BIOINFORMATICS

Vol. 20 Suppl. 1 2004, pages i326–i333
DOI: 10.1093/bioinformatics/bth906

Learning kernels from biological networks by maximizing entropy

Koji Tsuda[1,2]* and William Stafford Noble[3,4]

[1]Max Planck Institute for Biological Cybernetics, Spemannstrasse 38, 72076 Tübingen, Germany, [2]Computational Biology Research Center, National Institute of Advanced Industrial Science and Technology (AIST), 2-43 Aomi Koto-ku, Tokyo, Japan, [3]Department of Genome Sciences and [4]Department of Computer Science, 1705 NE Pacific Street, University of Washington, Seattle, WA 98109, USA

Received on January 15, 2004; accepted on March 1, 2004

ABSTRACT

Motivation: The diffusion kernel is a general method for computing pairwise distances among all nodes in a graph, based on the sum of weighted paths between each pair of nodes. This technique has been used successfully, in conjunction with kernel-based learning methods, to draw inferences from several types of biological networks.

Results: We show that computing the diffusion kernel is equivalent to maximizing the von Neumann entropy, subject to a global constraint on the sum of the Euclidean distances between nodes. This global constraint allows for high variance in the pairwise distances. Accordingly, we propose an alternative, locally constrained diffusion kernel, and we demonstrate that the resulting kernel allows for more accurate support vector machine prediction of protein functional classifications from metabolic and protein–protein interaction networks.

Availability: Supplementary results and data are available at noble.gs.washington.edu/proj/maxent

Contact: koji.tsuda@tuebingen.mpg.de

1 INTRODUCTION

Many types of genomic data can be usefully represented using networks. In such a network, nodes represent genes or proteins, and edges may represent physical interaction of the proteins (Schwikowski *et al*., 2000; Uetz *et al*., 2000; von Mering *et al*., 2002), gene regulatory relationships (Lee *et al*., 2002; Ihmels *et al*., 2002; Segal *et al*., 2003), edges in a metabolic pathway, similarities between protein sequences (Yona *et al*., 1999), etc. These networks provide tools for visualizing and understanding biological phenomena, as well as a framework for predicting new network edges or predicting protein function, localization, etc. This paper describes a general method for deriving a Euclidean embedding of genes or proteins from any type of biological network. The embedding may be used to perform various types of visualization and inference tasks.

*To whom correspondence should be addressed.

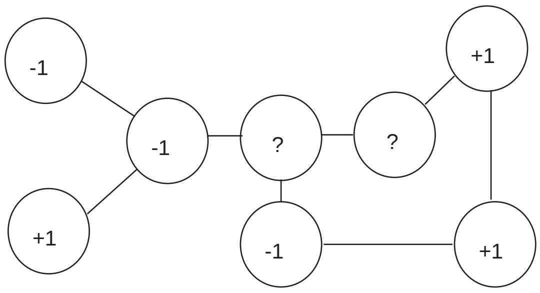

Fig. 1. A schematic illustration of the function prediction problem on a protein network. According to the existence or non-existence of a specific function, annotated proteins are labeled by either '$+1$' or '-1'. The task is to predict labels of unannotated proteins marked as '?'. This particular network is undirected, but some networks (e.g. metabolic or regulatory networks) would have directed edges.

For concreteness, we focus here on the problem of predicting protein function using a protein network as input. This problem has previously been addressed using majority vote (Schwikowski *et al*., 2000; Hishigaki *et al*., 2001), graph-based (Vazquez *et al*., 2003), Bayesian (Deng *et al*., 2003) and discriminative learning methods (Vert and Kanehisa, 2003; Lanckriet *et al*., 2004). The problem can be described as a two-class classification problem on an undirected graph (Fig. 1). According to the existence of a specific function, annotated proteins are labeled either $+1$ or -1. When predicting the label of an unannotated protein, one relies on some notion of 'closeness' or 'distance' between nodes. For example, majority vote methods simply use the shortest path distance, and count the number of positive samples in the neighborhood. However, we need a robust distance measure in order to achieve high prediction performance, because the shortest path distance is sensitive to the insertion or deletion of individual edges.

Inferring closeness among the nodes of a network is an extremely ill-posed problem. When there are n samples, our task is to find an $n \times n$ symmetric similarity matrix from an undirected graph. A biological network represents local proximity relationships between samples (proteins or genes); i.e. two samples are likely to share some property when connected by an edge. Hence, with respect to each edge, we have prior knowledge that connected samples should be close to each other. However, the number of free parameters in the similarity matrix is $n(n+1)/2$, which is much larger than the number of edges.

Moreover, it is preferable that the resulting similarity matrix be a valid kernel matrix so that function prediction can be performed by the support vector machines (SVMs) or other high-performance kernel classifiers (Noble, 2004; Schölkopf and Smola, 2002). Technically, a kernel matrix is required to be positive definite. The positive definiteness ensures that these samples can implicitly be embedded into a Euclidean space (called the feature space). Another important feature of the kernel representation is that multiple matrices representing heterogeneous data can be combined easily, e.g. by summing the kernel matrices (Pavlidis *et al*., 2001; Lanckriet *et al*., 2004; Vert and Kanehisa, 2003; Tsuda *et al*., 2003).

