Un-normlized and Random Walk Hypergraph Laplacian Un-supervised Learning

Loc Hoang Tran[1(✉)], Linh Hoang Tran[2], and Hoang Trang[3]

[1] Computer Science Department/University of Minnesota, Minneapolis, USA
tran0398@umn.edu
[2] ECE Department/Portland State University, Portland, USA
linht@pdx.edu
[3] Ho Chi Minh City University of Technology-VNU HCM
Ho Chi Minh City, Vietnam
hoangtrang@hcmut.edu.vn

Abstract. Most network-based clustering methods are based on the assumption that the labels of two adjacent vertices in the network are likely to be the same. However, assuming the pairwise relationship between vertices is not complete. The information a group of vertices that show very similar patterns and tend to have similar labels is missed. The natural way overcoming the information loss of the above assumption is to represent the given data as the hypergraph. Thus, in this paper, the two un-normalized and random walk hypergraph Laplacian based un-supervised learning methods are introduced. Experiment results show that the accuracy performance measures of these two hypergraph Laplacian based un-supervised learning methods are greater than the accuracy performance measure of symmetric normalized graph Laplacian based un-supervised learning method (i.e. the baseline method of this paper) applied to simple graph created from the incident matrix of hypergraph.

Keywords: Hypergraph Laplacian · Clustering · Un-supervised learning

1 Introduction

In data mining problem sceneries, we usually assume the pairwise relationship among the objects to be investigated such as documents [1,2], or genes [3], or digits [1,2]. For example, if we group a set of points in Euclidean space and the pairwise relationships are symmetric, an un-directed graph may be employed. In this un-directed graph, a set of vertices represent objects and edges link the pairs of related objects. However, if the pairwise relationships are asymmetric, the object set will be modeled as the directed graph. Finally, a number of data mining methods for un-supervised learning [4] (i.e. clustering) and semi-supervised learning [5,6,7] (i.e. classification) can then be formulated in terms of operations on this graph.

However, in many real world applications, representing the set of objects as un-directed graph or directed graph is not complete. Approximating complex relationship as pairwise will lead to the loss of information. Let us consider classifying a set of

P.C. Vinh et al. (Eds.): ICTCC 2014, LNICST 144, pp. 254–263, 2015.
DOI: 10.1007/978-3-319-15392-6_25

genes into different gene functions. From [3], we may construct an un-directed graph in which the vertices represent the genes and two genes are connected by an edge if these two genes show a similar pattern of expression (i.e. the gene expression data is used as the datasets in [3]). Any two genes connected by an edge tend to have similar functions. However, assuming the pairwise relationship between 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 is to represent the gene expression data as the hypergraph [1,2]. A hypergraph is a graph in which an edge (i.e. a hyper-edge) can connect more than two vertices. However, the clustering methods for this hypergraph datasets have not been studied in depth. Moreover, the number of hyper-edges may be large. Hence this leads to the development of the clustering method that combine the dimensional reduction methods for the hypergraph dataset and the popular hard k-mean clustering method. Utilizing this idea, in [1,2], the symmetric normalized hypergraph Laplacian based un-supervised learning method have been developed and successfully applied to zoo dataset. To the best of our knowledge, the random walk and un-normalized hypergraph Laplacian based un-supervised learning methods have not yet been developed and applied to any practical applications. In this paper, we will develop the random walk and un-normalized hypergraph Laplacian based un-supervised learning methods and apply these two methods to the zoo dataset available from UCI repository.

We will organize the paper as follows: Section II will introduce the definition of hypergraph Laplacians and their properties. Section III will introduce the un-normalized, random walk, and symmetric normalized hypergraph Laplacian based un-supervised learning algorithms in detail. In section IV, we will apply the symmetric normalized graph Laplacian based un-supervised learning algorithm (i.e. the current state of art network based clustering method) to zoo dataset available from UCI repository and compare its accuracy performance measure to the two proposed hypergraph Laplacian based un-supervised learning algorithms' accuracy performance measures. Section V will conclude this paper and the future directions of research of these methods will be discussed.

2 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 $R^{|E|*|E|}$ diagonal matrix containing the weights of all hyper-edges in its diagonal entries.

2.1 Definition of Incidence Matrix H of G

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

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

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

$$d(v) = \sum_{e \in E} w(e) * h(v, e)$$

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

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 $R^{|v| * |v|}$ matrix and D_e is the $R^{|e| * |e|}$ matrix.

