The Third International Symposium on Optimization and Systems Biology (OSB'09)
Zhangjiajie, China, September 20–22, 2009
Copyright © 2009 ORSC & APORC, pp. 25–32

A Study of Network-based Kernel Methods on Protein-Protein Interaction for Protein Functions Prediction

Wai-Ki Ching[1,*] Limin Li[1,†] Yat-Ming Chan[1,‡]
Hiroshi Mamitsuka[2,§]

[1] Advanced Modeling and Applied Computing Laboratory, Department of Mathematics,
The University of Hong Kong, Hong Kong
[2] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji,
Kyoto 611-0011, Japan

Abstract Predicting protein functions is an important issue in the post-genomic era. In this paper, we studied several network-based kernels including Local Linear Embedding (LLE) kernel method, Diffusion kernel and Laplacian Kernel to uncover the relationship between proteins functions and Protein-Protein Interactions (PPI). We first construct kernels based on PPI networks, we then apply Support Vector Machine (SVM) techniques to classify proteins into different functional groups. 5-fold cross validation is then applied to the selected 359 GO terms to compare the performance of different kernels and guilt-by-association methods including neighbor counting methods and Chi-square methods. Finally we made predictions of functions of some unknown genes and verified the preciseness of our prediction in part by the information of other data source.

Keywords Protein Function Prediction; Kernel Method; Local Linear Embedding (LLE) Kernel; Laplacian Kernel; Diffusion Kernel; Support Vector Machine

1 Introduction

Assigning biological functions to an uncharacterized protein is an immediate challenge in the post-genomic era. To our best knowledge, even for the most well-studied organisms such as yeast, there are still about one-fourth of the proteins remain uncharacterized. Recently different data sources and different methods have been used to predict protein functions including those based on Protein-Protein Interaction (PPI), structure, sequence relationship, gene expression data, see for instance [9, 13, 14, 19, 20]. The classical methods for learning the protein functions are based on sequence similarity tools such as FASTA and BLAST. In such methods, the query protein sequence is used as an input to find a significantly similar sequence whose function has been characterized.

[*] Email: wching@hkusua.hku.hk
[†] Corresponding Author. Email: liminli@hkusua.hku.hk
[‡] Email: ymchan@maths.hku.hk
[§] Email: mami@kuicr.kyoto-u.ac.jp

High-throughout experimental techniques have generated a large amount of data which are useful for inferring the functional roles of proteins. Gene expression data is one of these useful data sources, and several function prediction methods have been proposed [1, 4]. However, discrepancies of prediction may arise due to the corruptions of gene expression data. Occasionally, the microarrays contain bad probes or are even damaged, and some locations in the gene expression matrix are corrupted. Protein-Protein Interaction (PPI) plays a key role in many cellular processes. The distortion of protein interfaces may lead to the development of many diseases. The global picture of protein interactions in the cell provides a new way to understand the mechanisms of protein recognition at the molecular level. This newly available large-scale PPI data gives an opportunity to study protein functions in the context of a network. The PPI data can be represented as a network, with nodes representing proteins and edges representing the interactions between the nodes. Many methods have been proposed to elucidate protein functions using PPI data. One of the simplest methods is the guilty-by-association methods, i.e., the neighbor-counting method [16]. The method predicts for a given protein up to three functions that are most common among its neighbors. The Chi-square method [7], it computes the Chi-square scores of function assignment and assign the functions with several largest scores to a given protein. Vazquez et al. [17], Karaoz [8] and Nabieva [15] applied graph algorithms such as cut-based approach and flow-based approach for functional analysis. In contrast to the local neighbor-counting methods, these methods take into account the full topology of the network. Deng et al. proposed Markov Random Field (MRF) method [3] to predict yeast protein functions based on a PPI network. They assign functions to unknown proteins with a probability representing the confidence of the prediction. From the experimental results, MRF method shows 52% precision and recall and is much better than those simple guilty-by-association methods. Lanckriet et al. [11] considered a Support Vector Machine (SVM) approach for predicting protein functions using a diffusion kernel on a protein interaction network. The diffusion kernel provides means to incorporate all neighbors of proteins in the network. Lee et al. [12] developed a novel Kernel Logistic Regression (KLR) method based on diffusion kernel for protein interaction networks and showed that the prediction accuracy is comparable to the protein function classifier based on the SVM, using a diffusion kernel.

The remainder of this paper is structured as follows. Section 2 gives an introduction to the kernel methods. In Section 3, numerical experiments are given to demonstrate the effectiveness of our proposed method. Finally concluding remarks are given in Section 4 to address further research issues.

