One-Class Support Vector Machines for Protein-Protein Interactions Prediction

Hany Alashwal, Safaai Deris and Razib M. Othman

***Abstract*—**Predicting protein-protein interactions represent a key step in understanding proteins functions. This is due to the fact that proteins usually work in context of other proteins and rarely function alone. Machine learning techniques have been applied to predict protein-protein interactions. However, most of these techniques address this problem as a binary classification problem. Although it is easy to get a dataset of interacting proteins as positive examples, there are no experimentally confirmed non-interacting proteins to be considered as negative examples. Therefore, in this paper we solve this problem as a one-class classification problem using one-class support vector machines (SVM). Using only positive examples (interacting protein pairs) in training phase, the one-class SVM achieves accuracy of about 80%. These results imply that protein-protein interaction can be predicted using one-class classifier with comparable accuracy to the binary classifiers that use artificially constructed negative examples.

***Keywords*—**Bioinformatics, Protein-protein interactions, One-Class Support Vector Machines

I. Introduction

The completion of the Human Genome Project (HGP) (1990-2003) brought a revolution in biological and bioinformatics research. Currently, researchers have in hand the complete DNA sequences of genomes for many organisms—from microbes to plants to humans. Proteomics research is emerging as the "next step" of genomics.

The proteomics research is extensively concerned with the elucidation of the structure, interactions, and functions of proteins that constitute cells and organisms. Genomics research has already produced a massive quantity of molecular interaction data, contributing to maps of specific cellular networks. In fact, large-scale attempts have explored the complex network of protein interactions in the Saccharomyces cerevisiae [1] - [3].

Meanwhile, the recent studies of proteomics and molecular biology led the researchers to recognize that protein-protein interactions (PPI) affect almost all processes in a cell [4], [5]. It has been reported that even simple single-celled organisms such as yeast have about 6000 proteins interact by at least three interactions per protein, i.e. a total of 20,000 interactions or more [6]. It is also estimated that, there may be nearly 100,000 interactions in the human body.

Prediction of protein–protein interaction is an important problem because it helps to understand the basis of cellular operations and other functions. It has been shown that proteins with similar functions are more likely to interact [5]. If the function of one protein is known then the function of its binding partners is likely to be related. This helps to understand the functional roles of unannotated protein by knowing its interaction partners. Drug discovery is another area where protein–protein interaction prediction plays an important role.

For that reasons, identifying protein-protein interactions represents a crucial step toward understanding proteins functions. In the last few years, the problem of computationally predicting protein-protein interactions has gain a lot of attention. Methods based on the machine learning theory have been proposed [7]-[9]. Most of these methods address this problem as a binary classification problem. Although, constructing a positive dataset (i.e. pairs of interacting proteins) is relatively an easy task by using one of the available databases of interacting proteins, there is no data on experimentally confirmed non-interacting protein pairs have been made available. To cope with this problem, some researchers created an artificial negative protein interaction dataset for *S. cerevisiae* by randomly generating 100,000 protein pairs from this organism that are not described as interacting in the Database of Interacting Proteins (DIP) [10] without putting any further restrictions on such pairs, as in [11].

Since only data of interacting proteins pairs (positive data) are available and sampled well, the problem of predicting protein-protein interactions is essentially a one class classification problem. In this respect, we propose a recent method, one-class support vector machines (OCSVM) for protein-protein interactions predictions.

Manuscript received July 16, 2006. This work was supported in part by the Ministry of Science, Technology and Environment, Malaysia, under Grant 74289.

Hany Alashwal is a Ph.D. candidate at the Faculty of Computer Science and Information Systems, Univeristi Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, (phone: +607-5537791; fax: +607-5565044; e-mail: hany@siswa.utm.my).

Safaai Deris is a Prof. at the Software Engineering Department at the Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, (e-mail: safaai@fsksm.utm.my).

Razib M. Othman is a lecturer at the Software Engineering Department at the Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia, (e-mail: razib@fsksm.utm.my).

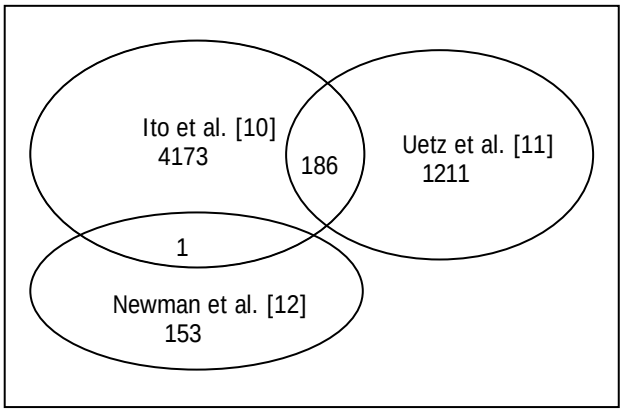

Fig. 1 Protein-protein interactions as detected in independent studies

II. Related Works

Most of the protein-protein interactions data were identified by high-throughput technologies like the yeast two-hybrid system, which are known to yield many false positives [12]. The comparison of the main three high-throughput datasets [1]–[3], shows that the overlap of the detected interactions obtained in these three studies is very small (Fig. 1). The numbers shown in Fig. 1 are the numbers of proteins interactions. Each oval represents a high throughput study, and the overlaps between the studies are given at the intersections. In addition to the problem of false positive in the high-throughput technology, the *in vivo* small scale experiments that identify protein-protein interaction are still time-consuming and labor-intensive; besides, they identify a small number of interactions. As a result, methods for computational prediction of protein-protein interactions based on sequence information are becoming increasingly important.

