# Protein-Protein Interaction Document Mining

**Shing-Hwang Doong**[1] **Shu-Fen Hong**[1,2]

[1]Department of Information Management, ShuTe University
[2]Department of Computer Center, National KaoHsiung Marine University

## Abstract

Protein-protein interactions (*PPI*) are very important to the understanding of metabolic pathway. Many digital publications are available today; some of them discuss *PPI* and some of them do not. If machine learning techniques can be used to detect those *PPI* documents automatically, it would save researchers tremendous amount of time to construct a biological pathway. In this study, we analyze this document mining problem by using different kinds of feature representations and classification algorithms. Latent semantic indexing (*LSI*) and information gain (*IG*) were used to extract features from a document for classification, while support vector machine (*SVM*) and Naïve Bayesian (*NB*) were the selected algorithms. It is found that the combination of *LSI* and *SVM* provided the best solution.

**Keywords**: Latent semantic index, document mining, support vector machine, information gain, protein-protein interaction

## 1. Introduction

Since the completion of human genome sequencing project, system biology becomes a very important field in bioinformatics. Establishing various pathways of bio-molecules (DNA, RNA and protein) is the ultimate goal of system biology, and protein-protein interactions are key ingredients in the understanding of these pathways. Protein-protein interactions can be studied from a biological perspective. For example, Aytuna et al. [1] used protein sequences and structures to predict interactions. Nowadays, many biomedical documents discussing specific protein relationships are available in the digital format. These documents summarize experimental results from labs around the world. Thus, they represent the first hand information related to protein-protein interaction networks. But, it is very difficult, if not impossible, to extract required information from this ever growing set of biomedical documents. For example, the national center for biotechnology information (NCBI) has included more than 16 million citations in its PubMed service. How to efficiently and effectively extract *PPI* information from this vast amount of documents has become a meaningful and interesting job in document mining.

Prior studies have used information retrieval and information extraction techniques to help discover interesting biological facts. Ono et al. [8] employed a protein name dictionary, surface clues on word patterns and simple part-of-speech rules to extract information on *PPI* from scientific literature. They achieved a recall rate of 86.8% and a precision rate of 94.3% for yeast. Marcitte et al. [6] used a Bayesian approach to identify Medline abstracts as describing interactions between yeast proteins. More than 80 discriminating words (e.g. complex, interaction) were determined from a training set and used to score a log likelihood function. Donaldson et al. [4] adopted a support vector machine approach to mine the biomedical literature for *PPI*. They used a binary vector space model consisting of words and two-word phrases to represent a document. The (at most) 1500 words and phrases with the highest information gain were retained for the final feature representation. The researchers have achieved a *PPI* classifying system with precision, accuracy and recall equal to 92%, 90% and 92% respectively. Finally, Homayouni et al. [5] adopted a different approach, namely the latent semantic indexing, for feature representation of Medline abstracts to cluster genes.

As can be seen from the above literature review, researchers have used various feature representations of documents and algorithms to mine biomedical literature for useful information. However, it seems that the combination of *LSI* for feature representation and *SVM* for classification has not been tried before for *PPI* document mining. In this study, we combine various feature representations and algorithms to mine biomedical literature for *PPI* documents.

## 2. Methods

Four protocols were analyzed in this study. They were formed by combining two feature representations (*LSI* and *IG*) and two classification algorithms (*SVM* and *NB*).

### 2.1. Latent Semantic Indexing

Documents written with natural language frequently come with the problem of polysemy and/or synonymy. In many cases, combinations of words will provide more discriminating power to classify documents. For example, Donaldson et al. [4] used two-word phrases in their PreBIND system to detect *PPI* documents. On the other hand, Berry et al. [2] used the singular value decomposition (*SVD*) theory from matrix computation to extract significant combinations of keywords.