Maximum entropy methods have been proved to be effective for solving general ill-posed problems (Wu, 1997). However, maximum entropy methods are mostly concerned with the estimation of a probability distribution, not a kernel matrix. In this paper, we generalize the maximum entropy framework to estimate a positive definite kernel matrix. A kernel matrix is identified by maximizing the von Neumann entropy subjected to a set of constraints derived from a network. The von Neumann entropy is a natural extension of the Shannon entropy to positive definite matrices, which is commonly used in quantum physics (Nielsen and Chuang, 2000).

Based on this general theory, kernel matrices are constructed from biological networks. In the process, constraints on the kernel matrix must be imposed, which amount to specifying the geometry of the embedded samples in the feature space. We propose the following two types of constraints derived from the biological network: (i) global constraint: the sum of Euclidean distances between connected samples is upper-bounded by a constant and (ii) local constraints: the Euclidean distance between each pair of connected samples is upper-bounded by a smaller constant.

The maximum entropy kernel from the global constraint exactly corresponds to the diffusion kernel (Kondor and Lafferty, 2002), which has been used successfully for making predictions from biological networks (Lanckriet *et al*., 2004; Vert and Kanehisa, 2003). When combined with SVMs, the diffusion kernel has already been shown to work quite well on yeast function prediction (Lanckriet *et al*., 2004); i.e. the prediction accuracy was better than the accuracy of a Bayesian network (Deng *et al*., 2003).

However, one drawback inherent in the diffusion kernel is that, in the feature space, the distances between connected samples have high variance. This means that some pairs of connected samples are particularly close to one another, and others are not. In practice, this high variance leads to scaling problems in the feature space: some distances are so large that many of the others are effectively zero. This outcome is obviously inappropriate, because all the edges have been determined by the same experimental procedure.

We show that the maximum entropy kernel based on local constraints resolves this scaling problem and thereby shows better accuracy in yeast function prediction. We call this a 'locally constrained diffusion kernel'. In experiments based on metabolic pathways (Goto *et al*., 2002) and protein–protein interaction networks (von Mering *et al*., 2002), our new kernel outperforms the diffusion kernel significantly.

2 MAXIMUM ENTROPY LEARNING OF KERNELS

Maximum entropy learning (or estimation) is a common approach for solving ill-posed problems that involves estimating a probability distribution on a continuous or discrete domain (Wu, 1997). Given a set of constraints, one is required to pick one distribution from the set of distributions satisfying all the constraints. A natural choice is to pick the distribution with maximum entropy, which corresponds to the most 'smooth' distribution. With linear constraints, it is well known that the optimal solution is described as a Gibbs distribution (Wu, 1997). The maximum entropy method aims to take the most 'neutral' distribution, which contains no unwanted structure. This section describes how this approach is extended for learning a kernel matrix.

2.1 Kernels and distances on a graph

SVMs work by embedding samples into a vector space called a feature space, and searching for a linear discriminant function in such a space (Schölkopf and Smola, 2002). In our case, the n nodes in a graph are mapped to n points in the feature space $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in F$. The embedding is defined implicitly by specifying an inner product via a positive definite kernel matrix $K_{ij} = \boldsymbol{x}_i^\top \boldsymbol{x}_j, i, j = 1, \ldots, n$. Because the discriminant function is solely represented by inner products, we do not need to have an explicit representation of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$. Once a kernel matrix is determined, the (squared) Euclidean distance between two points can also be computed as $D_{ij} := \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = K_{ii} + K_{jj} - 2K_{ij}$.

Assume that a set of linear constraints $\mathrm{tr}(KU_j) \leq 0, j = 1, \ldots, m$ are given as prior knowledge, where U_j is an $n \times n$ symmetric matrix. In our case, such constraints are derived from biological networks (Section 3). Because the scale of K does not affect the performance of the SVM, let us constrain $\mathrm{tr}(K) = 1$. Now the problem is formulated as choosing one kernel matrix that satisfies all these constraints. However, because the number of edges is usually small, there are infinitely many choices. Such a problem is called an ill-posed problem (Wu, 1997), where the solution cannot be determined uniquely from the information at hand.

2.2 von Neumann entropy

In quantum physics, the notion of entropy has been generalized to positive definite matrices with trace one (Nielsen and Chuang, 2000). The entropy of a positive definite matrix K is defined as

$$E(K) = -\mathrm{tr}(K \log K), \quad K \succ 0, \ \mathrm{tr}(K) = 1, \tag{1}$$

where log is the matrix logarithm operation and $K \succ 0$ denotes that K is positive definite. The quantity $E(K)$ is called von Neumann entropy. In von Neumann entropy, the notion of a smooth distribution is extended to the smoothness of eigenvalues, because the von Neumann entropy can be written as the Shannon entropy of eigenvalues (Nielsen and Chuang, 2000). As seen in kernel Principal Components Analysis (PCA), a few high eigenvalues means that the samples are concentrated in a linear subspace of low dimensionality (Schölkopf and Smola, 2002). Roughly speaking, maximizing entropy amounts to placing the samples in the feature space as evenly as possible.