Several well-known computational methods of predicting protein–protein interaction have been studied and described in [13]. An important computation method based on phylogenetic profiles uses similarity of genes to predict the interactions [14], [15], where the similarity of genes is calculated based on presence or absence of genes in different species. Different method based on conservation of gene neighborhoods was employed in bacteria for prediction based on adjacency of genes in different species [16]. However, one of the problems that exist in the previous methods is that these methods need complete genomes for many species to produce good results. Gene fusion method traces a single protein in other domains where the interacting proteins are same at some point [17]. However, this method is only applicable to proteins with shared domains.

There are many protein sequence features that can be used to facilitate the computational prediction of protein-protein interactions (e.g. domain structure, amino acids hydrophobicity and sub-cellular localization). The most common sequence feature that has been used for this purpose is the protein domains structure. The motivation for this choice is that molecular interactions are typically mediated by a great variety of interaction domains [18]. Therefore it is logical to assume that the patterns of domain occurrence in interacting proteins provide useful information for training the prediction methods. In a recent study [19], the notion of potentially interacting domain pair (PID) was introduced to describe domain pairs that occur in interacting proteins more frequently than would be expected by chance. Accordingly, the protein domain structure is informative enough to facilitate the computational prediction of protein-protein interactions.

From the literature, it is noticeable that most of the work that has been done to solve protein-protein interactions prediction problem considers it a binary classification problem. However, this assumption is not reflecting the reality of the problem where only data of interacting proteins pairs (positive dataset) are available and sampled well [10] and so far there is no data on experimentally confirmed non-interacting protein pairs have been made available [20]. Many researchers cope with this difficulty by artificially creating a negative protein interactions dataset using randomly generated protein pairs that are not described as interacting in the databases of interacting proteins without putting any further restrictions on such pairs [7]–[9]. One problem with this approach is that in many cases selected “non-interacting” protein pairs will possess features that are substantially different from those typically found in the positive interaction set. This effect may simplify the learning task and artificially raise classification accuracy for training data. There is no guarantee, however, that the generalized classification accuracy will not degrade if the predictor is presented with new, previously unseen data which are hard to classify.

The Support Vector Machines (SVM), which can perform binary classification and regression estimation tasks, have been commonly used as a binary classifier to predict protein-protein interactions [7]–[9]. SVM were first proposed by Vapnik [21] and have recently been used in a range of problems including pattern recognition, bioinformatics, and text categorization. Schölkopf in [22] points out that a particular advantage of SVM over other learning algorithms is that it can be analyzed theoretically using concepts from computational learning theory and at the same time can achieve good performance when applied to real problems.

The SVM classifies data with different class labels by determining a set of support vectors that are members of the set of training inputs that outline a hyperplane in the feature space. The algorithm is chosen in such a way to maximize the distance from the closest patterns, which is called the margin. SVM aims to minimize an upper bound of generalization error through maximizing the margin between the separating hyperplane and data. On the contrary, the traditional methods minimize the empirical training error by mapping the input data space to high-dimensional feature data set and apply the structure risk minimization [23]. SVM has a good performance of classification in large data set and complex patterns processing such as text categorization [24], face detection [25], and object detection in machine vision [26].

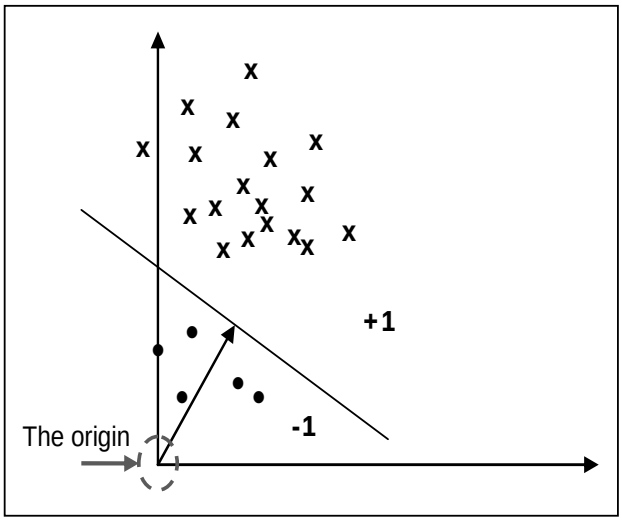

Fig. 2 Classification in one-class SVM

SVM has the following advantages to process biological data [27]: (i) SVM is computationally efficient and it is characterized by fast training which is essential for high-throughput screening of large protein datasets. (ii) SVM is readily adaptable to new data, allowing for continuous model updates in parallel with the continuing growth of biological databases. (iii) SVM provides a principled means to estimate generalization performance via an analytic upper bound on the generalization error. This means that a confidence level may be assigned to the prediction, and avoids problems with overfitting inherent in neural network function approximation.