Let $A(m, n)$ be a keyword-document matrix, where $m$ is the number of keywords and $n$ the number of documents. For example, $A$ could be obtained from the *tf-idf* vector space models for a set of documents [10]. According to the *SVD* theory, $A$ can be decomposed into

$$A = U\Sigma V^T \tag{1}$$

where $U$ and $V$ are orthogonal matrices obtained from the eigenvectors of $AA^T$ and $A^TA$ respectively, and $\Sigma$ is a diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_p = 0$, $p = \min(m, n)$ and $r$ is the rank of $A$. Each column of $U$ is a weighted combination of the original keywords, and the lefter a column is, the larger its associated singular value becomes. The *LSI* theory tries to use the first few columns (called latent semantic indexes) of $U$ and their associated singular values to approximate $A$ in (1). The rationale behind this approach is when complete latent semantic indexes are used, though equation (1) becomes an equality, noise resulted from natural language writing can disturb the classification process. Therefore, with a set of documents having 3000 keywords, one may request the largest 200 latent semantic indexes be used to represent a document. Each index becomes a weighted combination of 3000 keywords and each document is encoded with respect to these 200 indexes. Notice that it is usually difficult to interpret the meaning of a latent semantic index.

## 2.2. Information Gain

Different keywords have different discriminating power to classify documents into *PPI* or non-*PPI* class. For example, ‘interact’ has significantly higher power than ‘extract’ in doing the job. Quinlan [9] used an entropy formula to compute the information gain for each keyword and selected keywords with highest information gain for feature representation.

*IG* approach has been used in many text classification problems due to its simplicity. However, when using *IG* to select features, one must be careful not to involve too much information in the computation of entropy. That is, the process should not include test documents for the computation of *IG* because the class labels for these documents are presumably unknown at this stage. On the other hand, the computation of *LSI* does not use the class label of these documents; therefore they can be combined with the training documents to compute the *SVD*. This is one advantage of *LSI* over *IG*.

## 2.3. Support Vector Machine

*SVM* is a family of machine learning algorithms which are based on statistical learning theory [3]. Because of its high performance, *SVM* is receiving the attentions of many researchers lately and has been successfully applied to many domains including hand written digit recognition, text categorization, bioinformatics, and so on. Training a *SVM* model is converted to the problem of finding an optimal separating hyperplane (*OSH*). An *OSH* has maximum margin separating opposite classes of training examples.

Another important aspect of *SVM* is that we can map original input vector into a higher dimensional feature space so that the classification problem can be easily done in the feature space. *SVM* uses a kernel function to handle this feature mapping. Many *SVM* tools provide three typical kernel functions for users - polynomial kernel, radial basis function kernel (*RBF*), and sigmoid kernel. Most researches use the *RBF* kernel because its function can substitute the other kernel functions.

## 2.4. Naïve Bayesian

A Naïve Bayesian [7] classifier is a probability based classifier. The *NB* approach for *PPI* detection is to find which keyword features are present or absent with prior probabilities in a document. Then, the Bayes’ theorem is used to calculate the posterior probability that a document belongs to the *PPI* class after observing data in the training set.

## 2.5. Assessment

The accuracy (A), recall (R), precision (P) and F measure (FM) are used to assess the performance of various protocols in this study.

$$\text{A} = (TP+TN) / (TP+FP+FN+TN), \tag{2}$$

$$\text{P} = TP / (TP+FP), \tag{3}$$

$$\text{R} = TP / (TP+FN), \tag{4}$$

$$\text{FM} = 2\,PR / (P+R) \tag{5}$$

*TP* is the number of *PPI* documents that have been correctly predicted as *PPI*; *FP* the number of non-*PPI* documents incorrectly predicted as *PPI*; *TN* the number of non-*PPI* documents correctly predicted as non-*PPI*, and *FN* the number of *PPI* documents incorrectly predicted as non-*PPI*.

In addition, receiver operating characteristic (*ROC*) curve is computed to further compare different protocols. In sketching the *ROC* curve, the sensitivity rate and the specificity rate are computed for various parameter values. In this study, the number of *IG* keywords (or *LSI* indices) is the variable parameter that can take the values of 100, 200, 300, 400 and 500. The sensitivity rate is defined to be the recall rate of *PPI* documents, and the specificity rate is the recall rate of non-*PPI* documents. A *ROC* curve is the curve connecting (sensitivity, 1- specificity) data points over the various parameter values. The more up and left a *ROC* curve, the better its associated protocol.

