Hypergraph and Protein Function Prediction with Gene Expression Data

Loc Hoang Tran
University of Minnesota/Computer Science Department, Minneapolis, USA
Email: tran0398@umn.edu

Linh Hoang Tran
Portland State University/ECE Department, Portland, USA
Email: linht@pdx.edu

***Abstract*—Most network-based protein (or gene) function prediction methods are based on the assumption that the labels of two adjacent proteins in the network are likely to be the same. However, assuming the pairwise relationship between proteins or genes is not complete. The information a group of genes that show very similar patterns of expression and tend to have similar functions (i.e. the functional modules) is missed. The natural way overcoming the information loss of the above assumption is to represent the gene expression data as the hypergraph. Thus, in this paper, the three un-normalized, random walk, and symmetric normalized hypergraph Laplacian based semi-supervised learning methods applied to hypergraph constructed from the gene expression data in order to predict the functions of yeast proteins are introduced. Experiment results show that the average accuracy performance measures of these three hypergraph Laplacian based semi-supervised learning methods are the same. However, their average accuracy performance measures of these three methods are much greater than the average accuracy performance measures of un-normalized graph Laplacian based semi-supervised learning method (i.e. the baseline method of this paper) applied to gene co-expression network created from the gene expression data.**

***Index Terms*—hypergraph Laplacian, protein, function, prediction, semi-supervised learning**

I. Introduction

Protein function prediction plays a very important role in modern biology. Detecting the function of proteins by biological experiments is very time-consuming and difficult. Hence a lot of computational methods have been proposed to infer the functions of the proteins by using various types of information such as gene expression data and protein-protein interaction networks [1].

The classical way predicting protein function infers the similarity to function from sequence homologies among proteins in the databases using sequence similarity algorithms such as FASTA [2] and PSI-BLAST [3]. Next, to predict protein function, graph which is the natural model of relationship between proteins or genes can also be employed. This model can be protein-protein interaction network or gene co-expression network. In this model, the nodes represent proteins or genes and the edges represent for the possible interactions between nodes. Then, machine learning methods such as Support Vector Machine [5], Artificial Neural Networks [4], un-normalized graph Laplacian based semi-supervised learning method [6], the symmetric normalized and random walk graph Laplacian based semi-supervised learning methods [7], or neighbor counting method [8] can be applied to this graph to infer the functions of un-annotated protein. The neighbor counting method labels the protein with the function that occurs frequently in the protein's adjacent nodes in the protein-protein interaction network and hence does not utilized the full topology of the network. However, the Artificial Neural Networks, Support Vector Machine, un-normalized, symmetric normalized and random walk graph Laplacian based semi-supervised learning method utilizes the full topology of the network. The Artificial Neural Networks and Support Vector Machine are all supervised learning methods. The neighbor counting method, the Artificial Neural Networks, and the three graph Laplacian based semi-supervised learning methods are all based on the assumption that the labels of two adjacent proteins in graph are likely to be the same. However, SVM do not rely on this assumption. Unlike graphs used in neighbor counting method, Artificial Neural Networks, and the three graph Laplacian based semi-supervised learning methods are very sparse, the graph (i.e. kernel) used in SVM is fully-connected.

The Artificial Neural Networks method is applied to the single protein-protein interaction network. However, the SVM method and three graph Laplacian based semi-supervised learning methods try to use weighted combination of multiple networks (i.e. kernels) such as gene co-expression network and protein-protein interaction network to improve the accuracy performance measures. [5] (SVM method) determines the optimal weighted combination of networks by solving the semi-definite problem. [6] (un-normalized graph Laplacian based semi-supervised learning method) uses a dual problem and gradient descent to determine the weighted combination of networks. [7] Uses the integrated network combined with equal weights, i.e. without optimization

Manuscript received July 1, 2013; revised February 13, 2014.

doi: 10.12720/joace.3.2.164-170

due to the integrated network combined with optimized weights has similar performance to the integrated network combined with equal weights and the high time complexity of optimization methods.

