Predicting Protein-Protein Interactions as a One-Class Classification Problem

Hany Alashwal, Safaai Deris and Razib M. Othman
Artificial Intelligence and Bioinformatics Laboratory
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
Email: hany@siswa.utm.my ; safaai@fsksm.utm.my ; razib@fsksm.utm.my

Abstract

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 used to predict protein-protein interactions. However, most of these techniques address this problem as a binary classification problem. While it is easy to get a dataset of interacting protein as positive example, there is no experimentally confirmed non-interacting protein to be considered as a negative set. Therefore, in this paper we solve this problem as a one-class classification problem using One-Class SVM (OCSVM). Using only positive examples (interacting protein pairs) for training, the OCSVM achieves accuracy of 80%. These results imply that protein-protein interaction can be predicted using one-class classifier with reliable accuracy.

Keywords

One-Class Classifier, Support Vector Machine, Bioinformatics, Protein-Protein Interaction Prediction.

1.0 Introduction

The recent studies of molecular biology led to recognize that protein-protein interactions affect almost all processes in a cell (Lodish et al., 2000; Alberts et al., 2002). It is estimated 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 (Uetz et al., 2005). It is also estimated that, there may be nearly 100,000 interactions in the human body.

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. Most of these methods consider this problem a binary classification problem. While, 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, nevertheless there is no data on experimentally confirmed non-interacting protein pairs have been made available. To cope with this problem, Deane et al., (2002) created a negative protein interaction data set 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) without putting any further restrictions on such pairs. 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 (OCSVMs) for protein-protein interactions predictions.

2.0 Related Works

Most of the interactions data was identified by high-throughput technologies like the yeast two-hybrid system, which are known to yield many false positives (Phizicky and Fields, 1995). In addition, in vivo 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.

The most common sequence feature 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 (Pawson and Nash, 2003). It is thus logical to assume that the patterns of domain occurrence in interacting proteins provide useful information for training PPI prediction methods. In a recent study, Kim et al. (2002) introduced the notion of potentially interacting domain pair (PID) to describe domain pairs that occur in interacting proteins more frequently than would be expected by chance.

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 data) is available and sampled well (Uetz et al., 2000; Ito et al., 2002). However, so far no data on experimentally confirmed non-interacting protein pairs have been made available (Huang et al., 2004). Many researchers tried to avoid this problem by creating a negative protein interaction data set by randomly generating protein pairs that are not described as interacting in the databases of interacting proteins without putting any further restrictions on such pairs (Deane et al., 2002; Chung et al., 2004; Dohkan et al., 2004). 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.

3.0 One-Class Support Vector Machines

One-class classification problem is a special 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 [Shin et al., 2005]. 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 (Figure 1). Then using relaxation parameters, it separates the members of the one class from the origin. Then the standard binary SVM techniques are employed.

The origin

Figure 1. Classification in one-class SVM.

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)=<w, x>+b (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 1 Sa ia jK (xi , xj) (2) a2 i , j

Subject to 0 = a i =1land

.

I ai =1 (3)

where ai 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), . (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 (xi , xj) 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:

f :X .RN (4)

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

K (xi , xj) = <f(xi),f(xj)> (5)

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

4.0 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 (Kim et al., 2002; Pawson & Nash, 2003; Ng et al.,2003). For these reasons, this study used the domain structure as protein features to facilitate the prediction of protein-protein interactions using the one-class SVM.

The domain data was retrieved from the PFAM database. PFAM is a reliable collection of multiple sequence alignments of protein families and profile hidden Markov models (Bateman et al., 2004). 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 each protein was thus formulated as:

x = [d1, d2, …, di, …, dn] (6)

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

5.0 Materials and Implementation

5.1 Data sets

We obtained the protein interaction data from the Database of Interacting Proteins (DIP; http://www.dip.doe-mbi.ucla.edu/). 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 (Deane et al., 2001).

The proteins sequences files were obtained for the Saccharomyces Genome Database (SGD; http://www.yeastgenome.org/). 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.

5.2 Data Preprocessing

Since proteins domains are highly informative for the protein-protein interaction, we used the domain structure of a protein as the main feature of the 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 (Mulder et al.,2003) 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. Figure 2 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 A and B which happened to be indexed 12 and 56 respectively in the above step, then we assign "1" to the 12th and 56th elements in the feature vector, and "0" to all the other elements. Also if the domain A appears three times then we assign "3" to the 12th 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. Figure 3 shows the format of the feature vectors to be used by SVM.

Figure 2. An example of protein domains structure of the yeast genome.