2 The Kernel Methods

In this section, we first give a brief description of kernel methods and three network-based kernels: diffusion kernel, Laplacian kernel, and Local Linear Embedding (LLE) kernel. After the kernel is generated, SVM method is then applied to each GO term to classify whether a new gene is in the GO term or not.

Kernel methods [10, 12] attempt to express the correlations or similarities between pairs of points in the data space Ω in terms of a kernel function $K : \Omega \times \Omega \mapsto R$, and thereby implicitly construct a mapping $\phi : \Omega \mapsto H_K$ to a Hilbert space (feature space) H_K, in which the kernel can be represented as an inner product: $K(x,y) = (\phi(x), \phi(y))$. Besides expressing the known structure of the data space, the function or the kernel K must satisfy two mathematical requirements: (i) it must be symmetric, i.e., $K(x,y) = K(y,x)$ and (ii) it should be positive semi-definite. In fact, effectiveness of a kernel-based method lies on the fact that it can implicitly map a data point to a higher dimensional feature space which can better captures the inherent structure of the data. The kernel K of a graph G with N nodes is an $N \times N$ real symmetric matrix such that and its element K_{ij} represents the similarity between Node i and Node j. We will make use of the graph-like structure of a PPI network to construct the global similarity for any pair of proteins in the network, and perform SVM classification based on the kernel.

To facilitate our discussion, we introduce the following notations. Let G be a PPI network of N proteins. Then one can represent the network G by its adjacent matrix $W = (w_{ij}) \in \mathbb{R}^{N\times N}$ where $w_{ij} = 1$ means there is an edge between Node i and Node j in the network, otherwise there is no edge between them. We define $D = (d_{ij})$, where

$$d_{ii} = \Sigma_j w_{ij} \quad \text{and} \quad d_{ij} = 0 \text{ if } i \neq j.$$

The graph Laplacian is defined as $L = D - W$. We consider the feature for each protein determined by its neighborhood relationship with all the other proteins, then the trivial linear kernel can be defined as $K_{linear} = W^T W$.

Diffusion Kernel: Kondor and Lafferty [10] proposed a general method for establishing similarities among the nodes of a graph based on a random walk on the graph. This method efficiently accounts for all possible paths connecting two nodes, and for the lengths of those paths. Nodes that are connected by shorter paths or by many paths are considered to be more similar to each other. Let the eigenvalue decomposition of L be

$$L = U \cdot \text{diag}(\lambda_1, \cdots, \lambda_N) \cdot U^{-1}, \tag{1}$$

then the kernel generated is defined as

$$K = U \cdot \text{diag}(e^{-\frac{\sigma^2}{2}\lambda_1}, \cdots, e^{-\frac{\sigma^2}{2}\lambda_N}) \cdot U^{-1} = e^{-\frac{\sigma^2}{2}L}. \tag{2}$$

The diffusion constant σ controls the rate of diffusion through the network. By varying the parameter σ, one can get different kernels. The diffusion kernel has been applied by Lanckriet et al. [11] in protein-protein network to predict protein functions.

Laplacian Kernel: This kernel [6] is a kind of network-based kernel and is generated by the adjacent matrix W. The Laplacian kernel is defined as

$$K = L^{\dagger} = (D - W)^{\dagger} \tag{3}$$

where $L^{\dagger}$ is the pseudo-inverse of the matrix L.

Local Linear Embedding Kernel: The LLE is an unsupervised learning algorithm that computes low-dimensional, neighborhood-preserving embeddings for high-dimensional inputs [18]. The input of LLE is N high-dimensional data points(m dimension), and output is the corresponding N low-dimensional data points (d-dimension). The three main steps in LLE are the followings:

(i) Identify the neighbors of each data point $x_i \in R^m$. Denote N_i is the index set for the k-neighbors of x_i;

(ii) Compute the weights that best linearly reconstruct x_i from its neighbors. This can be done by solving the minimization problem:

$$\min_{A=(a_{ij})\in R^{N\times N}} \left\{ \sum_{i=1}^{N} \left| x_i - \sum_{j\in N_i}^{k} \alpha_{ij} x_j \right|^2 \right\}. \tag{4}$$

(iii) Find the low-dimensional embedding vectors by solving

$$\min_{Y=[y_1,\cdots,y_N]\in R^{d\times N}} \left\{ \sum_{i=1}^{N} \left| y_i - \sum_{j\in N_i}^{k} \alpha_{ij} y_j \right|^2 \right\}. \tag{5}$$

with the constraints $\frac{1}{N}YY^T = I$ and $Y\mathbf{e} = 0$, where $\mathbf{e}$ is the column vector with all ones. It has been shown that this problem can be solved by the eigenvalue problem of the matrix $M = (I-A)^T(I-A)$, where A is the weight matrix obtained in Step (ii). The optimal d-dimensional embedding Y can be obtained by the $(N-1-d)$th to $(N-1)$th eigenvectors of M when its eigenvalues are in decreasing order.