## 3. Results

Experimental data was selected from the PreBIND database (http://bind.ca) and NCBI PubMed service. 500 records were randomly selected from PreBIND's "yeast_point_170504.txt" file with a PISCORE > 0. A record in this file includes PMID (PubMed ID), two yeast proteins that appear in the corresponding PubMed abstract, and a PISCORE indicating interaction possibility of these two proteins. A condition of PISCORE > 0 indicates that the abstract includes at least one sentence describing an interaction of proteins. On the other hand, a condition of PISCORE = 0 does not necessarily mean the abstract is non-*PPI*. For example, the interaction relationship may be described in two different sentences using demonstrative pronouns. Therefore, we manually read those abstracts with PISCORE = 0 to determine their classes. From this set of documents, 100 abstracts discussing *PPI* and 400 abstracts not discussing *PPI* were selected. The final experimental data has 1000 abstracts including 600 *PPI* documents and 400 non-*PPI* documents. These abstracts were preprocessed with stop-words removal and stemming procedures. Keywords appearing in less than 3 abstracts were also dropped before further processing. This left us with 3762 keywords in the end. The *tf-idf* model was used for the initial feature representation [10].

Five fold cross validation was used to assess the four protocols. The experimental data set was first randomly partitioned into five parts with equal size. In each run of the cross validation, 800 abstracts were selected as the training data while the remaining 200 abstracts formed the test data. For the *IG* feature representation, *n* keywords with the highest information gains were selected for the final feature representations. Only abstracts from the training set were used to compute the information gains. For the *LSI* approach, the entire set of 1000 abstracts was used to find the *SVD* of the keyword-document matrix. Indices corresponding to the largest *n* singular values were used for the final feature representations. In both cases, *n* was 100, 200, 300, 400 and 500.

After abstracts were represented by the *IG* or *LSI* features, *NB* and *SVM* algorithms were subsequently applied to train and predict *PPI* documents. Results for the five fold cross validation are summarized in Tables 1 and 2. One can see that the combination of *LSI* and *SVM* (LSI_SVM) provided the best solution. With this protocol, the largest 200 *LSI* indices already yield a near 100% perfect performance.

In order to compare these four protocols, *ROC* curves were plotted for all protocols (Fig. 1). One can see that protocol 1 (LSI_SVM) provided the best performance, IG_SVM was the second best protocol, and the other two protocols were about the same since they enclosed about the same area.

## 4. Discussions

This study analyzed four protocols for mining *PPI* documents from biomedical abstracts. These protocols were formed by combining two feature representations (*LSI* and *IG*) for text and two classification algorithms (*SVM* and *NB*) from machine learning. It was found that the combination of *LSI* and *SVM* provided the best performance for the *PPI* detection work.

The *IG* approach for feature representation suffered from the absence of test documents in the computation of information gains for selecting final features. On the other hand, the working of *LSI* does not have this restriction. The *LSI* approach does require a little longer time to process the *SVD* computation.

On the algorithmic perspective, *SVM* has been shown to outperform *NB* in many applications. It is explainable that the combination of *LSI* and *SVM* has provided the best performance in this study. Our result from the LSI_SVM protocol has even outperformed the one from Ono et al. [8] or Donaldson et al. [4]. Future study can focus on how to handle the keyword-document matrix more efficiently when subjects of interest in the document set change as time moves forwards.

## Acknowledgement

This research was partly supported by a grant from National Science Council of Taiwan under contract number: NSC-94-2213-E-366-013.

## References

[1] A. Aytuna, A. Gursoy, O. Keskin, "Prediction of protein-protein interactions by combining structure and sequence conservation in protein

interfaces," *Bioinformatics*, Vol. 21-12, pp. 2850-2855, 2005.

[2] M. Berry, S. Dumais, and G. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM review*, Vol. 37:4, pp. 573-595, 1995.

[3] C. Cortes, and V. Vapnik, "Support vector networks," *Machine Learning*, Vol. 20, pp. 273-297, 1995.

[4] I. Donaldson, J. Martin, and B. de Bruijn, "PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine," *BMC Bioinformatics*, Vol. 4:11, 2003.

[5] R. Homayouni, K. Heinrich, L. Wei, and M. Berry, "Gene clustering by latent semantic indexing of MEDLINE abstracts," *Bioinformatics*, Vol. 21:1, pp. 104-115, 2005.