2.2 Definition of the Un-normalized Hypergraph Laplacian

The un-normalized hypergraph Laplacian is defined as follows

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

2.3 Properties of L

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

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

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 eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|V|}$

Proof:

1. We know that

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

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

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

$$= \sum_{u \in V} f(u)^2 \sum_{e \in E} w(e)h(u,e) - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} f(u)f(v)h(u,e)h(v,e)$$
$$= \sum_{u \in V} f(u)^2 d(u) - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} f(u)f(v)h(u,e)h(v,e)$$
$$= f^T D_v f - f^T HWD_e^{-1}H^T f$$
$$= f^T (D_v - HWD_e^{-1}H^T) f$$
$$= f^T L f$$

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

   Since for every vector $f \in R^{|V|}$, $f^T L f = \frac{1}{2}\sum_{e \in E} \sum_{\{u,v\} \subseteq E} \frac{w(e)}{d(e)} (f(u) - f(v))^2 \geq 0$. 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

   $$L1 = (D_v - HWD_e^{-1}H^T)1 = d_v - HWD_e^{-1}d_e = d_v - HWone = d_v - Hw = d_v - d_v = 0$$

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

2.4 The Definitions of Symmetric Normalized and Random Walk Hypergraph Laplacians

The symmetric normalized hypergraph Laplacian (defined in [1,2]) is defined as follows

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

The random walk hypergraph Laplacian (defined in [1,2]) is defined as follows

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

2.5 Properties of L_{sym} and L_{rw}

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

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

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 $Lu = \lambda D_v u$

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 [1].
2. (2) can be seen easily by solving

$$\begin{aligned} L_{sym}w = \lambda w &\Leftrightarrow \left(I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}\right) w = \lambda w \\ &\Leftrightarrow D_v^{-\frac{1}{2}} \left(I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}\right) w = \lambda D_v^{-\frac{1}{2}} w \\ &\Leftrightarrow D_v^{-\frac{1}{2}} w - D_v^{-1} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} w = \lambda D_v^{-\frac{1}{2}} w \end{aligned}$$

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

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

This completes the proof.

3. (3) can be seen easily by solving

$$\begin{aligned} L_{rw}u = \lambda u &\Leftrightarrow (I - D_v^{-1} H W D_e^{-1} H^T) u = \lambda u \\ &\Leftrightarrow D_v (I - D_v^{-1} H W D_e^{-1} H^T) u = \lambda D_v u \\ &\Leftrightarrow (D_v - H W D_e^{-1} H^T) u = \lambda D_v u \\ &\Leftrightarrow L u = \lambda D_v u \end{aligned}$$

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

$$\begin{aligned} L_{rw} 1 &= (I - D_v^{-1} H W D_e^{-1} H^T) 1 \\ &= 1 - D_v^{-1} H W D_e^{-1} d_e \\ &= 1 - D_v^{-1} H W one \\ &= 1 - D_v^{-1} H w \\ &= 1 - D_v^{-1} d_v \\ &= 0 \end{aligned}$$

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).

3 Algorithms

Given a set of points $\{x_1, x_2, \dots, x_n\}$ where n is the total number of points (i.e. vertices) in the hypergraph *G=(V,E)* and given the incidence matrix *H* of *G*.

Our objective is to partition these *n* points into *k* groups.

Random walk hypergraph Laplacian based un-supervised learning algorithm

First, we will give the brief overview of the random walk hypergraph Laplacian based un-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. Compute the random walk hypergraph Laplacian $L_{rw} = I - D_v^{-1} H W D_e^{-1} H^T$
3. Compute all eigenvalues and eigenvectors of L_{rw} and sort all eigenvalues and their corresponding eigenvector in ascending order. Pick the first k eigenvectors $v_2, v_3, \dots, v_{k+1}$ of L_{rw} in the sorted list. k can be determined in the following two ways:
    a. k is the number of connected components of L_{rw} [4]
    b. k is the number such that $\frac{\lambda_{k+2}}{\lambda_{k+1}}$ or $\lambda_{k+2} - \lambda_{k+1}$ is largest for all $2 \leq k \leq n$
4. Let $V \in R^{n*k}$ be the matrix containing the vectors $v_2, v_3, \dots, v_{k+1}$ as columns.
5. For $i = 1, .., n$, let $y_i \in R^{1*k}$ be the vector corresponding to the *i-th* row of V.
6. Cluster the points y_i for all $1 \leq i \leq n$ with k-means clustering method.

