

Combinatorial and Random Walk Hypergraph Laplacian Eigenmaps

Loc Hoang Tran, Linh Hoang Tran, Hoang Trang, and Le Trung Hieu

Abstract—Most network-based machine learning 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. However, representing the dataset as the hypergraph will not lead to the perfection. The number of hyper-edges may be large; hence this will lead to high time complexity of the clustering methods or the classification methods when we try to apply the clustering/classification methods to this hypergraph dataset. Thus, there exists a need to develop the dimensional reduction methods for the hypergraph datasets. In this paper, the two un-normalized and random walk hypergraph Laplacian Eigenmaps are introduced. Experiment results show that the accuracy performance measures of these two hypergraph Laplacian Eigenmaps combined with graph based semi-supervised learning method are greater than the accuracy performance measure of graph based semi-supervised learning method alone (i.e. the baseline method of this paper) applied to the original hypergraph datasets.

Index Terms—Hypergraph, Laplacian, Eigenmaps, semi-supervised learning, graph.

I. INTRODUCTION

In many of the artificial intelligence disciplines, the information retrieval topics or the data mining areas, we have to solve the problems where the sample datasets are in a very high dimension space. This data is usually embedded in a differential manifold of low dimension. One of the main tasks in machine learning field is to develop the accurate representations for this type of data. The dimensional reduction methods have a long history. The most popular dimensional reduction method is Principal Component Analysis (PCA) [1], [2]. Principal Component Analysis (PCA) is used to compress the information contained in the dataset. The main idea of this technique is to compute the most “important” components from the original features. In the other words, the objective of PCA is to transform the original features into others, named principal components, and apply linear combinations of those features. Hence PCA tries to reduce the dimensionality and preserve as much as

possible the original randomness (i.e. the variance) in the high dimensional space. The problem of this method is that it does not consider the explicit form of the differential manifold structure in which the data probably lie.

In the last two decades, a lot of researchers have studied advanced dimensional reduction methods in which the local distances are conserved as much as possible. These advanced dimensional reduction methods such as Locally Linear Embedding (LLE) [3] or Laplacian Eigenmaps (LE) [4], [5] try to obtain and use the neighborhood information of each data point by considering the whole dataset as a graph.

However, in many real world applications, representing the dataset as un-directed graph is not complete. Approximating complex relationship as pairwise will lead to the loss of information. Let us consider classifying a set of genes into different gene functions. From [6], 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 [6]). 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 [7] (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 [7]-[9]. A hypergraph is a graph in which an edge (i.e. a hyper-edge) can connect more than two vertices. However, representing the dataset as the hypergraph will not lead to the perfection. The number of hyper-edges may be large; hence this will lead to high time complexity of the clustering methods or the classification methods when we try to apply the clustering/classification methods to this hypergraph dataset. Thus, there exists a need to develop the dimensional reduction methods for the hypergraph datasets. In [8], [9], the symmetric normalized hypergraph Laplacian Eigenmap has been developed and successfully applied to zoo dataset. To the best of my knowledge, the random walk and un-normalized hypergraph Laplacian Eigenmaps 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 Eigenmaps and apply these two methods combined with graph based semi-supervised learning method to the zoo dataset available from UCI repository and the modified 20-newsgroup dataset.

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 Eigenmaps algorithms in detail. In Section IV, we

Manuscript received December 31, 2014; revised March 25, 2015.

Loc Tran is with University of Technology, Sydney (e-mail: tran0398@umn.edu).

Linh Tran is with Portland State University, Portland (e-mail: linht@pdx.edu).

Hoang Trang is with Ho Chi Minh City University of Technology, Vietnam (e-mail: hoangtrang@hcmut.edu.vn).

Hieu Le is with the IC Design Lab, Ho Chi Minh City University of Technology, Vietnam (e-mail: lehieu.ee@gmail.com).

will apply the symmetric normalized graph Laplacian based semi-supervised learning algorithm (i.e. the current state of art network based semi-supervised learning method) to the original zoo dataset available from UCI repository and the modified 20-newsgroup dataset and compare its accuracy performance measure to the accuracy performance measures of the two proposed hypergraph Laplacian Eigenmaps algorithms combined with the symmetric normalized graph Laplacian based semi-supervised learning algorithm. Section V will conclude this paper and the future directions of research of these methods 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

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

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

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 $R^{|V| \times |V|}$ 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

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

- 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 $0 \leq \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{|V|}$

Proof:

We know that

$$\begin{aligned} & \frac{1}{2} \sum_{e \in E} \sum_{\{u,v\} \subseteq E} \frac{w(e)}{d(e)} (f(u) - f(v))^2 \\ &= \frac{1}{2} \sum_{e \in E} \sum_{\{u,v\} \subseteq E} \frac{w(e)}{d(e)} (f(u)^2 + f(v)^2 - 2f(u)f(v)) \\ &= \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} (f(u)^2 - f(u)f(v)) h(u, e)h(v, e) \\ &= \sum_{e \in E} \sum_{u \in V} w(e)f(u)^2 h(u, e) \sum_{v \in V} \frac{h(v, e)}{d(e)} \\ & \quad - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} f(u)f(v)h(u, e)h(v, e) \\ &= \sum_{e \in E} \sum_{u \in V} w(e)f(u)^2 h(u, e) \\ & \quad - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} f(u)f(v)h(u, e)h(v, e) \\ &= \sum_{u \in V} f(u)^2 \sum_{e \in E} w(e)h(u, e) \\ & \quad - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} f(u)f(v)h(u, e)h(v, e) \\ &= \sum_{u \in V} f(u)^2 d(u) - \sum_{e \in E} \sum_{u,v \in V} \frac{w(e)}{d(e)} f(u)f(v)h(u, e)h(v, e) \\ & \quad = f^T D_v f - f^T H W D_e^{-1} H^T f \\ & \quad = f^T (D_v - H W D_e^{-1} H^T) f \\ & \quad = f^T L f \end{aligned}$$

L is symmetric follows directly from its own definition.

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

1) 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

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

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

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

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

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

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

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

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

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

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

- 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 $\mathbf{1}$ as eigenvector. 0 is an eigenvalue of L_{sym} with eigenvector $D_v^{-\frac{1}{2}} \mathbf{1}$
- 5) L_{sym} is symmetric and positive semi-definite and L_{sym} and L_{rw} have $|V|$ non-negative real-valued eigenvalues $0 \leq \lambda_1 \leq \dots \leq \lambda_{|V|}$

Proof:

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

Can be seen easily by solving

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

This completes the proof.

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

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 , $\mathbf{1} \in R^{|V|}$ be vector of all ones, and $one \in R^{|E|}$ be the vector of all ones. Hence we have

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

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

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 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 compute the eigenvectors of the three hypergraph Laplacians.

A. Random Walk Hypergraph Laplacian Eigenmap Algorithm

First, we will give the brief overview of the random walk hypergraph Laplacian Eigenmap 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:

k is the number of connected components of L_{rw} [4].

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 $V \in R^{n \times k}$ be the matrix containing the vectors v_2, v_3, \dots, v_{k+1} as columns and V is the final result

B. Un-Normalized Hypergraph Laplacian Eigenmap Algorithm

Next, we will give the brief overview of the un-normalized hypergraph Laplacian Eigenmap 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:

k is the number of connected components of L [4]

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 $V \in R^{n \times k}$ be the matrix containing the vectors v_2, v_3, \dots, v_{k+1} as columns and V is the final result

C. Symmetric Normalized Hypergraph Laplacian Eigenmap Algorithm

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

k is the number of connected components of L_{sym} [4]

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 $V \in R^{n \times k}$ be the matrix containing the vectors v_2, v_3, \dots, v_{k+1} as columns and V is the final result

IV. EXPERIMENTS AND RESULTS

A. Datasets

In this paper, we used the zoo data set which can be obtained from UCI repository and the modified 20-newsgroup dataset (see <https://cs.uwaterloo.ca/~ppoupart/teaching/cs486-fall12/assignments/assignments.html>). 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. In this dataset, each attribute is the hyper-edge. The modified 20-newsgroup dataset is the dataset with binary occurrence values for 3566 words across 1768 documents. It is classified into two different classes. In this dataset, each word is the hyper-edge. Our task is to embed the animals in the zoo dataset and documents in modified 20-newsgroup dataset into Euclidean space by using random walk and un-normalized hypergraph Laplacian Eigenmaps. We embed those animals and documents into Euclidean space by using the eigenvectors of the hypergraph Laplacians associated with the k smallest eigenvalues different from 0. The way to find the optimal k is described in section III. Finally, the graph based semi-supervised learning method is applied to the original and the transformed datasets. The information about the graph based semi-supervised learning method can be found from [10].

There are three ways to construct the similarity graph from the incident matrix H and the transformed matrix of zoo dataset (or modified 20-newsgroup dataset):

- 1) The ε -neighborhood graph: Connect all animals (or documents) whose pairwise distances are smaller than ε .
- 2) k -nearest neighbor graph: Animal (or document) i is connected with animal (or document) j if animal (or document) i is among the k -nearest neighbor of animal (or document) j or animal (or document) j is among the k -nearest neighbor of animal (or document) i .
- 3) The fully connected graph: All animals (or documents) are connected.