2.3 Maximizing entropy

We propose to learn a kernel matrix by maximizing the von Neumann entropy,

$$\begin{aligned} \min_K \quad & \mathrm{tr}(K \log K), \\ & \mathrm{tr}(K) = 1, \quad \mathrm{tr}(KU_j) \le 0, \ \ j = 1, \ldots, m. \end{aligned} \tag{2}$$

Note that we dropped the constraint $K \succ 0$, because the optimal solution of (2) is always positive definite, as shown below. If the optimization problem is feasible (i.e. if there is a matrix satisfying all the constraints, then it can be converted into the dual unconstrained problem by means of Lagrange multipliers. Since $\mathrm{tr}(K \log K)$ is a strictly convex function (Nielsen and Chuang, 2000), the optimal solution of the dual problem corresponds to that of the original problem as follows.

THEOREM 1. *The dual problem of (2) is described as*

$$\min_{\boldsymbol{\alpha} \ge 0} \ \log \mathrm{tr}\left[\exp\left(-\sum_j \alpha_j U_j \right) \right]. \tag{3}$$

The optimal dual parameter $\boldsymbol{\alpha}^*$ corresponds to the optimal primal parameter K^* as

$$K^* = \frac{1}{Z(\boldsymbol{\alpha}^*)} \exp\left(-\sum_j \alpha_j^* U_j \right), \tag{4}$$

where $Z(\boldsymbol{\alpha}^*) = \mathrm{tr}\left[\exp\left(-\sum_j \alpha_j^* U_j\right)\right]$.

The proof is given in Appendix A. The solution K^* is always positive definite because the matrix exponential maps any symmetric matrix to a positive definite matrix. The dual optimization problem (3) is strictly convex, so the global optimal solution can be obtained by standard convex optimization methods without local minima problems. In the following experiments, we use a simple gradient descent algorithm (details in Appendix C).

2.4 Relaxing constraints

When the problem (2) is infeasible, one can relax the problem as follows:

$$\begin{aligned} \min_{K,\xi} \quad & \mathrm{tr}(K \log K) + \lambda \textstyle\sum_{j=1}^m \xi_j, \\ & \mathrm{tr}(K) = 1, \quad \mathrm{tr}(KU_j) \le \xi_j, \\ & \xi_j \ge 0, \quad j = 1, \ldots, m. \end{aligned} \tag{5}$$

Here, λ controls the trade-off between entropy and the violation of constraints. The resulting dual problem is almost the same as the original dual problem, but with a different constraint $0 \le \alpha_j \le \lambda$. The derivation is described in Appendix B.

3 LOCALLY CONSTRAINED DIFFUSION KERNEL

Let A be the $n \times n$ adjacency matrix of a graph. Also, let D be the $n \times n$ diagonal matrix such that D_{ii} is the vertex degree of i-th node, i.e. the number of edges involving the i-th node. The so-called graph Laplacian matrix is defined as $L = D - A$. The diffusion kernel (Kondor and Lafferty, 2002) is then defined as

$$K = \exp(-\beta L),$$

where the diffusion parameter $\beta > 0$ determines the degree of diffusion. The diffusion kernel can be interpreted in terms of lazy random walking for sufficiently small β (Kondor and Lafferty, 2002). At each step of random walking, the next node is randomly chosen from the neighboring nodes according to the transition probabilities. The ramdom walk is 'lazy' here, because one can stay at the same node (i.e. self-loop). If we fix the transition probability to every neighboring node as a constant β, then the self-loop probability of node i is $1 - d_i\beta$, where d_i is the degree of node i. The kernel value K_{ij} is then equivalent to the probability that a random walk starting from i will be at j after infinite time steps. Figure 2 shows the actual values of diffusion kernels with different settings of β. When β is sufficiently large, the kernel values among distant nodes capture the long-range relationships between proteins or genes.

Let us consider the normalized version of diffusion kernel,

$$K = \frac{1}{Z(\beta)} \exp(-\beta L), \tag{6}$$

where $Z(\beta) = \mathrm{tr}[\exp(-\beta L)]$. Note that the normalization (6) does not affect the results of kernel-based learning algorithms (Schölkopf and Smola, 2002). This kernel matrix can be derived as the optimal solution for the following