Un-normalized hypergraph Laplacian based un-supervised learning algorithm

Next, we will give the brief overview of the un-normalized hypergraph Laplacian based un-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. Compute the un-normalized hypergraph Laplacian $L = D_v - H W D_e^{-1} H^T$
3. Compute all eigenvalues and eigenvectors of L and sort all eigenvalues and their corresponding eigenvector in ascending order. Pick the first k eigenvectors $v_2, v_3, \dots, v_{k+1}$ of L in the sorted list. k can be determined in the following two ways:
    a. k is the number of connected components of L [4]
    b. k is the number such that $\frac{\lambda_{k+2}}{\lambda_{k+1}}$ or $\lambda_{k+2} - \lambda_{k+1}$ is largest for all $2 \leq k \leq n$
4. Let $V \in R^{n*k}$ be the matrix containing the vectors $v_2, v_3, \dots, v_{k+1}$ as columns
5. For $i = 1, .., n$, let $y_i \in R^{1*k}$ be the vector corresponding to the *i-th* row of *V*
6. Cluster the points y_i for all $1 \leq i \leq n$ with k-means clustering method

Symmetric normalized hypergraph Laplacian based un-supervised learning algorithm

Next, we will give the brief overview of the symmetric normalized hypergraph Laplacian based un-supervised learning algorithm which can be obtained from [1,2]. The outline of this algorithm is as follows

1. Construct D_v and D_e from the incidence matrix H of G
2. Compute the symmetric normalized hypergraph Laplacian $L_{sym} = I - D_v^{-\frac{1}{2}} HWD_e^{-1}H^T D_v^{-\frac{1}{2}}$
3. Compute all eigenvalues and eigenvectors of L_{sym} and sort all eigenvalues and their corresponding eigenvector in ascending order. Pick the first k eigenvectors $v_2, v_3, \dots, v_{k+1}$ of L_{sym} in the sorted list. k can be determined in the following two ways:
   a. k is the number of connected components of L_{sym} [4]
   b. k is the number such that $\frac{\lambda_{k+2}}{\lambda_{k+1}}$ or $\lambda_{k+2} - \lambda_{k+1}$ is largest for all $2 \le k \le n$
4. Let $V \in R^{n*k}$ be the matrix containing the vectors $v_2, v_3, \dots, v_{k+1}$ as columns
5. For $i = 1, .., n$, let $y_i \in R^{1*k}$ be the vector corresponding to the *i-th* row of V
6. Cluster the points y_i for all $1 \le i \le n$ with k-means clustering method

At step 6 of the above three algorithms, k-means clustering method is used for simplicity and is not discussed. Next, the k-mean clustering methods will be discussed. The k-mean clustering method is considered the most popular method in clustering field [4]. The k-mean clustering method can be completed in the following four steps:

1. Randomly choose *k* initial cluster centers (i.e. centroids).
2. For every feature vector, associate it with the closest centroid.
3. Recalculate the centroid for all *k* clusters.
4. Repeat step 2 and step 3 until convergence.

In the other words, the k-mean clustering method is trying to minimize the objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} r_{ij} ||F(i,:) - c_j||^2$$

In the above formula, c_j is the centroid of the cluter *j*. $F(i,:)$ is the *i-th* feature vector. The matrix R is defined as follows

$$r_{ij} = \begin{cases} 1 \; if \; feature \; vector \; i \; belongs \; to \; cluster \; j \\ \quad 0 \; otherwise \end{cases}$$

Moreover, we can also easily see that

$$J = trace(\sum_{j=1}^{k}\sum_{i \in j}\left(F(i,:) - c_j\right)^T (F(i,:) - c_j))$$

Finally, the current state of the art network based clustering method (i.e. the symmetric normalized graph Laplacian based un-supervised learning method) can be completed in the following steps.

1. Compute the symmetric graph Laplacian L_{g-sym}: $L_{g-sym} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$.
2. Compute all eigenvalues and eigenvectors of L_{g-sym} and sort all eigenvalues and their corresponding eigenvector in ascending order. Pick the first k eigenvectors $v_2, v_3, \dots, v_{k+1}$ of L_{g-sym} in the sorted list. k can be determined in the following two ways:
   a. k is the number of connected components of L_{g-sym} [4]
   b. k is the number such that $\frac{\lambda_{k+2}}{\lambda_{k+1}}$ or $\lambda_{k+2} - \lambda_{k+1}$ is largest for all $2 \le k \le n$
3. Let $V \in R^{n*k}$ be the matrix containing the vectors $v_2, v_3, \dots, v_{k+1}$ as columns.
4. Compute the new matrix $U \in R^{n*k}$ from V as follows
$$u_{ij} = \frac{v_{ij}}{\sqrt{\sum_l v_{il}^2}}$$
5. For $i = 1, .., n$, let $y_i \in R^{1*k}$ be the vector corresponding to the *i-th* row of U.
6. Cluster the points y_i for all $1 \le i \le n$ with k-means clustering method.

The way describing how to construct *W* and *D* will be discussed in the next section.