The un-normalized, symmetric normalized, and random walk graph Laplacian based semi-supervised learning methods are developed based on the assumption that the labels of two adjacent proteins or genes in the network are likely to be the same [6]. In this paper, we use gene expression data for protein function prediction problem. Hence this assumption can be interpreted as pairs of genes showing a similar pattern of expression and thus sharing edges in a gene co-expression network tend to have similar function. However, assuming the pairwise relationship between proteins or genes is not complete, the information a group of genes that show very similar patterns of expression and tend to have similar functions [8] (i.e. the functional modules) is missed. The natural way overcoming the information loss of the above assumption is to represent the gene expression data as the hypergraph [9], [10]. A hypergraph is a graph in which an edge (i.e. a hyper-edge) can connect more than two vertices. In [9], [10], the symmetric normalized hypergraph Laplacian based semi-supervised learning method have been developed and successfully applied to text categorization and letter recognition applications. To the best of my knowledge, the hypergraph Laplacian based semi-supervised learning methods have not yet been applied to protein function prediction problem. In this paper, we will develop the symmetric normalized, random walk, and un-normalized hypergraph Laplacian based semi-supervised learning methods and apply these three methods to the hypergraph constructed from gene expression data available from [11]. In the other words, the hypergraph is constructed by applying k-mean clustering method to this gene expression dataset.

We will organize the paper as follows: Section 2 will introduce the definition hypergraph Laplacians and their properties. Section 3 will introduce the un-normalized, random walk, and symmetric normalized hypergraph Laplacian based semi-supervised learning algorithms in detail. Section 4 will show how to derive the closed form solutions of normalized and un-normalized hypergraph Laplacian based semi-supervised learning algorithm from regularization framework. In section 5, we will apply the un-normalized graph Laplacian based semi-supervised learning algorithm (i.e. the current state of art method applied to protein function prediction problem) to gene co-expression network created from gene expression data available from [11] and compare its accuracy performance measure to the three hypergraph Laplacian based semi-supervised learning algorithms' accuracy performance measures. Section 6 will conclude this paper and the future direction of researches of this protein function prediction problem utilizing discrete operator of graph will be discussed.

II. Hypergraph Definitions

Given a hypergraph $G=(V,E)$, where V is the set of vertices and E is the set of hyper-edges. Each hyper-edge $e \in E$ is the subset of V. Please note that the cardinality of e is greater than or equal two. In the other words, $|e| \geq 2$, for every $e \in E$. Let $w(e)$ be the weight of the hyper-edge e. Then W will be the $R^{|E|\times|E|}$ diagonal matrix containing the weights of all hyper-edges in its diagonal entries.

A. Definition of Incidence Matrix H of G

The incidence matrix H of G is a $R^{|V|\times|E|}$ matrix that can be defined as follows

$$h(v,e) = \begin{cases} 1 \text{ if vertex } v \text{ belongs to hyperedge } e \\ 0 \text{ otherwise} \end{cases}$$

From the above definition, we can define the degree of vertex v and the degree of hyper-edge e as follows

$$d(v) = \Sigma_{e\in E} w(e) \times h(v,e)$$

$$d(e) = \Sigma_{v\in V} h(v,e)$$

Let D_v and D_e be two diagonal matrices containing the degrees of vertices and the degrees of hyper-edges in their diagonal entries respectively. Please note that D_v is the $R^{|v|\times|v|}$ matrix and D_e is the $R^{|e|\times|e|}$ matrix.

B. Definition of the Un-Normalized Hypergraph Laplacian

The un-normalized hypergraph Laplacian is defined as follows

$$L = D_v - HWD_e^{-1}H^T$$

C. Properties of L

1) For every vector $f \in R^{|V|}$, we have

$$f^T Lf = \frac{1}{2}\Sigma_{e\in E}\Sigma_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)}\left(f(u) - f(v)\right)^2$$

2) L is symmetric and positive-definite
3) The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector 1
4) L has $|V|$ non-negative, real-valued eigen values $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|V|}$

Proof:

1) We know that

$$\frac{1}{2}\Sigma_{e\in E}\Sigma_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)}\left(f(u) - f(v)\right)^2$$

$$= \frac{1}{2}\Sigma_{e\in E}\Sigma_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)}\left(f(u)^2 + f(v)^2 - 2f(u)f(v)\right)$$

$$= \Sigma_{e\in E}\Sigma_{u,v\in V} \frac{w(e)}{d(e)}\left(f(u)^2 - f(u)f(v)\right)h(u,e)h(v,e)$$

$$= \Sigma_{e\in E}\Sigma_{u\in V} w(e) f(u)^2 h(u,e) \Sigma_{v\in V} \frac{h(v,e)}{d(e)} - \Sigma_{e\in E}\Sigma_{u,v\in V} \frac{w(e)}{d(e)} f(u) f(v) h(u,e) h(v,e)$$

$$= \Sigma_{e\in E}\Sigma_{u\in V} w(e) f(u)^2 h(u,e) - \Sigma_{e\in E}\Sigma_{u,v\in V} \frac{w(e)}{d(e)} f(u) f(v) h(u,e) h(v,e)$$

$$= \Sigma_{u\in V} f(u)^2 \Sigma_{e\in E} w(e) h(u,e) - \Sigma_{e\in E}\Sigma_{u,v\in V} \frac{w(e)}{d(e)} f(u) f(v) h(u,e) h(v,e)$$

$$= \Sigma_{u\in V} f(u)^2 d(u) - \Sigma_{e\in E}\Sigma_{u,v\in V} \frac{w(e)}{d(e)} f(u) f(v) h(u,e) h(v,e)$$

$$= f^T D_v f - f^T H W D_e^{-1} H^T f$$

$$= f^T \left(D_v - H W D_e^{-1} H^T \right) f$$

$$= f^T L f$$

2) L is symmetric follows directly from its own definition.

Since for every vector $f \in R^{|v|}$,

$$f^T L f = \frac{1}{2} \Sigma_{e\in E} \Sigma_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)} (f(u) - f(v))^2 \geq 0$$

We conclude that L is positive-definite.

3) The fact that the smallest eigenvalue of L is 0 is obvious.

Next, we need to prove that its corresponding eigenvector is the constant one vector 1.

Let $d_v \in R^{|v|}$ be the vector containing the degrees of vertices of hypergraph G, $d_e \in R^{|E|}$ be the vector containing the degrees of hyper-edges of hypergraph G, $w \in R^{|E|}$ be the vector containing the weights of hyper-edges of G, $1 \in R^{|v|}$ be vector of all ones, and $one \in R^{|E|}$ be the vector of all ones. Hence we have

$$L1 = (D_v - H W D_e^{-1} H^T) 1 = d_v - H W D_e^{-1} d_e$$
$$= d_v - H W \text{one} = d_v - H w = d_v - d_v = 0$$

4) (4) follows directly from (1)-(3).

D. The Definitions of Symmetric Normalized and Random Walk Hypergraph Laplacians

The symmetric normalized hypergraph Laplacian (defined in [9], [10]) is defined as follows

$$L_{sym} = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$$

The random walk hypergraph Laplacian (defined in [9], [10]) is defined as follows

$$L_{rw} = I - D_v^{-1} H W D_e^{-1} H^T$$

E. Properties of L_{sym} and L_{rw}

1) For every vector $f \in R^{|v|}$, we have

$$f^T L_{sym} f = \frac{1}{2} \Sigma_{e\in E} \Sigma_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2$$

2) λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ is an eigenvalue of L_{sym} with eigenvector $w = D_v^{\frac{1}{2}} u$

3) λ is an eigenvalue of L_{rw} with eigenvector u if and only if λ and u solve the generalized eigen-problem $L_u = \lambda D_V u$

4) 0 is an eigenvalue of L_{rw} with the constant one vector 1 as eigenvector. 0 is an eigenvalue of L_{sym} with eigenvector $D_v^{\frac{1}{2}} 1$

5) L_{sym} is symmetric and positive semi-definite and L_{sym} and L_{rw} have $|V|$ non-negative real-valued eigenvalues $0 \leq \lambda_1 \leq \cdots \leq \lambda_{|V|}$

Proof:

1) The complete proof of (1) can be found in [9].