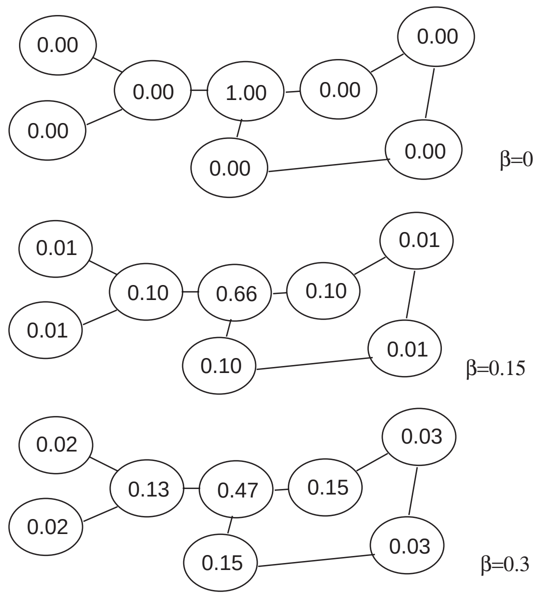

Fig. 2. Actual values of the diffusion kernel for different settings of the diffusion parameter β. Each value on a node shows the kernel value between the node and the 'central node' (i.e. the node with 1.00 when $\beta = 0$). As the diffuson parameter increases, the kernel values diffuse more completely through the graph.

maximum entropy problem:

$$\min_K \ \operatorname{tr}(K \log K), \quad \operatorname{tr}(K) = 1, \ \operatorname{tr}(KL) \le c,$$

where c is a positive constant, which is an implicit function of β. Thus, the diffusion kernel is the maximum entropy solution subjected to single global constraint $\operatorname{tr}(KL) \le c$. Let $\{s_j, t_j\}_{j=1}^m$ denote the node pairs connected by m edges. The quantity $\operatorname{tr}(KL)$ equals the sum of Euclidean distances between connected samples:

$$\operatorname{tr}(KL) = \sum_{j=1}^m \|\boldsymbol{x}_{s_j} - \boldsymbol{x}_{t_j}\|^2. \tag{7}$$

Because the global constraint constrains the sum of distances only, each distance is allowed to be drastically different. In fact, in our experiments, we observed high variances in the distances produced by this kernel (Section 4).

In order to impose a more uniform network structure, let us consider the following local constraints:

$$\|\boldsymbol{x}_{s_j} - \boldsymbol{x}_{t_j}\|^2 \le \gamma, \quad j = 1, \dots, m.$$

The maximum entropy problem (without relaxation) is described as

$$\begin{aligned} \min_K \ & \operatorname{tr}(K \log K), \quad \operatorname{tr}(K) = 1, \\ & \operatorname{tr}(KV_j) \le \gamma, \ \ j = 1, \dots, m, \end{aligned} \tag{8}$$

where

$$[V_j]_{st} = \begin{cases} 1 & (s = s_j, t = s_j) \text{ or } (s = t_j, t = t_j) \\ -1 & (s = s_j, t = t_j) \text{ or } (s = t_j, t = s_j) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Note that $L = \sum_j V_j$. When written in the standard form (2), $U_j = V_j - \gamma I$. We call the resulting kernel a 'locally constrained diffusion kernel'. As will be shown in the next section, this method does not produce extremely large distances, and yields good results in function prediction.

4 EXPERIMENTS

We computed kernels from two different types of yeast biological networks. The first network was derived by Vert and Kanehisa (2003) from the LIGAND database of chemical reactions in biological pathways (www.genome.ad.jp/ligand). In this graph, two proteins are linked if they catalyze two successive reactions, in which the primary product of the first reaction is the primary substrate of the second reaction. Thus, a path in this graph represents a possible series of reactions catalyzed by proteins along the path. The resulting metabolic network contains 755 proteins and 7860 edges.

The second network was created by von Mering *et al.* (2002) from protein–protein interactions identified via six different methods: high-throughput yeast two-hybrid, correlated mRNA expression, genetic interaction (synthetic lethality), tandem affinity purification, high-throughput mass-spectrometric protein complex identification and computational methods. All interactions were classified into one of three confidence categories, high-, medium- and low-confidence, based on the number of different methods that identify an interaction as well as the number of times the interaction is observed. In these experiments, we used a medium confidence network containing 2617 proteins and 11 855 edges.

The reduced variance in the values produced by the locally constrained diffusion kernel can be seen in Figure 3. This figure displays, using heat maps, distance matrices derived from the metabolic network using locally and globally constrained diffusion kernels. In the globally constrained case, some distances are so large that the others are almost invisible, whereas the distances produced by the locally constrained kernel are more uniform. A direct comparison of the local and global distributions produced by both networks is given in Table 1. In both cases, the global kernel produces some extremely large distances, whereas the local kernel distances concentrate around γ.