In this paper we propose to solve the problem of predicting protein-protein interactions as a one-class classification problem. In this respect we propose a recent method, one-class support vector machines (SVM) for predicting protein-protein interactions.

III. One-Class Support Vector Machines

One-class classification problem is a special case from the binary classification problem where only data from one class are available and sampled well. This class is called the target class. The other class which is called the outlier class, can be sampled very sparsely, or can be totally absent. It might be that the outlier class is very hard to measure, or it might be very expensive to do the measurements on these types of objects. For example, in a machine monitoring system where the current condition of a machine is examined, an alarm is raised when the machine shows a problem. Measurements on the normal working conditions of a machine are very cheap and easy to obtain. On the other hand, measurements of outliers would require the destruction of the machine in all possible ways. It is very expensive, if not impossible, to generate all faulty situations [28]. Only a method trained on just the target data can solve the monitoring problem.

Basically, one-class SVM treats the origin as the only member of the second class (see Fig. 2). Then using relaxation parameters, it separates the members of the one class from the origin. Then the standard binary SVM techniques are employed.

The OCSVM algorithm maps input data into a high dimensional feature space (via a kernel) and iteratively finds the maximal margin hyperplane which best separates the training data from the origin. The OCSVM may be viewed as a regular two-class SVM where all the training data lies in the first class, and the origin is taken as the only member of the second class. Thus, the hyperplane (or linear decision boundary) corresponds to the classification function:

$$f(x) = \langle w, x \rangle + b \tag{1}$$

where w is the normal vector and b is a bias term. The OCSVM solves an optimization problem to find the function f with maximal geometric margin. We can use this classification function to assign a label to a test example x. If $f(x) < 0$ we label x as an anomaly, otherwise it is labeled normal.

Using kernels, solving the OCSVM optimization problem is equivalent to solving the following dual quadratic programming problem:

$$\min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{2}$$

$$\text{Subject to} \quad 0 \le \alpha_i \le \frac{1}{\nu l}, \quad \text{and} \quad \sum_i \alpha_i = 1 \tag{3}$$

where α_i is a Lagrange multiplier (or "weight" on example i such that vectors associated with non-zero weights are called "support vectors" and solely determine the optimal hyperplane), ν (*nu*), is a parameter that controls the trade-off between maximizing the distance of the hyperplane from the origin and the number of data points contained by the hyperplane, l is the number of points in the training dataset, and $K(x_i, x_j)$ is the kernel function. By using the kernel function to project input vectors into a feature space, we allow for nonlinear decision boundaries. Given a feature map:

$$\phi : X \to \Re^N \tag{4}$$

where ϕ maps training vectors from input space X to a high-dimensional feature space, we can define the kernel function as:

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{5}$$

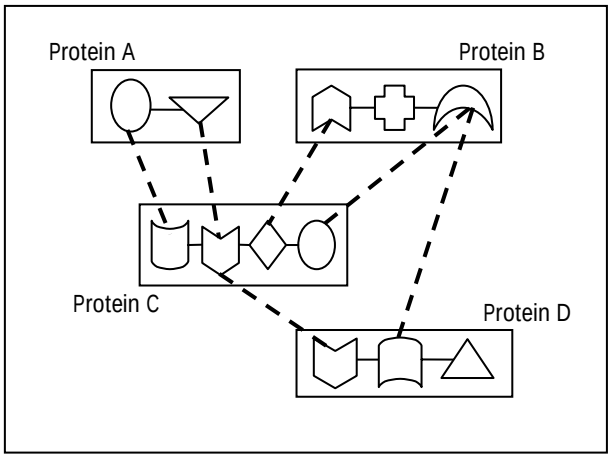

Fig. 3 Domains mediate protein-protein interactions

Feature vectors need not to be computed explicitly, and in fact it greatly improves computational efficiency to directly compute kernel values $K(x_i, x_j)$.

IV. Proteins Feature Representation

The construction of an appropriate feature space that describes the training data is essential for any supervised machine learning system. In the context of protein-protein interactions, it is believed that the likelihood of two proteins to interact with each other is associated with their structural domain composition [18], [19], [29]. For this reason, this study used the domain structure as protein features to facilitate the prediction of protein-protein interactions.

Within a protein, a structural domain (simply called "domain") is an element of overall structure that is self-stabilizing and often folds independently of the rest of the protein chain. Many domains are not unique to the proteins produced by one gene or one gene family but instead appear in a variety of proteins. Domains often are named and singled out because they play an important role in the biological function of the protein they belong to; for example, the *calcium-binding* domain of *calmodulin*.