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

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

This describes how we construct W of the simple graph. D is the diagonal matrix and its i -th element is defined as follows:

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

B. Experiments and Results

In this section, we experiment with the above proposed un-normalized and random walk hypergraph Laplacian Eigenmaps combined with the graph based semi-supervised learning method and the graph based semi-supervised

learning method applied directly to the two original datasets in terms of accuracy performance measure. The accuracy performance measure Q is given as follows

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

All experiments were implemented in Matlab 6.5 on virtual machine. The accuracy performance measures of the above proposed methods and the graph based semi-supervised learning method alone are given in the following Table I and Table II.

TABLE I: ACCURACIES OF THE TWO PROPOSED METHODS COMBINED WITH GRAPH BASED SEMI-SUPERVISED LEARNING METHOD AND THE GRAPH BASED SEMI-SUPERVISED LEARNING METHOD ALONE FOR ZOO DATASET

Accuracies (%)		
Graph based semi-supervised learning method	Un-normalized hypergraph Laplacian Eigenmap + Graph based semi-supervised learning method	Random walk hypergraph Laplacian Eigenmap + Graph based semi-supervised learning method
84	88	88

TABLE II: ACCURACIES OF THE TWO PROPOSED METHODS COMBINED WITH GRAPH BASED SEMI-SUPERVISED LEARNING METHOD AND THE GRAPH BASED SEMI-SUPERVISED LEARNING METHOD ALONE FOR THE MODIFIED 20-NEWSGROUP DATASET

Accuracies (%)		
Graph based semi-supervised learning method	Un-normalized hypergraph Laplacian Eigenmap + Graph based semi-supervised learning method	Random walk hypergraph Laplacian Eigenmap + Graph based semi-supervised learning method
78.08	83.03	98.02

From the above two tables, we recognized that the accuracy of the random walk hypergraph Laplacian Eigenmap is slightly better than the accuracy of the un-normalized hypergraph Laplacian Eigenmap. Interestingly, the accuracies of the two proposed hypergraph Laplacian Eigenmaps combined with graph based semi-supervised learning method are significantly better than accuracy of the graph based semi-supervised learning method alone.

V. CONCLUSION

We have proposed the detailed algorithms the two un-normalized and random walk hypergraph Laplacian Eigenmaps applying to the zoo dataset and the modified 20-newsgroup dataset. Experiments show that these two methods combined with the graph based semi-supervised learning method greatly perform better than the graph based semi-supervised learning method alone.

Moreover, in the future, we can also develop the hypergraph based semi-supervised ranking methods utilizing these three hypergraph Laplacians. Then these methods can be applied to disease gene prioritization problem in bio-informatics field. 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 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.

To the best of my knowledge, the un-normalized hypergraph p-Laplacian Eigenmap 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] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," *Springer Series in Statistics*, Springer New York Inc., 2001
- [2] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Computer Vision and Pattern Recognition*, 1991.
- [3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, no. 290, pp. 2323–2326, 2000.
- [4] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, no. 15, pp. 1373–1396, 2002.
- [5] T. Ameet, S. Kumar, and H. Rowley, "Large-scale manifold learning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
- [6] K. Tsuda, H. H. Shin, and B. Schoelkopf, "Fast protein classification with multiple networks bioinformatics (ECCB'05)," *Bioinformatics*, vol. 21, no. 2, pp. 59-65, 2005.
- [7] L. H. Tran and L. H. Tran, "Hypergraph and protein function prediction with gene expression data," *Journal of Automation and Control Engineering*, vol. 3, no. 2, pp. 164-170, April 2015.
- [8] J. Huang, B. Schölkopf *et al.*, "Beyond pairwise classification and clustering using hypergraphs max planck institute technical report 143," Max Planck Institute for Biological Cybernetics, Tübingen, Germany, 2005.
- [9] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: clustering, classification, and embedding," *Advances in Neural Information Processing System*, vol. 19, pp. 1601-1608, 2007.
- [10] L. Tran, "Application of three graph Laplacian based semi-supervised learning methods to protein function prediction problem," *Learning (cs.LG); Computational Engineering*, p. 4289, 2012.

Loc Tran completed the bachelor of science and master of science degrees 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 the bachelor of science and master of science degrees in electrical and computer engineer at Portland State University. Currently, he's a PhD student at Portland State University.

Hoang Trang completed the bachelor of science and master of science degrees at Ho Chi Minh City University of Technology in 2002 and 2004 respectively. He completed the PhD degree in 2008. Currently, he's a lecturer at Ho Chi Minh City University of Technology.

Hieu Le completed the bachelor of technology in electronics and communications at Hochiminh University of Technology in 2011, completed the master of science in communication engineering at National Chiao Tung University in 2013. Currently, he's a researcher in IC Design Lab at Hochiminh University of Technology.