Format of the feature vectors

```
<class> .=. +1 | -1 ( interaction: +1, no interaction: -1)
<index> .=. integer (>=1) (feature index)
<value> .=. integer (>=0) (feature value)
<line> .=. <class> <domain>:<value> <domain>:<value> … <domain>:<value>
```

Example

```
+1 8:1 13:1 22:1 23:2 26:1 40:1 72:1 77:1 ……….. (default: value = 0)
+1 21:1 27:1 52:2 56:3 58:1 81:2 84:1 90:1 ………
.
……
…
-1 32:1 34:1 55:1 58:1 82:1 91:1 102:1 103:1 ……
…
-1 21:1 28:2 48:1 66:1 69:1 73:1 93:1 102:1 ……
…
……
…
```

Figure 3: Feature vectors format.

6.0 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 (version 2.5) software and modified it to train and test the one-class SVMs proposed in this paper. This is an integrated software tool for support vector classification, regression, and distribution estimation, which can handle one-class SVMs. The LIBSVM 2.5 is available at http://www.csie.ntu.edu.tw/wcjlin/libsvm. In order to train our one-class SVMs, we examine out the following four kernels find appropriate parameter values:

•
Linear: K(xi , xj)
=
xiTx j .
•
Polynomial: K(xi , xj)
=
(.xiTx j
+
r)d ,.>
0.
•
Radial basis Function (RBF):
2

K(xi , xj)
=
exp(
.

xi
-
xj

),.>
0.

•
Sigmoid: K(xi , xj) = tahn(.xiTx j + r). where
.
(gama), r, and d are kernel parameters to be set for a specific problem. We carried out our experiments using the above mentioned kernels.

Domain Feature with Linear Kernel
70.00%
71.00%
72.00%
73.00%
74.00%
75.00%
76.00%
77.00%
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
Nu
Accuracy
Domain Feature with Polynomial Kernel
66.00%
68.00%
70.00%
72.00%
74.00%
76.00%
78.00%
80.00%
82.00%
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Nu
Accuracy
gama = 1024
gama = 64
gama = 4
gama = 0.25
gama = 0.03125
(a) (b)
Domain Feature with RBF Kernel
20.00%
30.00%
40.00%
50.00%
60.00%
70.00%
80.00%
90.00%
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Nu
Accuracy
gama = 1024
gama = 64
gam = 4
gama = 0.25
gma = 0.03125
Domain Feature with Sigmoid Kernel
60.00%
65.00%
70.00%
75.00%
80.00%
85.00%

90.00%
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Nu
Accuracy
gama = 1024
gama = 64
gama = 4
gama = 0.25
gama = 0.03125
(c) (d)
Figure 4. One-class SVM performance using different kernels.

Appropriate parameters for one-class SVMs 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 Figure 4, OCSVM is very sensitive to the choice of parameters. However, since one-class SVMs 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 results of our experiments are summarized in Figure 4. 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.

These results are comparable to the results that have been obtained by Deane et al., (2002), Gomez et al.(2003), and Dohkan et al., (2004) with slightly better accuracy . However, Chung et al. (2004) reported accuracy of 94% by 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.

7.0 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 SVMs that find maximum margin hyperplanes in a high-dimensional feature space, emulating Vapnik's SVMs. 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

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

Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., Studholme, D.J., Yeats, C., & Eddy, S.R. (2004).

The Pfam: Protein Families Database", Nucleic Acids Research: Database Issue, 32, pp: D138D141.

Chung, Y., Kim, G., Hwang, Y., & Park, H., (2004). Predicting Protein-Protein Interactions from One Feature Using SVM. In proceedings of IEA/AIE 2004, pp:50-55

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

Dohkan, S., Koike, A. and Takagi, T. (2004) 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.

Gomez, S.M., Noble, W.S., & Rzhetsky, A. (2003). Learning to predict protein-protein interactions from protein sequences. Bioinformatics, 19(15), pp: 1875-1881.

Huang, Y., Frishman, D., Muchnik, I. (2004). Predicting Protein-Protein Interactions by a Supervised Learning Classifier. Computational Biology and Chemistry , 28, 4, 291-301.

Ito, T., Tashiro, K., Muta, S., Ozawa, R., Chiba, T., Nishizawa, M., Yamamoto, K., Kuhara, S. and Sakaki, Y. (2000). 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.

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

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

Mulder, et al. (2003).The InterPro Database brings increased coverage and new features. Nucleic Acids Research, 31, pp: 315-318.

Ng, S.K., Zhang, Z., Tan, S.H., & Lin, K. (2003). InterDom: a database of putative interacting protein domains for validating predicted protein interactions and complexes. Nucleic Acids Research, 31, pp: 251–254.

Pawson, T., & Nash, P. (2003). Assembly of cell regulatory systems through protein interaction domains. Science, 300, pp: 445-452.

Phizicky, E.M., & Fields, S. (1995). Protein-protein interactions: Method for detection and analysis. Microbiological Reviews, pp.94-123.

Shin, H.J., Eom D.H. and Kim S.S. (2005). One-class support vector machines: an application in machine fault detection and classification. Computers and Industrial Engineering, 48 (2), pp: 395–408

Uetz,P. et al. (2000) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature 403: 623-627

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