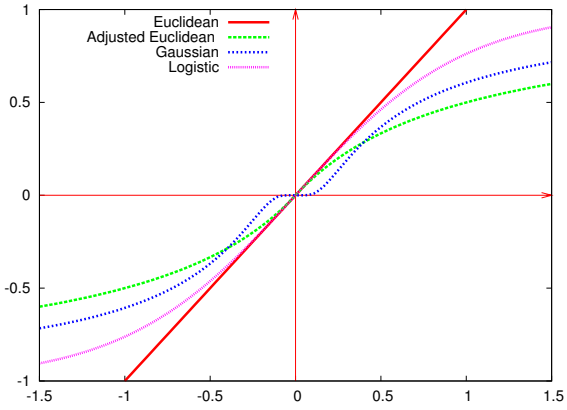

Figure 2. Similarities in function of the inverted Euclidean distance $1/d$

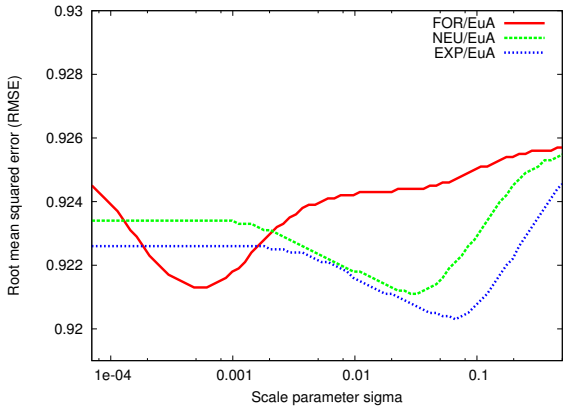

Figure 3. The accuracy of rating prediction using the Euclidean similarity function for all three kernels

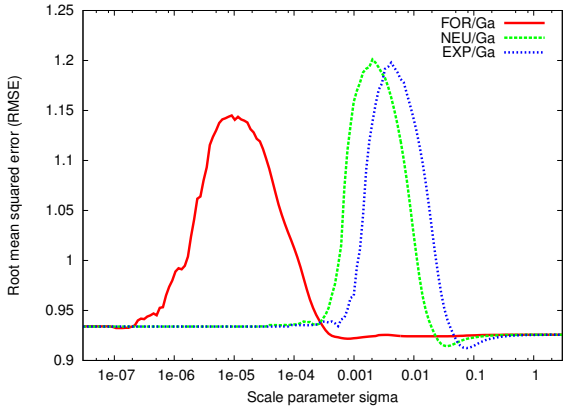

Figure 4. The accuracy of rating prediction using the Gaussian similarity function for all three kernels

5. Evaluation

In this section, we show the performance of recommendation and rating prediction algorithms using all similarity functions.

We use the Netflix Prize corpus of ratings[1]. 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%.

We implement all kernels as defined above and refer to them using using the abbreviations introduced there.

We measure the accuracy of rating prediction using the root mean squared error (RMSE) which is the average over all absolute differences between the actual and the predicted rating. This precedure is standard in the collaborative filtering literature [1], and is also the procedure used on the Netflix Prize website.

Figures 3 and 4 show the performance of the Euclidean and Gaussian similarity functions respectively. Figure 5 shows a comparison of all similarity functions, focussing on the greatest prediction accuracy range.

Table 1 shows for each graph kernel and similarity function combination, the best, worst and asymptotic performance in function of the parameter σ. The asymptotic performance for $\sigma \to 0$ corresponds to the performance of the baseline algorithms using simple inverted squared Euclidean distance. For comparison, the root mean squared error of the baseline algorithm using the Pearson correlation as weights is 0.9322.

We make the following observations.

[1] http://www.netflixprize.com/

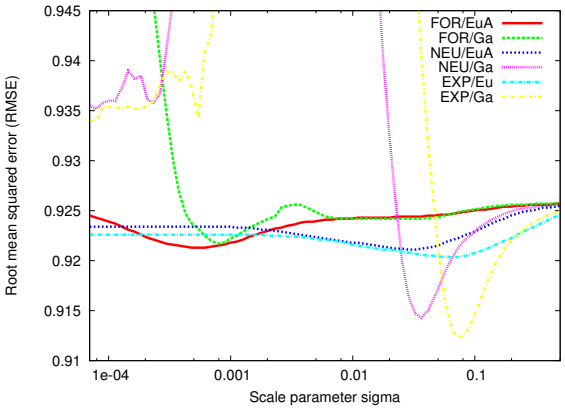

Figure 5. Comparison of all similarity functions and graph kernels, restricted to the range where the biggest accuracy is attained

		Best	Worst	$\sigma \to 0$	$\sigma \to \infty$
FOR	EuA	0.9213	0.9265	0.9264	0.9265
	Ga	0.9217	1.1450	0.9337	0.9256
NEU	EuA	0.9211	0.9257	0.9233	0.9256
	Ga	0.9143	1.2010	0.9337	0.9256
EXP	EuA	0.9203	0.9257	0.9225	0.9256
	Ga	0.9123	1.1978	0.9337	0.9256

Table 1. Overall comparison of all graph kernels and all similarity functions. For every combination, we show the best and worst performance, and the asymptotic performance for small and big σ.

Best similarity function. The overall best performance is achieved by the Gaussian similarity function. We explain this by noting that the Gaussian kernel emphasizes the weight of users that are very similar. This suggests that the Euclidean similarity function gives too much weight to other users that are not similar to the active user.

Behavior in function of σ. All similarity functions show a global minimum error for a finite value of σ. The adjusted Euclidean function performs only by a small amount better than the regular inverted Euclidean function. The Gaussian similarity performs worse than average for a specific range of σ values. As expected, the asymptotic performance corresponds to the baseline case.

Best graph kernels. In the comparison of the three graph kernels, we observe that the exponential diffusion kernels performs best. The von Neumann kernel has similar but worse performance, and the random forest kernels gives the least accurate predictions.

6. Conclusion

We showed that graph kernels as used in collaborative filtering systems can be interpreted as the inverted squared Euclidean distance in a space defined by the underlying graph. We used that fact to replace the inverted squared Euclidean function by other similarity functions. We showed that the performance of collaborative recommendation and rating prediction algorithms can be increased by using such alternative similarity functions.

For future work, other graph kernels may lead to different performance for the various similarity functions. We have also been experimenting with versions of graph kernels that are suited for *signed* graph data. We expect them to perform well generally, but have not tried them in conjunction with alternative similarity functions.

References

[1] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. Conf. Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[2] P. Chebotarev and E. Shamis. The matrix-forest theorem and measuring relations in small social groups. *Automation and Remote Control 58*, 9:1505, 1997.

[3] F. Fouss, L. Yen, A. Pirotte, and M. Saerens. An experimental investigation of graph kernels on a collaborative recommendation task. In *Proc. Int. Conf. on Data Mining*, pages 863–868, 2006.

[4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2):133–151, 2001.

[5] T. Ito, M. Shimbo, T. Kudo, and Y. Matsumoto. Application of kernels to link analysis. In *Proc. Int. Conf. on Knowledge Discovery in Data Mining*, pages 586–592, 2005.

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

[7] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proc. Int. Conf. on Machine Learning*, pages 315–322, 2002.