Domains sometimes act completely independently of each other, as in the case of a catalytic domain and a binding domain, where the two domains don't interact with each other, but their association is synergistically because the linker between them means that the catalytic domain is kept in close contact to its substrate. In other cases structural interactions between domains do occur. In this case, the interaction between the domains should be considered as something akin to quaternary structure, rather that treating the whole complex as a single protein.

Fig. 3 illustrates the idea of potentially interacting domain pairs. As depicted in Fig. 3, domain combination pair based approach considers the interactions of domains as the mediator for protein-protein interactions. There exist multiple possible choices for the interaction of domains or domain combinations that can be inferred from a protein interaction, with only the interaction information of two proteins.

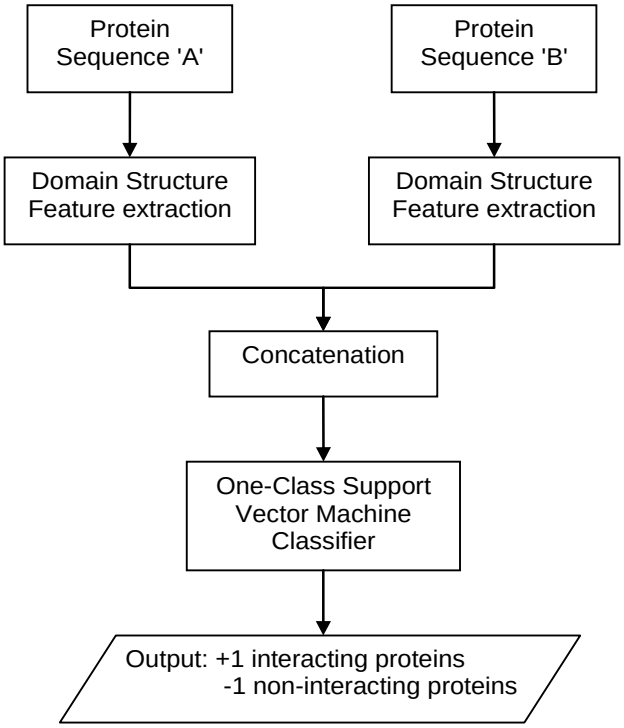

Fig. 4 Overview of One-Class SVM based protein–protein interaction prediction method.

In this study, the domain data was retrieved from the PFAM database [30]. PFAM is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models. The current version 10.0 contains 6190 fully annotated PFAM-A families. PFAM-B provides additional PRODOM-generated alignments of sequence clusters in SWISSPROT and TrEMBL that are not modeled in PFAM-A.

When the domain information is used, the dimension size of the feature vector becomes the number of domains appeared in all the yeast proteins. The feature vector for protein *p* was thus formulated as:

$$\boldsymbol{x} = [d_1, d_2, \ldots, d_i, \ldots, d_n] \quad (6)$$

where $d_i = m$ when the protein *p* has *m* pieces of domain d_i, and $d_i = 0$ otherwise. This formula allows the effect of multiple domains to be taken into account.

V. Materials and Implementation

An overview of the one-class SVM classifier for predicting protein–protein interaction is shown in Fig. 4. Experimentally found protein interactions obtained from Database of Interacting Proteins (DIP) are used for training the one-class SVM classifier. Interaction partners, 'protein A' and 'protein B', are converted to feature vectors based on domain structure. Then, predicting if two proteins can interact is done by passing their feature vectors into the one-class SVM