4 Experiments and Results

Datasets

In this paper, we used the zoo data set which can be obtained from UCI repository. The zoo data set contains 100 animals with 17 attributes. The attributes include hair, feathers, eggs, milk, etc. The animals have been classified into 7 different classes. Our task is to embed the animals in the zoo dataset into Euclidean space by using random walk and un-normalized hypergraph Laplacian Eigenmaps and by using the symmetric normalized graph Laplacian Eigenmaps. We embed those animals into Euclidean space by using the eigenvectors of the graph Laplacian and hypergraph Laplacians associated with the 7 (i.e. number of classes) smallest eigenvalues different from 0. Finally, the k-mean clustering method is applied to the transformed dataset.

There are three ways to construct the similarity graph from the incident matrix H of zoo dataset:

a. The ε-neighborhood graph: Connect all animals whose pairwise distances are smaller than ε.
b. k-nearest neighbor graph: Animal *i* is connected with animal *j* if animal *i* is among the k-nearest neighbor of animal *j* or animal *j* is among the k-nearest neighbor of animal *i*.
c. The fully connected graph: All animals are connected.

In this paper, the similarity function is the Gaussian similarity function

$$w_{ij} = s(H(i,:), H(j,:)) = \exp\left(-\frac{d\big(H(i,:), H(j,:)\big)}{t}\right)$$

In this paper, t is set to 10 and the 3-nearest neighbor graph is used to construct the similarity graph from the zoo dataset. This describes how we construct W of the simple graph. D is the diagonal matrix and its *i-th* element is defined as follows:

$$d_i = \sum_j w_{ij}$$

Experiments and Results

In this section, we experiment with the above proposed un-normalized and random walk hypergraph Laplacian based un-supervised learning methods (i.e. hypergraph spectral clustering) and the current state of the art method (i.e. the symmetric normalized graph Laplacian based un-supervised learning method) which is spectral clustering method in terms of accuracy performance measure. The accuracy performance measure Q is given as follows

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

All experiments were implemented in Matlab 6.5 on virtual machine. The accuracy performance measures of the above proposed methods and the current state of the art method is given in the following table 1

Table 1. Accuracies of the two proposed methods and the current state of the art method

Accuracy Performance Measures (%)		
Graph (symmetric normalized)	Hypergraph (random walk)	Hypergraph (un-normalized)
89.43	94.86	93.71

From the above table, we recognized that the accuracy of the random walk hypergraph Laplacian method is slightly better than the accuracy of the un-normalized hypergraph Laplacian method. Interestingly, the accuracies of the two proposed hypergraph Laplacian methods are significantly better than accuracy of the current state of the art method.

5 Conclusion

We have proposed the detailed algorithms the two un-normalized and random walk hypergraph Laplacian based un-supervised learning methods applying to the zoo dataset. Experiments show that these two methods greatly perform better than the un-normalized graph Laplacian based un-supervised learning method since these two methods utilize the complex relationships among points (i.e. not pairwise relationship). These two methods can also be applied to digit recognition and text classification. These experiments will be tested in the future. Moreover, these two methods can not only be used in the clustering 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 two 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 hypergraph p-Laplacian based un-supervised learning method has not yet been developed. This method is worth investigated because of its difficult nature and its close connection to partial differential equation on hypergraph field.

References

1. Zhou, D., Huang, J., Schölkopf, B.: Beyond Pairwise Classification and Clustering Using Hypergraphs Max Planck Institute Technical Report 143. Max Planck Institute for Biological Cybernetics, Tübingen, Germany (2005)
2. Zhou, D., Huang, J., Schölkopf, B.: Learning with Hypergraphs: Clustering, Classification, and Embedding. In: Schölkopf, B., Platt, J.C., Hofmann, T. (eds.) Advances in Neural Information Processing System (NIPS), vol. 19, pp. 1601–1608. MIT Press, Cambridge (2007)
3. Tran, L.: Application of three graph Laplacian based semi-supervised learning methods to protein function prediction problem. CoRR abs/1211.4289 (2012)
4. Luxburg, U.: A Tutorial on Spectral Clustering Statistics and Computing **17**(4), 395–416 (2007)
5. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002)
6. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with Local and Global Consistency. In: Thrun, S., Saul, L., Schölkopf, B. (eds.) Advances in Neural Information Processing Systems (NIPS), vol. 16, pp. 321–328. MIT Press, Cambridge (2004)
7. Tsuda, K., Shin, H.H., Schoelkopf, B.: Fast protein classification with multiple networks. Bioinformatics (ECCB 2005) **21**(Suppl. 2), ii59–ii65 (2005)
8. Tran, L.: Hypergraph and protein function prediction with gene expression data. CoRR abs/1212.0388 (2012)