2) (2) can be seen easily by solving

$$L_{sym} w = \lambda w \Leftrightarrow (I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}) w = \lambda w$$
$$\Leftrightarrow D_v^{-\frac{1}{2}} (I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}) w = \lambda D_v^{-\frac{1}{2}} w$$
$$\Leftrightarrow D_v^{-\frac{1}{2}} w - D_v^{-1} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} w = \lambda D_v^{-\frac{1}{2}} w$$

Let $u = D_v^{-\frac{1}{2}} w$, (in the other words, $w = D_v^{\frac{1}{2}} u$), we have

$$L_{sym} w = \lambda w \Leftrightarrow u - D_v^{-1} H W D_e^{-1} H^T u = \lambda u$$
$$\Leftrightarrow (I - D_v^{-1} H W D_e^{-1} H^T) u = \lambda u$$
$$\Leftrightarrow L_{rw} u = \lambda u$$

This completes the proof.

3) (3) can be seen easily by solving

$$L_{rw}u = \lambda u \Leftrightarrow (I - D_v^{-1}HWD_e^{-1}H^T)u = \lambda u$$
$$\Leftrightarrow D_v(I - D_v^{-1}HWD_e^{-1}H^T)u = \lambda D_v u$$
$$\Leftrightarrow (D_v - HWD_e^{-1}H^T)u = \lambda D_v u$$
$$\Leftrightarrow Lu = \lambda D_v u$$

This completes the proof.

4) First, we need to prove that $L_{rw}1 = 0$.

Let $d_v \in R^{|v|}$ be the vector containing the degrees of vertices of hypergraph G, $d_e \in R^{|E|}$ be the vector containing the degrees of hyper-edges of hypergraph G, $w \in R^{|E|}$ be the vector containing the weights of hyper-edges of G, $1 \in R^{|V|}$ be vector of all ones, and $one \in R^{|E|}$ be the vector of all ones. Hence we have

$$L_{rw}1 = (I - D_v^{-1}HWD_e^{-1}H^T)1$$
$$= 1 - D_v^{-1}HWD_e^{-1}d_e$$
$$= 1 - D_v^{-1}HWone$$
$$= 1 - D_v^{-1}Hw$$
$$= 0$$

The second statement is a direct consequence of (2).

5) The statement about L_{sym} is a direct consequence of (1), then the statement about L_{rw} is a direct consequence of (2).

III. Algorithms

Given a set of proteins $\{x1,\cdots,xl,x(l+1),\cdots,x(l+u)\}$ where $n = l+u$ is the total number of proteins (i.e. vertices) in the hypergraph $G=(V,E)$ and given the incidence matrix H of G. The method constructing H from the gene expression data will be described clearly in the Experiments and Results section.

Define c be the total number of functional classes and the matrix $F \in R^{n\times c}$ be the estimated label matrix for the set of proteins $\{x1,\cdots,xl,x(l+1),\cdots,x(l+u)\}$, where the point x_i is labeled as sign (F_{ij}) for each functional class j $(1 \le j \le c)$. Please note that $\{x1,\cdots,xl\}$ is the set of all labeled points and $\{x(l+1),\cdots,x(l+u)\}$ is the set of all un-labeled points.

Let $Y \in R^{n\times c}$ the initial label matrix for n proteins in the hypergraph G be defined as follows

$$Y_{ij} = \begin{cases} 1 \text{ if } x_i \in \text{functional class } j \text{ and } 1 \le i \le l \\ -1 \text{ if } x_i \notin \text{functional class } j \text{ and } 1 \le i \le l \\ 0 \text{ if } l+1 \le i \le n \end{cases}$$

Our objective is to predict the labels of the un-labeled points $x_{l+1},\cdots,x_{l+u}$. Basically, all proteins in the same hyper-edge should have the same label.