[6] E. Marcotte, I. Xenarios, and D. Eisenberg, "Mining literature for protein-protein interactions," *Bioinformatics*, Vol. 17:4, pp. 359-363, 2001.

[7] A. McCallum, and K. Nigam, "A comparison of event models for Naïve-Bayesian text classification," AAAI workshop on learning for text categorization, 1998.

[8] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature," *Bioinformatics*, Vol. 17:2, pp. 155-161, 2001.

[9] J. Quinlan, "Discovering rules from large collections of examples: a case study," *Expert systems in the micro-electronic age*, pp. 168-201, 1979.

[10] B. Ricardo, and R. Bethier, "Modern information retrieval," *Addison-Wesley*, May 1999.

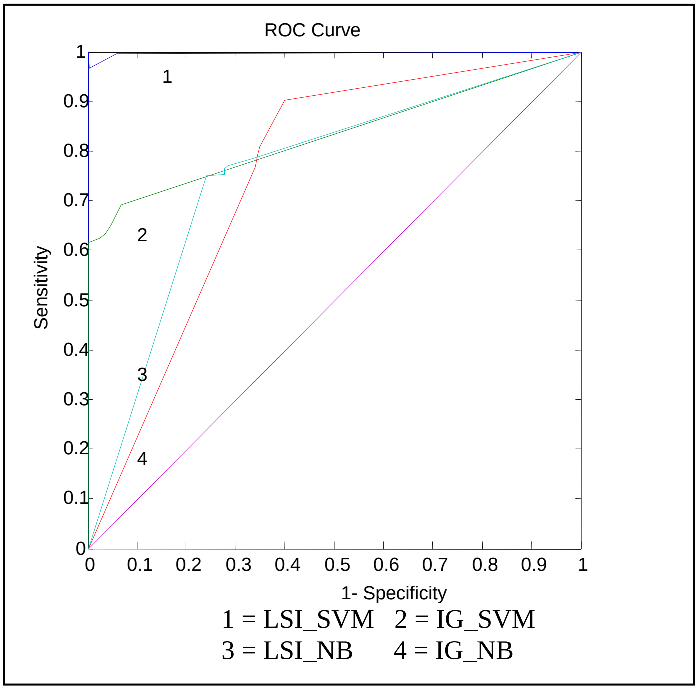

**Fig. 1: ROC curves for different protocols**

**Table 1: Summarized results for LSI feature representation**

| | LSI_SVM | | | | LSI_NB | | | |
|---|---|---|---|---|---|---|---|---|
| **# of features** | **A[%]** | **P[%]** | **R[%]** | **FM[%]** | **A[%]** | **P[%]** | **R[%]** | **FM[%]** |
| 100 | 95.6 | 96.01 | 96.65 | 96.33 | 72.00 | 59.83 | 90.20 | 71.94 |
| 200 | 99.7 | 99.83 | 99.67 | 99.75 | 73.00 | 76.83 | 77.87 | 77.35 |
| 300 | 100 | 100 | 100 | 100 | 72.00 | 76.67 | 76.67 | 76.67 |
| 400 | 100 | 100 | 100 | 100 | 73.30 | 75.50 | 79.06 | 77.24 |
| 500 | 100 | 100 | 100 | 100 | 74.10 | 74.67 | 80.72 | 77.58 |

**Table 2: Summarized results for IG feature representation**

| | IG_SVM | | | | IG_NB | | | |
|---|---|---|---|---|---|---|---|---|
| **# of features** | **A[%]** | **P[%]** | **R[%]** | **FM[%]** | **A[%]** | **P[%]** | **R[%]** | **FM[%]** |
| 100 | 72.73 | 98.34 | 69.20 | 81.24 | 73.58 | 77.11 | 78.55 | 77.82 |
| 200 | 67.73 | 99.33 | 65.17 | 78.70 | 75.00 | 84.33 | 76.44 | 80.19 |
| 300 | 65.33 | 100 | 63.40 | 77.60 | 74.10 | 84.33 | 75.41 | 79.62 |
| 400 | 63.84 | 99.83 | 62.43 | 76.82 | 76.70 | 87.17 | 77.03 | 81.79 |
| 500 | 62.64 | 99.83 | 61.66 | 76.23 | 74.10 | 85.33 | 74.96 | 79.81 |