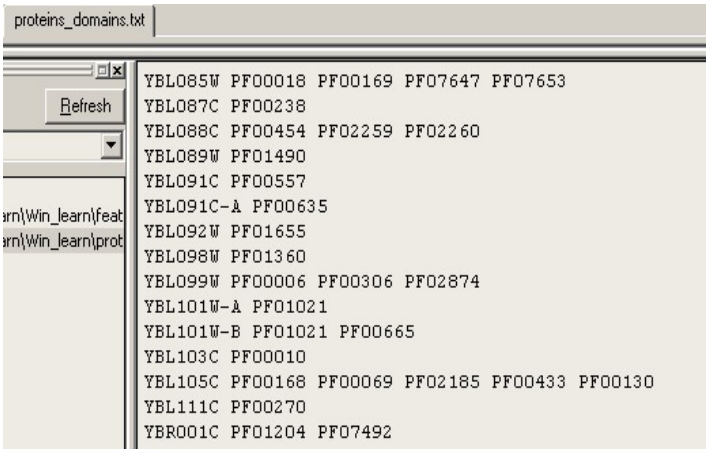

Fig. 5 An example of protein domains structure of the yeast genome

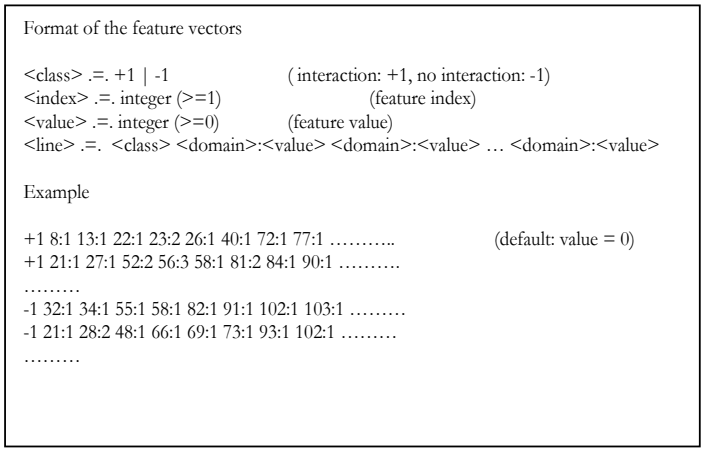

Fig. 6 Feature vectors format

classifier which generates the prediction output.

A. Data Sets

The protein interaction data was obtained from the Database of Interacting Proteins (DIP) [7]. The DIP database was developed to store and organize information on binary protein–protein interactions that was retrieved from individual research articles. The DIP database provides sets of manually compiled protein-protein interactions in Saccharomyces cerevisiae.

The majority of DIP entries are obtained from combined, non-overlapping data mostly obtained by systematic two-hybrid analyses. The current version contains 4749 proteins involved in 15675 interactions for which there is domain information. DIP also provides a high quality core set of 2609 yeast proteins that are involved in 6355 interactions which have been determined by at least one small-scale experiment or at least two independent experiments and predicted as positive by a scoring system [8].

The proteins sequences files were obtained for the Saccharomyces Genome Database (SGD) [31]. The SGD project collects information and maintains a database of the molecular biology of the yeast Saccharomyces cerevisiae. This database includes a variety of genomic and biological information and is maintained and updated by SGD curators. The proteins sequence information is needed in this research in order to elucidate the domain structure of the proteins involved in the interaction and to represent the amino acid hydrophobicity in the feature vectors.

B. Data Pre-processing

Since proteins domains are highly informative for the prediction of protein-protein interaction, we used the domain structure of a protein as the main feature of the protein sequence. We focused on domain data retrieved from the PFAM database which is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models. In order to elucidate the PFAM domain structure in the yeast proteins, we first obtain all sequences of yeast proteins from SGD. Given that sequence file, we then run InterProScan [32] to examine which PFAM domains appear in each protein. We used the stand-alone version of InterProScan. From the output file of InterProScan, we list up all PFAM domains that appear in yeast proteins and index them. Fig. 5 shows an example of protein domains that appears in yeast genome. The first column represents a protein whereas the following columns represent the domains that appear in the protein. The order of this list is not important as long we keep it through the whole procedure. The number of all domains listed and indexed in this way is considered the dimension size of the feature vector, and the index of each PFAM domain within the list now indicates one of the elements in a feature vector.

The next step is to construct a feature vector for each protein. For example, if a protein has domain D1 and D2 which happened to be indexed 17 and 64 respectively in the above step, then we assign "1" to the 17th and 64th elements in the feature vector, and "0" to all the other elements. Also if the domain D1 appears three times then we assign "3" to the 17th element in the feature vector and so on. Next we focus on the protein pair to be used for SVM training and testing. The assembling of feature vector for each protein pair can be done by concatenating the feature vectors of proteins constructed in the previous step. Fig. 6 shows the format of the feature vectors to be used by SVM. In this problem there are only two classes, +1 for interacting proteins and -1 for non-interacting proteins.

In the case of one-class SVM, only positive data was used in the training phase. The classifier should then be used to predict protein-protein interactions from a set of unknown protein pairs. However for testing purpose, we separated a part of the training data to be considered unknown to the classifier. This testing data was also combined with a similar number of random protein pairs that are not included in the DIP.