In the LLE method, we first constructs for each data point a local geometric structure that is invariant to translations and orthogonal transformations in its neighborhood. We then project the data points into a low-dimensional space that best preserves those local geometries. In the case of a PPI network, we assume that each protein can be represented as a m-dimensional vector and all the points lie on a d-dimensional manifold with noise, where m and d are both unknown. For each point, all its neighbors in the PPI network will then be used to construct the local geometry based on the hypothesis that the weights for its different neighbors are same in its neighborhood, thus we can put the weight matrix A in Step (ii) to be the normalized adjacent matrix $A = D^{-1}W$. After Step (iii) of LLE, the intuitive way to do the classification is to perform SVM on some kernel defined by the LLE output Y to classify proteins into different functional group.

Since the low dimension d is difficult to determine, we use the following alternative way to perform the SVM classification. Let λ_{max} be the largest eigenvalue of M, then the LLE kernel is defined as

$$K_{LLE} = \lambda_{max} I - M. \tag{6}$$

Here I is the identity matrix. It is easy to prove that the leading eigenvector of K_{LLE} is $\mathbf{e}$, and the second eigenvectors up to the $(d+1)$th eigenvaector provide the d-dimensional LLE embedding Y. Let $K_{LLE} = U\Lambda U^T$ where $U = [\mathbf{u}_1,\cdots,\mathbf{u}_N]$ and $\Lambda = \text{diag}(\lambda_1,\cdots,\lambda_N)$ with $\lambda_1 \geq \cdots \geq \lambda_N$ then $Y = [\mathbf{u}_2,\cdots,\mathbf{u}_N]^T$. Here we used this LLE kernel to perform SVM and to classify the proteins into different functions. In fact, there is a close relationship between this kernel and a Y-based kernel. We define a low dimensional kernel matrix based on low dimension embedding Y as $K_{Low} = Y^T \Lambda_d Y \in R^{N\times N}$ where $\Lambda_d = \text{diag}(\lambda_2,\cdots,\lambda_{d+1})$. It is easy to prove that

$$K_{LLE} - K_{Low} = \lambda_{max}\mathbf{e}\mathbf{e}^T + \sum_{i=d+2}^{N} \lambda_i \mathbf{u}_i \mathbf{u}_i^T. \tag{7}$$

This means when d is large enough, there's only a difference of a constant matrix with all same elements between LLE kernel K_{LLE} and Y-based kernel K_{Low}.

3 Experimental Results

3.1 Data Source

We use GO data at $http://www.geneontology.org/ontology/gene\_ontology.obo$ in our numerical experiment. The gene association data is taken from SGD in Feb 2008. The PPI data is downloaded from MIPS database, which contains a manually curated yeast protein interaction dataset [5] collected by curators from the literature.

3.2 The Gene Ontology

The Gene Ontology (GO) is a framework consisting of controlled vocabularies describing three aspects of gene product functions: (i) molecular function, (ii) biological process and (iii) cellular component. Each aspect of the functions is called an ontology. Each ontology is a Directed Acyclic Graph (DAG) where the GO terms are represented as nodes in the graph and are arranged hierarchically from a general one to a specific one. Here functional annotation of protein is defined by GO biological process. The hierarchical structure of GO indicates that if a gene is assigned to one term, then the gene will be assigned to all ancestors of this term indirectly. In the following discussion, we assume that the genes associated to a node include all the indirect genes associated to this node. It should be noted that a gene can be in more than one GO class.For each GO term T, all proteins that are annotated with T are labeled as positive, while all proteins that are not annotated with T are labeled as negative. Generally speaking, for each GO term, the number of negative proteins far exceeds the number of positive proteins. In this case, to test and compare the efficiency of different method, we randomly select a subset of negative proteins so that the number of positives and negatives are equal. Thus for each GO term, after labeling the training set, one can use SVM technique to generate a SVM classifier, which will be used to classify the unknown proteins into positive or negative classes.