Random walk hypergraph Laplacian based semi-supervised learning algorithm

In this section, we will give the brief overview of the random walk hypergraph Laplacian based semi-supervised learning algorithm. The outline of the new version of this algorithm is as follows

1) Construct D_v and D_e from the incidence matrix H of G
2) Construct $S_{rw} = D_v^{-1}HWD_e^{-1}H^T$
3) Iterate until convergence $F^{(t+1)} = \alpha S_{rw}F^{(t)} + (1-\alpha)Y$, where α is an arbitrary parameter belongs to [0,1]
4) Let F^* be the limit of the sequence $\{F^{(t)}\}$. For each protein functional class j, label each protein $x_i\,(l+1 \le i \le l+u)$ as sign (F_{ij}^*)

Next, we look for the closed-form solution of the random walk graph Laplacian based semi-supervised learning. In the other words, we need to show that

$$F^* = \lim_{t\to\infty} F^{(t)} = (1-\alpha)(I - \alpha S_{rw})^{-1}Y$$

Suppose $F^{(0)} = Y$. Thus, by induction,

$$F^{(t)} = \alpha^t S_{rw}^t Y + (1-\alpha)\sum_{i=0}^{t-1}(\alpha S_{rw})^i Y$$

Since S_{rw} is the stochastic matrix, its eigenvalues are in [-1,1]. Moreover, since 0<α<1, thus

$$\lim_{t\to\infty} \alpha^t S_{rw}^t = 0$$

$$\lim_{t\to\infty} \sum_{i=0}^{t-1} (\alpha S_{rw})^i = (I - \alpha S_{rw})^{-1}$$

Therefore,

$$F^* = \lim_{t\to\infty} F^{(t)} = (1-\alpha)(I - \alpha S_{rw})^{-1}Y$$

Now, from the above formula, we can compute F^* directly.

Symmetric normalized hypergraph Laplacian based semi-supervised learning algorithm

Next, we will give the brief overview of the symmetric normalized hypergraph Laplacian based semi-supervised learning algorithm which can be obtained from [9,10]. The outline of this algorithm is as follows

1) Construct D_v and D_e from the incidence matrix H of G
2) Construct $S_{sym} = D_v^{-\frac{1}{2}}HWD_e^{-1}H^T D_v^{-\frac{1}{2}}$
3) Iterate until convergence $F^{(t+1)} = \alpha S_{sym}F^{(t)} + (1-\alpha)Y$, where α is an arbitrary parameter belongs to [0,1]
4) Let F^* be the limit of the sequence $\{F^{(t)}\}$. For each protein functional class j, label each protein $x_i\,(l+1 \le i \le l+u)$ as sign (F_{ij}^*)

Next, we look for the closed-form solution of the normalized graph Laplacian based semi-supervised learning. In the other words, we need to show that

$$F^* = \lim_{t\to\infty} F^{(t)} = (1-\alpha)(I - \alpha S_{sym})^{-1}Y$$

Suppose $F^{(0)} = Y$. Thus, by induction

$$F^{(t)} = \alpha^t S_{sym}^t Y + (1-\alpha)\sum_{i=0}^{t-1}\left(\alpha S_{sym}\right)^i Y$$

Since S_{sym} is similar to S_{rw} ($S_{rw} = D_v^{-1}HWD_e^{-1}H^T = D_v^{-\frac{1}{2}} S_{sym} D_v^{\frac{1}{2}}$) which is a stochastic matrix, eigenvalues of S_{sym} belong to [-1,1]. Moreover, since 0<α<1, thus

$$\lim_{t\to\infty} \alpha^t S_{sym}^t = 0$$

$$\lim_{t\to\infty} \sum_{i=0}^{t-1}\left(\alpha S_{sym}\right)^i = (I - \alpha S_{sym})^{-1}$$

Therefore,

$$F^* = \lim_{t\to\infty} F^{(t)} = (1-\alpha)(I - \alpha S_{sym})^{-1} Y$$

Now, from the above formula, we can compute F^* directly.

Un-normalized hypergraph Laplacian based semi-supervised learning algorithm

Finally, we will give the brief overview of the un-normalized hypergraph Laplacian based semi-supervised learning algorithm. The outline of this algorithm is as follows

1) Construct D_v and D_e from the incidence matrix H of G
2) Construct $L = D_v - HWD_e^{-1}H^T$
3) Compute closed form solution $F^* = \gamma(L + \gamma I)^{-1} Y$, where γ is any positive parameter
4) For each protein functional class j, label each protein $x_i \left(l+1 \le i \le l+u\right)$ as $\text{sign}\left(F_{ij}^*\right)$

The closed form solution F^* of un-normalized hypergraph Laplacian based semi-supervised learning algorithm will be derived clearly and completely in Regularization Framework section.