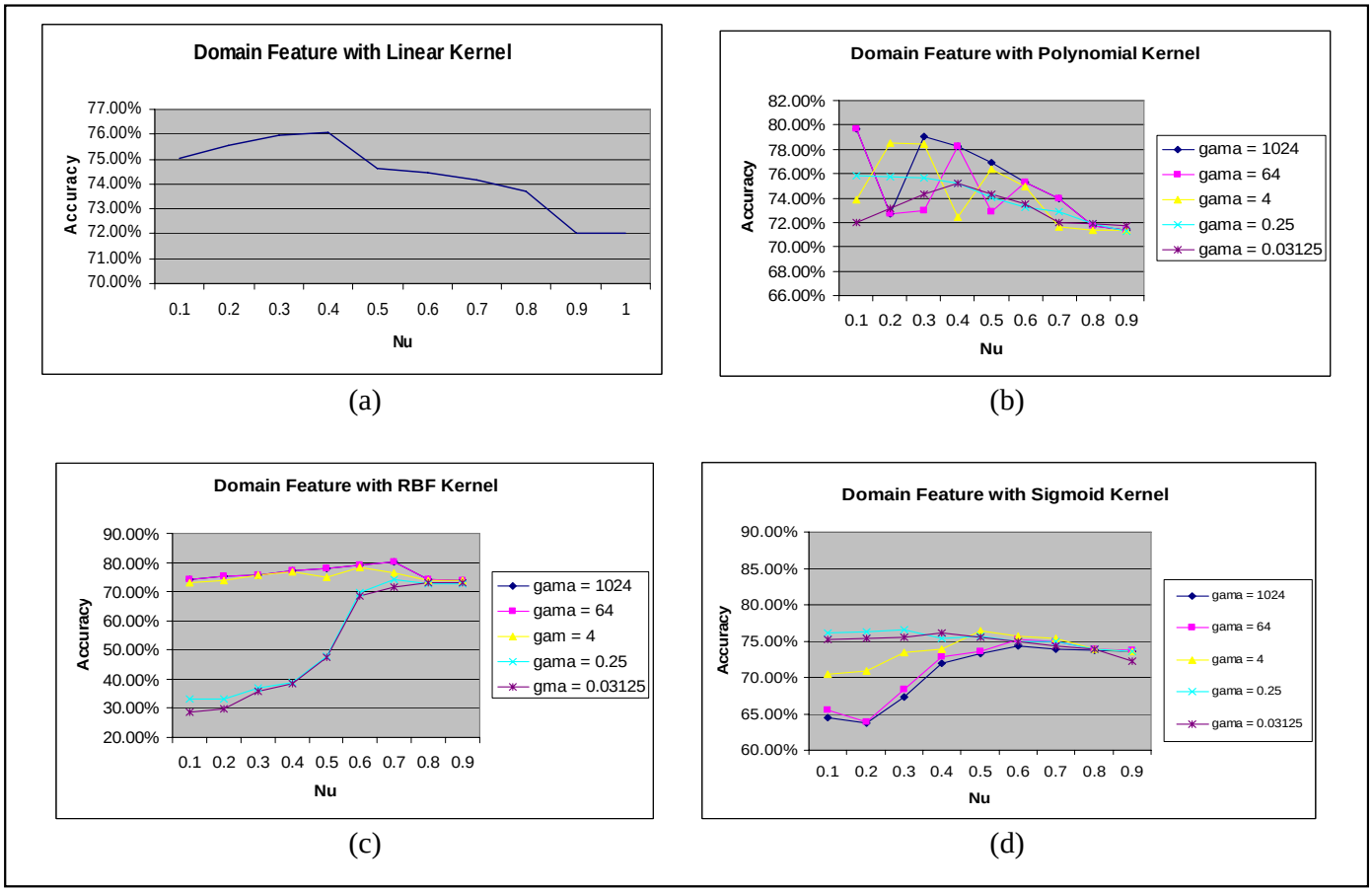

Fig. 7 One-class SVM performance for proteins interactions using different kernels

VI. Results and Discussion

We developed programs using Perl for parsing the DIP databases, sampling of records and sequences, and replacing amino acid sequences of interacting proteins with its corresponding feature. To make a positive interaction set, we represent an interaction pair by concatenating feature vectors of each proteins pair that are listed in the DIP-CORE as interacting proteins. Since we use domain feature we include only the proteins that have structure domains. The resulting positive set for domain feature contains 1879 protein pairs.

In our computational experiment, we employed the LIBSVM [33] (version 2.5) software and modified it to train and test the one-class SVM proposed in this paper. This is an integrated software tool for support vector classification, regression, and distribution estimation, which can handle one-class SVM. In order to train our one-class SVM, we examine out the following four kernels find appropriate parameter values:

- Linear: $K(x_i, x_j) = x_i^T x_j$.
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0.$
- Radial basis Function (RBF): $K(x_i, x_j) = \exp(\gamma \|x_i - x_j\|^2), \gamma > 0.$
- Sigmoid: $K(x_i, x_j) = tahn(\gamma x_i^T x_j + r).$

where γ (*gama*), r, and d are kernel parameters to be set for a specific problem. We carried out our experiments using the above mentioned kernels.

The results of our experiments are summarized in Fig. 7. These results indicate that it is informative enough to consider the existence of domains structure in the protein pairs to facilitate the prediction of protein-protein interactions. These results also indicate that the difference between interacting and non-interacting protein pairs can be learned from the available data using one-class classifier. It is also important to note that the choice of the parameters has a clear impact on the classifier performance.

Appropriate parameters for one-class SVM with different four kernels are set by the cross-validation process. We can see from this validation process that it is important to choose the appropriate parameters. As shown in Fig. 7, one-class SVM is very sensitive to the choice of parameters. However, since one-class SVM with linear kernel does not have the parameter *gama*, we executed the cross-validation process

only for parameter *nu*. Then the cross-validation accuracy is calculated in each run as the number of corrected prediction divided by the total number of data ((TP+TN)/(TP+FP+TN+FP)). Then the average is calculated for the 10 folds.