3.3 The Prediction Performance

We first extracted a subnetwork of the whole PPI network to make sure every protein in the subnetwork has been annotated by GO. The number of the nodes in the subnetwork used in the cross validation is 3187. We generated different kinds of graph kernels for these 3187 proteins. We note that we did not use all the GO Terms to check the classification performance because for most of GO Terms, there are too few positive genes (less than 30) and for some GO Terms, there are too many positive genes (more than 1000). We removed all these GO terms, and 359 GO terms are left. We then evaluated the classification performance by 5-fold cross validation using kernel methods with different kernels including linear kernel, LLE kernel, diffusion kernel and Laplacian kernel and guilt-by-association methods including neighbor counting method and Chi-square method. For the diffusion kernel, we chose the diffusion constant σ to be $0.5, 1, 2$ and 3. For each GO class, a classifier can be constructed by training the proteins in training data set. Then this classifier will be used to classify the proteins in the test data set into either positive or negative group. For each method, we calculate 359 AUCs for all the 359 GO Terms and an AUC for the multiple classification [2]. The ROC curves for these methods are shown

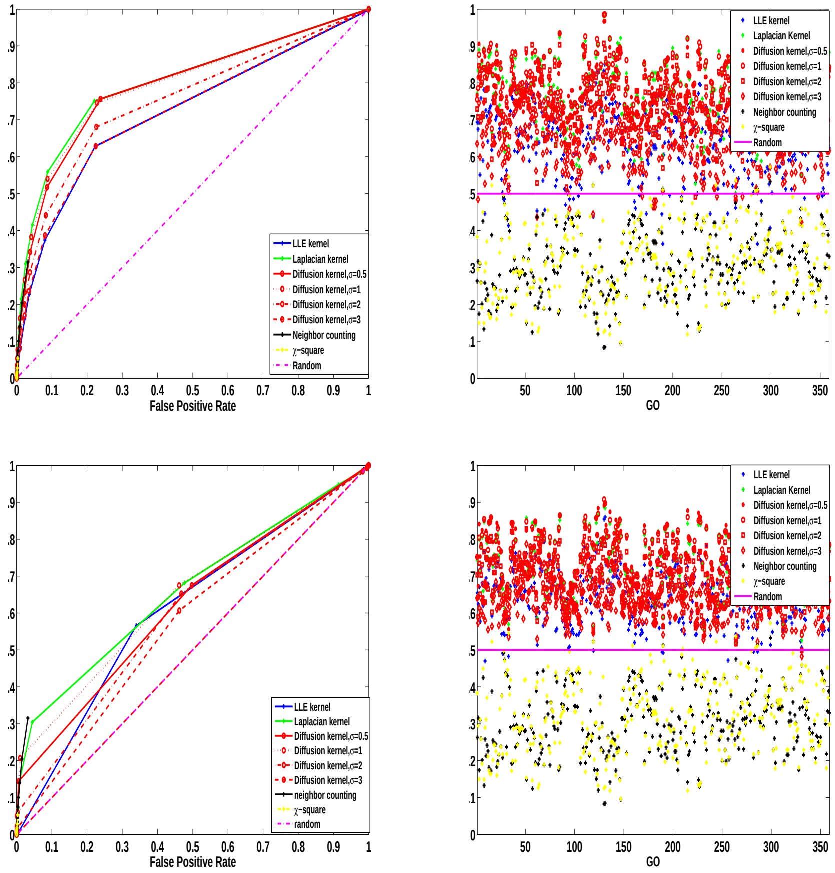

Figure 1: Prediction results: Top (unbalanced methods); Bottom (balanced methods); Left: (ROC curves); Right: (AUCs for different GO Terms).

in the left of Figure 1. The top row of Figure 1 is the results of cross validation on all the proteins in PPI, and the bottom row of Figure 1 is the results for balanced protein sets. Left Column is ROC curves of different methods, and right column is the AUCs of 359 GO Terms. Table 1 gives the AUCs of different methods. From Figure 1 and Table 1, one can see that for any specific kernel, unbalanced method is generally better than balanced method. The results also show that Laplacian kernel and diffusion kernel with diffusion constant 1 are better than other kernels.

4 Concluding Remarks

In this paper, we proposed network-based kernel methods to predict protein functions. Five-fold cross validation is then applied to compare different kernels. The results indicate that unbalanced methods are better than balanced methods, and Laplacian and diffusion

AUC	Balanced	Unbalanced
LLE	0.6353	0.7705
Laplacian	**0.6770**	**0.8370**
Diffusion,0.5	0.6591	0.8268
Diffusion,1	**0.6778**	**0.8316**
Diffusion,2	0.6562	0.7987
Diffusion,3	0.6124	0.7600
Neighbor counting	0.2449	
χ-square	0.2578	

Table 1: Comparison of balanced and unbalanced methods

kernels performs best among all the kernels. In our future research, we will consider different integration of the different data sources such as sequence, structure, expression data, and PPI network with different kernel methods.