In addition to comparing global and local kernels directly, we tested the kernels' utility in the context of an SVM classification task. We used the functional categories of the MIPS Comprehensive Yeast Genome Database (CYGD; mips.gsf.de/genre/proj/yeast) as a gold standard. These categories are not mutually exclusive and hence are appropriately

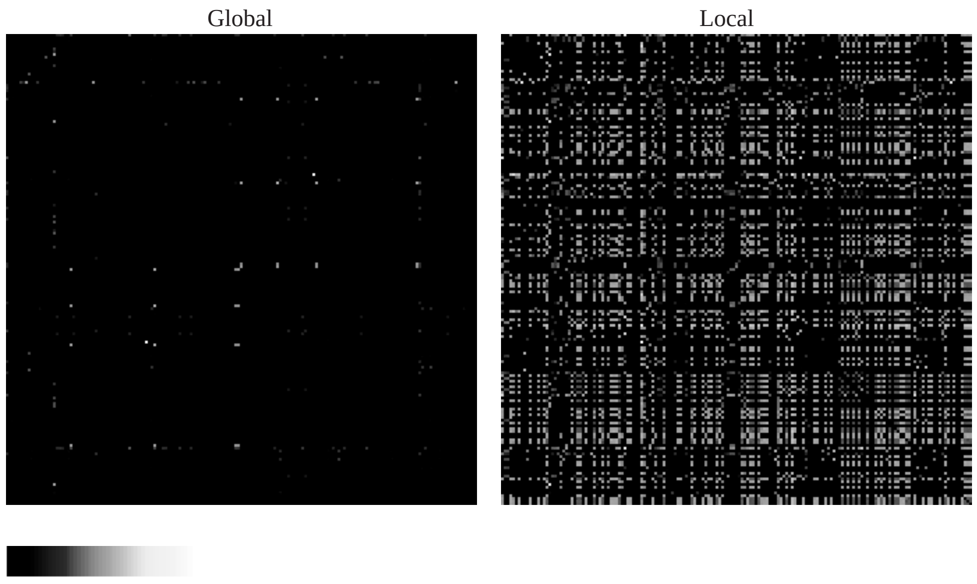

Fig. 3. Distance matrices from globally and locally constrained diffusion kernels. The figure shows heat map representations of two distance matrices, produced using the metabolic network with $\beta = 2$. Colors are mapped using a black-body heat map (lower left). For visualization purposes, the matrix includes only nodes with degree more than 30. These images were produced using matrix2png (Pavlidis and Noble, 2003) (see online figure for colours).

Table 1. Distribution of distances in globally and locally constrained diffusion kernels

Network	Constraint	$\leq\gamma$	2γ	10γ	100γ	1000γ	$>1000\gamma$
Metabolic	Global	7249	231	229	130	21	0
Metabolic	Local	3706	4139	15	0	0	0
Interaction	Global	9967	448	979	461	0	0
Interaction	Local	4927	6913	15	0	0	0

The digits show the number of distances that fall between the value of previous column and the current one. For these kernels, $\beta = 2$, and the constant γ is determined using (10).

treated as independent binary classification tasks. We selected all functional categories containing at least 30 positive examples (see online supplement for the complete list), resulting in 36 categories for the metabolic network and 76 categories for the protein–protein interaction network. In computing diffusion kernels, the parameter β was varied from 0.2 to 6. For each value of β, γ was set such that the overall strength of constraints was equal:

$$\gamma = \mathrm{tr}[\exp(-\beta L)L]/m. \quad (10)$$

The parameter λ was set to 100 throughout the experiments. For each category and each kernel, an SVM was trained using half of the data, and its performance was measured on the remaining half. We used the SVM implementation in the SPIDER software package (www.kyb.tuebingen.mpg.de/bs/people/spider). Here, we report results when the regularization parameter $C = 1000$; see the online supplement for similar results from other settings of C. The quality of the predictions produced by the SVM was measured using the receiver operating characteristic (ROC) score, which is the area under a curve that plots the true positive rate as a function of false positive rate, for varying classification thresholds. A perfect classifier ranks all the positives above all the negatives and receives an ROC score of 1; a random classifier receives a score of ∼0.5.

Figure 4 compares the classification performance of SVMs trained using locally versus globally constrained maximum entropy kernels. The figure shows that, for both types of network, the SVM performance is relatively robust across a wide range of values of the diffusion parameter β. Furthermore, in every case, the locally constrained kernel outperforms the globally constrained kernel. Figure 5 displays the same results without averaging, selecting for each class and each kernel the value of β that yields the best ROC score. The complete collection of ROC scores with respect to each CYGD classification is available online.

5 DISCUSSION

Much recent research has focused on the problem of inferring various types of biological networks from genome-wide data sets (Segal *et al.*, 2003; Pe'er *et al.*, 2001; Toh and Horimoto, 2002; Pilpel *et al.*, 2001; Yeung *et al.*, 2002). Here, we focus on the downstream problem: given an (observed or inferred) biological network, how do we use that network to draw inferences about the genes or proteins in the network? Such inferences might be drawn from any type of