The best results were found by the RBF kernel (Fig. 7 (c)). Even though, RBF kernel could give as low accuracy as 29% with unsuitable choice of parameters, it achieves around 80% with proper choice of parameters. These results are comparable to the results that have been obtained by [6], [8] with slightly better accuracy. However, [5] reported accuracy of 94% using hydrophobicity as the protein feature. The reason behind this big difference between our result and their results lies in the approach of constructing the negative interaction dataset. They assign random value to each amino acid in the protein pair sequence. This leads to get new pairs that considered negative interacting pairs and greatly different from the pairs in the positive interaction set. This leads to simplify the learning task and artificially raise classification accuracy for training data. There is no guarantee, however, that the generalized classification accuracy will not degrade if the predictor is presented with new, previously unseen data which are hard to classify. In our work we used only positive data in the training set. In this case we don't need any artificially generated negative data for the training phase. We believe this approach will make the learning problem more realistic and ensure that our training accuracy better reflects generalized classification accuracy.

VII. Conclusion

The problem of predicting protein-protein interactions possesses the features of one-class classification problem where only data from target class (i.e. interacting proteins) are available and sampled well. Therefore, in this paper we have presented one-class SVM that find maximum margin hyperplanes in a high-dimensional feature space, emulating Vapnik's SVM. The objective of this paper was to show that the one-class SVM method can be applied successfully to the problem of predicting protein-protein interactions. Experiments performed on real dataset show that the performance of this method is comparable to that of normal binary SVM using artificially generated negative set. Of course, the absence of negative information entails a price, and one should not expect as good results as when they are available. In conclusion the result of this study suggests that protein-protein interactions can be predicted from domain structure with reliable accuracy. Consequently, these results show the possibility of proceeding directly from the automated identification of a cell's gene products to inference of the protein interaction pairs, facilitating protein function and cellular signaling pathway identification.

References

[1] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki, "Toward a protein-protein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins," Proc. Natl. Acad. Sci. USA. 97: 1143-1147, 2000.

[2] P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, R.S. Judson, J.R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, et al., "A Comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae," *Nature* 403:623 627, 2000.

[3] J. R. Newman, E. Wolf, and P. S. Kim, "A computationally directed screen identifying interacting coiled coils from *Saccharomyces cerevisiae*," *Proc. Natl. Acad. Sci*. U. S. A. 97, 13203–13208, 2000.

[4] H. Lodish, A. Berk, L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, *Molecular cell biology* (4th edition). W.H. Freeman, New York, 2000.

[5] B. Alberts, A. Johnson, J. Lewis, M. Raff, K.Roberts, and P. Walter, *Molecular Biology of the Cell* (4th edition). Garland Science, 2002.

[6] P. Uetz and C. S. Vollert, "Protein-Protein Interactions," *Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine* (ERGPMM), Springer Verlag, 2005.

[7] J. R. Bock and D. A. Gough, "Predicting protein-protein interactions from primary structure," *Bioinformatics*, vol. 17(5), pp: 455-460, 2001.

[8] Y. Chung, G. Kim, Y. Hwang, and H. Park, "Predicting Protein-Protein Interactions from One Feature Using SVM," *In proceedings of IEA/AIE*'04, pp:50-55, 2004.

[9] S. Dohkan, A. Koike and T. Takagi, "Prediction of protein-protein interactions using Support Vector Machines," *In Proceedings of the Fourth IEEE Symposium on BioInformatics and BioEngineering* (BIBE2004), Taitung, Taiwan, 576-584, 2004

[10] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg, "DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions," *Nucleic Acids Research*, vol. 30(1), pp: 303- 305, 2002.

[11] C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg, "Protein interactions: two methods for assessment of the reliability of high throughput observations," *Molecular & Cellular Proteomics*, vol. 1(5), pp: 349-56, 2002.

[12] E. M. Phizicky and S. Fields, "Protein-protein interactions: Method for detection and analysis," *Microbiological Reviews*, pp.94-123, 1995.

[13] A. Valencia, F. Pazos, "Computational methods for the prediction of protein interactions," *Curr. Opin. Struct. Biol*. 12, pp: 368–373, 2002.

[14] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D. Eisenberg, T.O. Yeates, "Assigning protein functions by comparative genome analysis: protein phylogenetic profiles," *Proc. Natl. Acad. Sci*. USA 96, pp: 4285–4288, 1999.

[15] T. Gaasterland, M.A. Ragan, "Microbial genescapes: phyletic and functional patterns of ORF distribution among prokaryotes," *Microb. Comp. Genomics* 3 pp:199–217, 1998.

[16] J. Tamames, G. Casari, C. Ouzounis, A.Valencia, "Conserved clusters of functionally related genes in two bacterial genomes," *J. Mol. Evol*. 44 pp: 66–73, 1997.