IV. Regularization Frameworks

In this section, we will develop the regularization framework for the symmetric normalized hypergraph Laplacian based semi-supervised learning iterative version. First, let's consider the error function

$$E(F) = \frac{1}{2}\left\{\sum_{e\in E}\sum_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)} \left\| \frac{F_u}{\sqrt{d(u)}} - \frac{F_v}{\sqrt{d(v)}} \right\|^2 \right\} + \gamma \sum_{i=1}^{|V|} \left\| F_i - Y_i \right\|^2$$

In this error function $E(F)$, F_i and Y_i belong to R^c. Please note that c is the total number of protein functional classes and γ is the positive regularization parameters. Hence

$$F = \begin{bmatrix} F_1^T \\ \vdots \\ F_{|V|}^T \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_1^T \\ \vdots \\ Y_{|V|}^T \end{bmatrix}$$

Here $E(F)$ stands for the sum of the square loss between the estimated label matrix and the initial label matrix and the sum of the changes of a function F over the hyper-edges of the hypergraph [9].

Hence we can rewrite $E(F)$ as follows

$$E(F) = trace\left(F^T L_{sym} F\right) + \gamma trace\left(\left(F-Y\right)^T \left(F-Y\right)\right)$$

Our objective is to minimize this error function. In the other words, we solve

$$\frac{\partial E}{\partial F} = 0$$

This will lead to

$$\left(I - D_v^{-\frac{1}{2}} HWD_e^{-1}H^T D_v^{-\frac{1}{2}}\right)F + \gamma\left(F-Y\right) = 0$$

$$F - D_v^{-\frac{1}{2}} HWD_e^{-1}H^T D_v^{-\frac{1}{2}} F + \gamma F = \gamma Y$$

$$F - \frac{1}{1+\gamma} D_v^{-\frac{1}{2}} HWD_e^{-1}H^T D_v^{-\frac{1}{2}} F = \frac{\gamma}{1+\gamma} Y$$

$$\left(I - \frac{1}{1+\gamma} D_v^{-\frac{1}{2}} HWD_e^{-1}H^T D_v^{-\frac{1}{2}}\right)F = \frac{\gamma}{1+\gamma} Y$$

Let $\alpha = \frac{1}{1+\gamma}$. Hence the solution F^* of the above equations is

$$F^* = (1-\alpha)(I - \alpha D_v^{-\frac{1}{2}} HWD_e^{-1}H^T D_v^{-\frac{1}{2}})^{-1} Y$$

Please note that $S_{rw} = D_v^{-1}HWD_e^{-1}H^T$ is not the symmetric matrix, thus we cannot develop the regularization framework for the random walk hypergraph Laplacian based semi-supervised learning iterative version.

Next, we will develop the regularization framework for the un-normalized hypergraph Laplacian based semi-supervised learning algorithms. First, let's consider the error function

$$E(F) = \frac{1}{2}\left\{\sum_{e\in E}\sum_{\{u,v\}\subseteq E} \frac{w(e)}{d(e)} \left\| F_u - F_v \right\|^2 \right\} + \gamma \sum_{i=1}^{|V|} \left\| F_i - Y_i \right\|^2$$

In this error function $E(F)$, F_i and Y_i belong to R^c. Please note that c is the total number of protein functional classes and γ is the positive regularization parameters. Hence

$$F = \begin{bmatrix} F_1^T \\ \vdots \\ F_{|V|}^T \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_1^T \\ \vdots \\ Y_{|V|}^T \end{bmatrix}$$

Here $E(F)$ stands for the sum of the square loss between the estimated label matrix and the initial label matrix and the sum of the changes of a function F over the hyper-edges of the hypergraph [9].