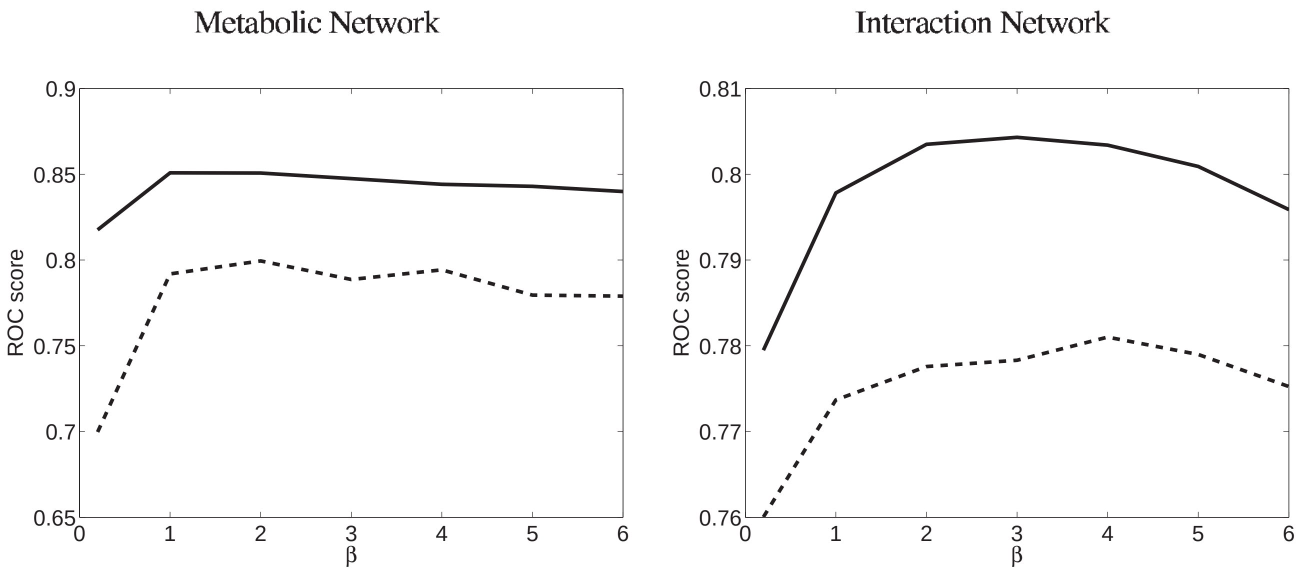

Fig. 4. Mean ROC score as a function of the diffusion parameter β. The plots show the mean ROC scores computed across the set of CYGD categories, using (**A**) the metabolic network and (**B**) the protein–protein interaction network. The solid and broken lines correspond to locally and globally constrained diffusion kernels, respectively.

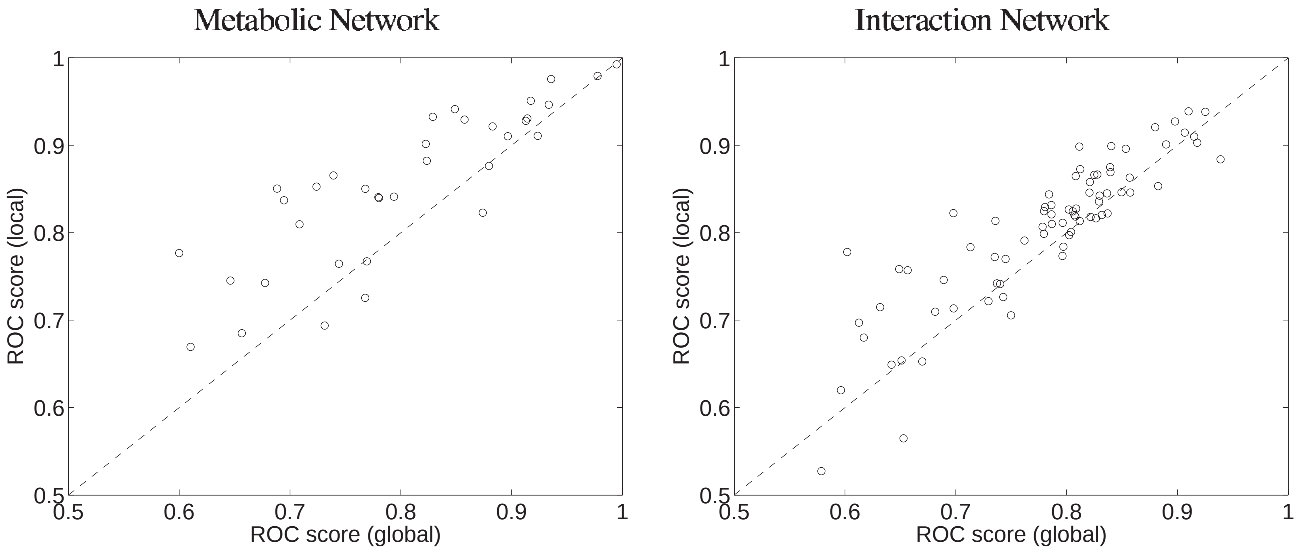

Fig. 5. Category-by-category comparison of SVM ROC scores from locally and globally constrained kernels. Each plot compares the performance of the locally constrained (y-axis) and globally constrained (x-axis) maximum entropy kernels. Each point in the plot corresponds to a single CYGD functional category. For both methods, the value of the diffusion parameter β is selected that yields the highest ROC score.

biological network and might concern molecular function, cellular component or biological process. Thus, the diffusion kernel, including the locally constrained version described here, has broad applicability.