Acknowledges

The authors would like to thank the two anonymous referees for their helpful comments and suggestions. Research support in part by HKRGC Grant 7017/07P and HKU CRCG Grants and HKU strategic theme grant on computational sciences and HKU Hung Hing Ying Physical Science Research Grant.

References

[1] Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M. and Haussler, D. (2000) *Knowledge-based analysis of microarray gene expression data by using support vector machines.* Proc. Natl. Acad. Sci., 97:262-267.

[2] David, J. and Robert, J. (2001) *A simple generalisation of the area under the ROC curve for multiple class classification problems.* Machine Learning, 45:171-186.

[3] Deng, M., Tu Z., Sun, F. and Chen, T. (2003) *Mapping gene ontology to proteins based on protein-protein interaction data.* Bioinformatics, 20:895-902.

[4] Eisen, M., Spellman, P., Brown, P. and Bostein, D. (1998) *Cluster analysis and display of genome-wide expression patterns.* Proc. Natl. Acad. Sci., 95:14863-14868.

[5] Guldener, U., Munsterkotter, M., Oesterheld, M., Pagel, P. and Ruepp, A. (2006) *MPact: The MIPS protein interaction resource on yeast.* Nucleic Acids Res., 34:436-441.

[6] Ham, J., Lee, D., Mika, S. and Scholkopf, B. (2004) *A kernel view of the dimensionality reduction of manifolds.* Proceedings of the Twenty-First International Conference on Machine Learning (AAAI Press, Menlo Park, CA), 47-54.

[7] Hishigaki, H., Nakai, K., Ono, T., Tanigami, A. and Takagi, T. (2001) *Assessment of prediction accuracy of protein function from protein-protein interaction data.* Yeast, 18:523-531.

[8] Karaoz, U., Murali, T., Letovsky, S., Zheng, Y., Ding, C., Cantor, C. and Kasif, S. (2004) *Whole-genome annotation by using evidence integration in functional-linkage networks.* Proc. Natl. Acad. Sci., 101:2888-2893.

[9] Kim, W., Krumpelman, C., and Marcotte, E. (2008) *Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy.* Genome Biology, 9 (Suppl 1):S5.

[10] Kondor, R. and Lafferty, J. (2002) *Diffusion kernels on graphs and other discrete input spaces.* Proc Int Conf Machine Learning, 315-322.

[11] Lanckriet, R., Deng, M., Cristianini, N., Jordan, M. and Noble, W. (2004) *Kernel-based data fusion and its application to protein function prediction in yeast.* Proceedings of the Pacific Symposium on Biocomputing, January 3-8, 300-311.

[12] Lee, H., Tu, Z., Deng, M., Sun, F. and Chen, T. (2006) *Diffusion Kernel-based logistic regression models for protein function prediction.* OMICS, a Journal of Integrative Biology, 1(10):40-55.

[13] Marcotte, E., Pellegrini, M., Thompson, M., Yeates, T. and Eisenberg, D. (1999) *A combined algorithm for genome-wide prediction of protein function.* Nature, 402:83-86.

[14] Marcotte, E., Pellegrini, M., Ng, H., Rice, D., Yeates, T. and Eisenberg, D. (1999) *Detecting protein function and protein-protein interactions from genome sequences.* Science, 285:751-753.

[15] Nabieva, E., Jim, K., Agarwal, A., Chazelle, B. and Singh, M. (2005) *Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps.* Bioinformatics, 21(Suppl 1):302-310.

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

[17] Vazquez, A., Flammini, A., Maritan, A. and Vespignani, A. (2003) *Global protein function prediction from proteinĺCprotein interaction networks.* Nat. Biotechnol., 21:697-700.

[18] Sam, R. and Lawrence, S. (2000) *Nonlinear dimensionality reduction by locally linear embedding.* Science, 290:2323-2326.

[19] Watson, J., Laskowski, R. and Thornton, J. (2005) *Predicting protein function from sequence and structural data.* Current Opinion in Structural Biology, 15:275-284.

[20] Zhao, X., Wang, Y. Chen, L. and Aihara, K. (2008) *Gene function prediction using labeled and unlabeled data.* BMC Bioinformatics, 9:57.