[17] A.J. Enright, I. Iliopoulos, N.C. Kyrpides, C.A. Ouzounis, "Protein interaction maps for complete genomes based on gene fusion events," *Nature* 402 pp:86–90, 1999.

[18] T. Pawson and P. Nash, "Assembly of cell regulatory systems through protein interaction domains," *Science*, vol. 300, pp: 445-452, 2003.

[19] W. K. Kim, J. Park, and J. K. Suh, "Large scale statistical prediction of protein-protein interaction by potentially interacting domain (PID) pair," *Genome Informatics*, vol. 13, pp: 42-50, 2002.

[20] D. S. Han, H. S. Kim, W. H. Jang, S. D. Lee, "PreSPI: A Domain Combination Based Prediction System for Protein-Protein Interaction," *Nucleic Acids Research*, vol. 32, no. 21, pp: 6312-6320, 2004.

[21] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer. 1995.

[22] B. Schölkopf and A. Smola, *Learning with kernels—support vector machines, regularization, optimization and beyond*, Cambridge, MA: MIT Press, 2002.

[23] K. R. Müller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, 12(2), 181–201, 2001.

[24] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," *In Proceedings of ACM-CIKM98*, Washington, DC (pp. 148–155). 1998.

[25] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," *In 1997 Conference on computer vision and pattern recognition* (pp. 130–136). Puerto Rico: IEEE. 1997.
[26] D. Roobaert and V. M. Hulle, "View-based 3d-object recognition with support vector machines," *In 1999 IEEE workshop on neural networks for signal processing* (pp. 77–84). Madison, WI: IEEE. 1999.
[27] J. R. Bock and D. A. Gough, "Predicting protein-protein interactions from primary structure," *Bioinformatics*, vol. 17(5), pp: 455-460, 2001.
[28] H. J. Shin , D. H. Eom, S. S. Kim, "One-class support vector machines: an application in machine fault detection and classification," *Computers and Industrial Engineering*, vol. 48 n. 2, pp:395-408, 2005.
[29] S. K. Ng, Z. Zhang, S. H. Tan, and K. Lin, "InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes," *Nucleic Acids Research*, vol. 31, pp: 251–254, 2003.
[30] A. Bateman, L. Coin, R. Durbin, R.D. Finn, V. Hollich, G. Jones, S. Khanna, A. Marshall, S.E. Moxon, L.L. Sonnhammer, D.J. Studholme, C. Yeats, and S.R. Eddy, "The Pfam Protein Families Database," *Nucleic Acids Research Database Issue*. 32:D138-D141, 2004.
[31] E. L. Hong, R. Balakrishnan, K.R. Christie, M.C. Costanzo, S.S. Dwight, S.R. Engel, D.G. Fisk, et al., "Saccharomyces Genome Database" http://www.yeastgenome.org/, (25th Dec 2005).
[32] N. J. Mulder, R. Apweiler, T. K. Attwood, A. Bairoch, D. Barrell, A. Bateman, D. Binns, et al., "The InterPro Database brings increased coverage and new features," *Nucleic Acids Research*, vol. 31, pp: 315-318, 2003.
[33] C. C. Chang and C. J. Lin, "LIBSVM : a library for support vector machines," 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Hany Alashwal is a doctoral candidate at the Faculty of Computer Science and Information Systems, the Universiti Teknologi Malaysia. He received the MSc degree in Computer Science from the Universiti Teknologi Malaysia in 2003 and the BSc degree in Mathematics and Computer Science from the Cairo University, Egypt in 1998. His research interests include bioinformatics, prediction of protein-protein interactions based on protein sequence features, and computational methods such as support vector machines, genetic algorithms, and software agent.

Safaai Deris is a Professor of Artificial Intelligence and Software Engineering at the Faculty of Computer Science and Information Systems, Deputy Dean at the School of Graduate Studies, and Director of Laboratory of Artificial Intelligence and Bioinformatics at the Universiti Teknologi Malaysia. He received the MEng degree in Industrial Engineering, and the DEng degree in Computer and System Sciences, both from the Osaka Prefecture University, Japan, in 1989 and 1997 respectively. His recent academic interests include the application and development of intelligent techniques in planning, scheduling, and bioinformatics.

Razib M. Othman is a doctoral candidate at the Faculty of Computer Science and Information System, the Universiti Teknologi Malaysia. He received the BSc and MSc degrees in Computer Science both from the Universiti Teknologi Malaysia, in 1999 and 2003 respectively. Currently, he is working on his PhD in Computational Biology. He also has interests in artificial intelligence, software agent, parallel computing, and web semantics. In March 2005, he was awarded the Young Researcher award by the Malaysian Association of Research Scientists (MARS). Two of his inventions, software products named *2D Engineering Drawing Extractor* and *2D Design Structure Recognizer*, have won 5 awards at the 21st Invention and New Product Exposition held in Pittsburgh, USA including the Best Invention of the Pacific Rim, and a gold medal award at the 34th International Exhibition of Inventions of New Techniques and Products held in Geneva, Switzerland.