Furthermore, in contrast to methods that directly derive label predictions from the network (Vazquez *et al*., 2003; Deng *et al*., 2003), the kernel approach detaches the distance design from the statistical inference. The derived distance may then be combined with other data and used for multiple purposes, including visualization, clustering, classification and regression.

ACKNOWLEDGEMENTS

K.T. would like to thank M. Warmuth, G. Rätsch, B. Schölkopf, M. Belkin and D. Zhou for fruitful discussions. W.S.N. is a Sloan Foundation Research Fellow and is supported by NSF awards ISI-0093302 and DBI-0078523.

REFERENCES

Deng,M., Chen,T. and Sun,F. (2003) An integrated probabilistic model for functional prediction of proteins. In Miller,W., Vingron,M., Istrail,S., Pevzner,P. and Waterman,M. (eds), *Proceedings of the Seventh Annual International Conference on Computational Biology (RECOMB)*. ACM Press, New York, pp. 95–103.

Goto,S., Okuno,Y., Hattori,M., Nishioka,T. and Kanehisa,M. (2002) LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res.*, **30**, 402–404.

Hishigaki,H., Nakai,K., Ono,T., Tanigaki,A. and Takagi,T. (2001) Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, **18**, 523–531.

Ihmels,J., Friedlander,G., Bergmann,S., Sarig,O., Ziv,Y. and Barkai,N. (2002) Revealing modular organization in the yeast transcriptional network. *Nat. Genet.*, **31**, 370–377.

Kondor,I. and Lafferty,J. (2002) Diffusion kernels on graphs and other discrete structures. In Sammut,C. and Hoffmann,A.G. (eds), *Machine Learning. Proceedings of the Nineteenth International Conference (ICML 2002)*, Morgan Kaufmann, San Francisco, pp. 315–322.

Lanckriet,G.R.G., Deng,M., Cristianini,N., Jordan,M.I. and Noble,W.S. (2004) Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the Pacific Symposium on Biocomputing (PSB)*. pp. 300–311.

Lee,T.I., Rinaldi,N.J., Robert,F., Odom,D.T., Bar-Joseph,Z., Gerber,G.K., Hannett,N.M., Harbison,C.R., Thompson,C.M., Simon,I. *et al*. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, **298**, 799–804.

Nielsen,M.A. and Chuang,I.L. (200) *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge.

Noble,W.S. (2004) Support vector machine applications in computational biology. In Schölkopf,B., Tsuda,K. and Vert,J.P. (eds), *Kernel Methods in Computational Biology*. MIT Press, Cambridge, MA (in press).

Pavlidis,P. and Noble,W.S. (2003) Matrix2png: a utility for visualizing matrix data. *Bioinformatics*, **19**, 295–296.

Pavlidis,P., Weston,J., Cai,J. and Grundy,W.N. (2001) Gene functional classification from heterogeneous data. In *Proceedings of the Fifth Annual International Conference on Computational Biology (RECOMB)*, ACM Press, pp. 242–248.

Pe'er,D., Regev,A., Elidan,G. and Friedman,N. (2001) Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17**, S215–S224.

Pilpel,Y., Sudarsanam,P., and Church,G.M. (2001) Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat. Genet.*, **2**, 153–159.

Schölkopf,B. and Smola,A.J. (2002) *Learning with Kernels*. MIT Press, Cambridge, MA.

Schwikowski,B., Uetz,P. and Fields,S. (2000) A network of protein–protein interactions in yeast. *Nat. Biotechnol.*, **18**, 1257–1261.

Segal,E., Shapira,M., Regev,A., Pe'er,D., Botstein,D., Koller,D. and Friedman,N. (2003) Module networks: identifying regulatory modules and their condition specific regulators from gene expression data. *Nat. Biotechnol.*, **34**, 166–176.

Toh,H. and Horimoto,K. Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, **18**, 287–297.

Tsuda,K., Akaho,S. and Asai,K. (2003) The em algorithm for kernel matrix completion with auxiliary data. *J. Mach. Learning Res.*, **4**, 67–81.

Uetz,P., Giot,L., Cagney,G., Mansfield,T.A., Judson,R.S., Knight,J.R., Lockshon,D., Narayan,V., Srinivasan,M., Pochart,P. *et al*. (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature*, **403**, 623–627.

Vazquez,A., Flammini,A., Maritan,A. and Vespignani,A. Global protein function prediction from protein–protein interaction networks. *Nat. Biotechnol.*, **21**, 697–700.

Vert,J.P. and Kanehisa,M. (2003) Graph-driven features extraction from microarray data using diffusion kernels and kernel CCA. In Becker,S., Thrun,S. and Obermayer,K. (eds), *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, pp. 1425–1432.

von Mering,C., Krause,R., Snel,B., Cornell,M., Olivier,S.G., Fields,S. and Bork,P. (2002) Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, **417**, 399–403.

Wu,N. (1997) *Maximum Entropy Method*. Springer-Verlag, New York.