Hence we can rewrite $E(F)$ as follows

$$E(F) = F^T LF + \gamma trace\left(\left(F-Y\right)^T \left(F-Y\right)\right)$$

Please note that un-normalized hypergraph Laplacian matrix is $L = D_v - HWD_e^{-1}H^T$. Our objective is to minimize this error function. In the other words, we solve

$$\frac{\partial E}{\partial F} = 0$$

This will lead to

$$LF + \gamma(F - Y) = 0$$
$$(L + \gamma I)F = \gamma Y$$

Hence the solution F^* of the above equations is

$$F^* = \gamma(L + \gamma I)^{-1}Y$$

Similarly, we can also obtain the other form of solution F^* of the normalized graph Laplacian based semi-supervised learning algorithm as follows (note the symmetric normalized hypergraph Laplacian matrix is

$$L_{sym} = I - D_v^{-\frac{1}{2}}HWD_e^{-1}H^TD_v^{-\frac{1}{2}}$$

$$F^* = \gamma(L_{sym} + \gamma I)^{-1}Y$$

V. Experiments and Results

A. Datasets

In this paper, we use the dataset available from [11] and the references therein. This dataset contains the gene expression data measuring the expression of 4062 S. cerevisiae genes under the set of 215 titration experiments. These proteins are annotated with 138 GO Biological Process functions. In the other words, we are given gene expression data ($R^{4062\times215}$) matrix and the annotation (i.e. the label) matrix ($R^{4062\times138}$). Every expression value is normalized to z-transformed score such that every gene expression profile has the mean 0 and the standard deviation 1.

B. Experiments

In this section, we experiment with the above proposed three methods and the current state of the art network-based method (i.e. the un-normalized graph Laplacian based semi-supervised learning method) in terms of classification accuracy performance measure. All experiments were implemented in Matlab 6.5 on virtual machine.

Given the gene expression data, we can define the co-expression similarity S_{ij} of gene *i* and gene *j* as the absolute value of the Pearson's correlation coefficient between their gene expression profiles. We have $S(i,j) = \left|corr\left(g(i,:), g(j,:)\right)\right|$, where *g(i,:)* and *g(j,:)* are gene expression profiles of gene *i* and gene *j* respectively. We can define the adjacency matrix *A* ($R^{4062\times4062}$) as follows

$$A(i,j) = \begin{cases} 1 \; if \; s(i,j) > threshold \\ 0 \; if \; s(i,j) \le threshold \end{cases}$$

In this paper, without bias, we can set threshold be 0.5. Then the un-normalized graph Laplacian based semi-supervised learning method can be applied to this adjacency matrix A. The un-normalized graph Laplacian based semi-supervised learning method (i.e. the current state of the art method in network-based methods for protein function prediction) will be served as the baseline method in this paper. Its average accuracy performance measure for 138 GO Biology Process functions will be compared with the average accuracy performance measures of thee hypergraph Laplacian based semi-supervised learning methods. Please note that three-fold cross validation is used to compute the average accuracy performance measures of all four methods used in this paper. The accuracy performance measure *Q* is given as follows

$$Q = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

Normally, clustering methods offer a natural way to the problem identifying groups of genes that show very similar patterns of expression and tend to have similar functions [8] (i.e. the possible functional modules) in the gene expression data. In this experiment, we use k-mean clustering method (i.e. the most popular "hard" clustering method) since there exists at least one protein that has one GO Biological Process function only. Without bias, if all genes in the gene expression data have at least two GO Biological Process functions, we will use "soft" k-mean clustering method or fuzzy c-means clustering method. Then each cluster can be considered as the hyper-edge of the hypergraph. By using these hyper-edges, we can construct the incidence matrix H of the hypergraph. To make things simple, we can determine the number of cluster of the k-means method as follows

$$number\ of\ cluster = \sqrt{\frac{number\ of\ proteins}{2}}$$

When H is already computed, the random walk, symmetric normalized, and un-normalized hypergraph Laplacian based semi-supervised learning can be implemented. Finally, their average accuracy performance measures for all 138 GO Biological Process functions will be computed. These average accuracy performance measures of the three hypergraph Laplacian based methods are given in the following Table I. In these experiments, the parameter α is set to 0.85 and $\gamma = 1$.