Yeung,M.K., Tegner,J. and Collins,J.J. (2002) Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl Acad. Sci., USA*, **99**, 6163–6168.

Yona,G., Linial,N. and Linial,M. (1999) Protomap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Prot. Struct. Funct. Genet.*, **37**, 360–678.

APPENDIX A: PROOF OF THEOREM 1

The Lagrangian of (2) is described as

$$L = \mathrm{tr}(K \log K) + \sum_j \alpha_j \mathrm{tr}(KU_j) + \beta(\mathrm{tr}K - 1), \qquad \text{(A1)}$$

where $\alpha_j \in \Re, \beta \in \Re$ and $\alpha_j \geq 0$. Setting the derivative with respect to K to the zero matrix, we have

$$I + \log K + \sum_j \alpha_j U_j + \beta I = 0.$$

This equation is solved as follows:

$$K = \exp\left[-\sum_j \alpha_j U_j - (1+\beta)I\right]. \qquad \text{(A2)}$$

$$= \exp[-(1+\beta)] \exp\left(-\sum_j \alpha_j U_j\right). \qquad \text{(A3)}$$

Substituting it into the Lagrangian, we have

$$L = -\exp[-(1+\beta)]\mathrm{tr}\left[-\exp\left(\sum_j \alpha_j U_j\right)\right] - \beta. \qquad \text{(A4)}$$

The optimal β that maximizes (A4) is obtained as

$$\beta = \log \mathrm{tr}\left[\exp\left(-\sum_j \alpha_j U_j\right)\right] - 1. \qquad \text{(A5)}$$

Substituting (A5) into (A3), K is represented by α_j only

$$K = \frac{1}{Z(\boldsymbol{\alpha})} \exp\left(-\sum_j \alpha_j U_j\right),$$

where $Z(\boldsymbol{\alpha}) = \mathrm{tr}\left[\exp\left(-\sum_j \alpha_j U_j\right)\right]$. Substituting (A5) into (A4), we finally have the dual problem

$$\max_{\boldsymbol{\alpha}\geq 0} -\log\, \mathrm{tr}\left[\exp\left(-\sum_j \alpha_j U_j\right)\right]. \qquad \text{(A6)}$$

APPENDIX B: DUAL PROBLEM FOR SOFT MARGIN

The Lagrangian of (5) is written as

$$\begin{aligned} L \;=\; & \mathrm{tr}(K \log K) + \lambda \sum_j \xi_j + \sum_j \alpha_j [\mathrm{tr}(KU_j) - \xi_j] \\ & - \sum_j \delta_j \xi_j + \beta[\mathrm{tr}(K) - 1], \end{aligned}$$

where the Lagrange multipliers $\alpha_j \geq 0, \delta_j \geq 0$. Setting the derivative with respect to ξ_j to zero, we have

$$\lambda - \alpha_i - \delta_i = 0,$$

which is rewritten as $\delta_i = \lambda - \alpha_i$. Since $\delta_i \geq 0$, the inequality $\alpha_i \leq \lambda$ is derived. Substituting it into the Lagrangian, we have

$$L = \mathrm{tr}(K \log K) + \sum_j \alpha_j \mathrm{tr}(KU_j) + \beta[\mathrm{tr}(K) - 1],$$

which is exactly the same as the original Lagrangian (A1). Therefore, following the same steps in Appendix A, we arrive at the same dual problem with new constraints $\alpha_i \leq \lambda$.

APPENDIX C: GRADIENT DESCENT OPTIMIZATION

We use the gradient descent method for minimizing the dual function

$$f(\boldsymbol{\alpha}) = \log\, \mathrm{tr}\left[\exp\left(-\sum_j \alpha_j U_j\right)\right].$$

The current parameter $\boldsymbol{\alpha}^{(t)}$ is updated to $\boldsymbol{\alpha}^{(t+1)}$ as follows:

$$\boldsymbol{\alpha}^{(t+1)} = \boldsymbol{\alpha}^{(t)} - \eta_t \frac{\nabla f(\boldsymbol{\alpha}^{(t)})}{\|\nabla f(\boldsymbol{\alpha}^{(t)})\|},$$

where η_t determines the learning rate, and $\nabla f(\boldsymbol{\alpha}_t)$ is the gradient vector. The k-th element of $\nabla f(\boldsymbol{\alpha}_t)$ is written as

$$\frac{\partial}{\partial \alpha_k} f(\boldsymbol{\alpha}) = \frac{\mathrm{tr}\left[-U_k \exp\left(-\sum_j \alpha_j U_j\right)\right]}{\mathrm{tr}\left[\exp\left(-\sum_j \alpha_j U_j\right)\right]}.$$

When the updated parameter falls out of the interval $0 \leq \alpha_j \leq \lambda$, it is pulled back to 0 or λ. In the experiments, we started from the initial rate $\eta_1 = 100$. When the rate is too large and $f(\boldsymbol{\alpha})$ is increased by the update, η is reduced to its half. We performed a maximum of 30 iterations in order to minimize the function.