From the above table, we recognized that the average accuracy performance measures for 138 GO Biological Process function of three hypergraph Laplacian based semi-supervised learning are equal. This will be investigated in the future and in the other biological datasets such as protein-protein interaction networks.

Interestingly, the average accuracy performance measures for 138 GO Biological Process of three hypergraph Laplacian based semi-supervised learning methods are much greater than the average accuracy performance measures of graph Laplacian based semi-supervised learning method.

TABLE I.

	Average Accuracy Performance Measure			
138 GO Biological Process functions	Graph (un-normalized)	Hypergraph (un-normalized)	Hypergraph (random walk)	Hypergraph (normalized)
	63.99	97.95	97.95	97.95

VI. Conclusions

We have proposed the detailed algorithms and regularization frameworks of the three un-normalized, symmetric normalized, and random walk hypergraph Laplacian based semi-supervised learning methods applying to protein function prediction problem. Experiments show that these three methods greatly perform better than the un-normalized graph Laplacian based semi-supervised learning method since these three methods utilize the complex relationships among proteins (i.e. not pairwise relationship). Moreover, these three methods can not only be used in the classification problem but also the ranking problem. In specific, given a set of genes (i.e. the queries) involved in a specific disease such as leukemia which is my future research, these three methods can be used to find more genes involved in leukemia by ranking genes in the hypergraph constructed from gene expression data. The genes with the highest rank can then be selected and checked by biology experts to see if the extended genes are in fact involved in leukemia. Finally, these selected genes will be used in cancer classification.

Recently, to the best of my knowledge, the un-normalized graph p-Laplacian based semi-supervised learning method have not yet been developed and applied to protein function prediction problem. This method is worth investigated because of its difficult nature and its close connection to partial differential equation on graph field.

References

[1] H. H. Shin, A. M. Lisewski, and O. Lichtarge, "Graph sharpening plus graph integration: A synergy that improves protein functional classification," *Bioinformatics*, vol. 23, no. 23, pp. 3217-3224, 2007.

[2] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," in *Proc. National Academy of Sciences of the United States of America*, vol. 85, no. 8, 1998, pp. 2444-2448.

[3] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, *et al*., "Expression monitoring by hybridization to high-density oligonucleotide arrays," *Nature Biotechnology*, vol. 14, no. 13, pp. 1675-1680, 1996.

[4] L. Shi, Y. Cho, and A. Zhang, "Prediction of protein function from connectivity of protein interaction networks," *International Journal of Computational Bioscience*, vol. 1, no. 1, pp. 210-1009, 2010.

[5] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble, "Kernel-based data fusion and its application to protein function prediction in yeast," *Pacific Symposium on Biocomputing*, 2004.

[6] K. Tsuda, H. H. Shin, and B. Schoelkopf, "Fast protein classification with multiple networks," *Bioinformatics*, vol. 21, issue suppl. 2, pp. ii59-ii65, 2005.

[7] L. Tran, "Application of three graph laplacian based semi-supervised learning methods to protein function prediction problem," CoRR abs/1211.4289, 2012.

[8] G. Pandey, G. Atluri, M. Steinbach, and V. Kumar, "Association analysis techniques for discovering functional modules from microarray data," in *Proc. ISMB Special Interest Group Meeting on Automated Function Prediction*, 2008.

[9] D. Zhou, J. Huang, and B. Schölkopf, "Beyond pairwise classification and clustering using hypergraphs," Max Planck Institute Technical Report 143, Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2005.

[10] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Advances in Neural Information Processing System 19*, B. Schölkopf, J. C. Platt, and T. Hofmann, Eds., Cambridge, MA: MIT Press, 2007, pp. 1601-1608.

[11] G. Pandey, L. C. Myers, and V. Kumar, "Incorporating functional inter-relationships into protein function prediction algorithms," *BMC Bioinformatics*, vol. 10, no. 1, 2009.

Loc Tran completed Bachelor of Science and Master of Science in Computer Science at University of Minnesota in 2003 and 2012 respectively. Currently, he's a PhD student at University of Technology, Sydney.

Linh Tran completed Bachelor of Science and Master of Science in Electrical and Computer Engineer at Portland State University. Currently, he's a PhD student at Portland State University.

