Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-143

Beyond Pairwise Classification and Clustering Using Hypergraphs

Dengyong Zhou[1], Jiayuan Huang[1,2], and Bernhard Schölkopf[1]

August 18, 2005

[1] Department Schölkopf. E-mail: {dengyong.zhou, bernhard.schoelkopf}@tuebingen.mpg.de.
[2] School of Computer Science, University of Waterloo, Waterloo ON, N2L 3G1, Canada. E-mail: j9huang@cs.uwaterloo.ca.

This report is available in PDF–format via anonymous ftp at ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/hypergraph.pdf. The complete series of Technical Reports is documented at: http://www.kyb.tuebingen.mpg.de/techreports.html

Beyond Pairwise Classification and Clustering Using Hypergraphs

Dengyong Zhou [†], Jiayuan Huang [‡†] and Bernhard Schölkopf[†]
[†]Max Planck Institute for Biological Cybernetics, 72076 Tübingen, Germany
[‡†]School of Computer Science, University of Waterloo, Waterloo ON, N2L 3G1, Canada
{dengyong.zhou, jiayuan.huang, bernhard.schoelkopf}@tuebingen.mpg.de

August 18, 2005

Abstract

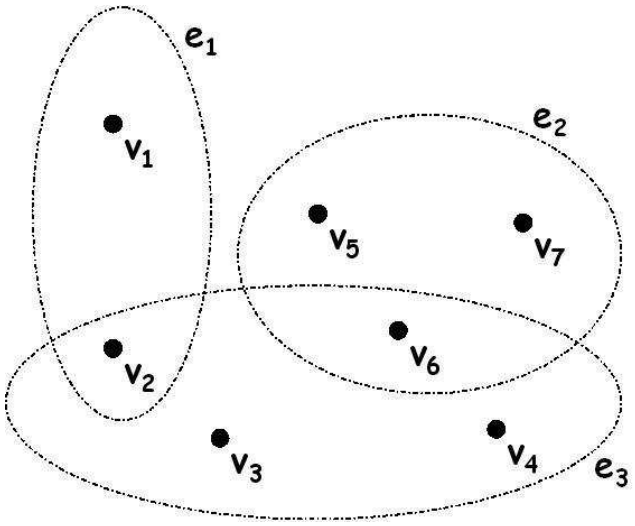
In many applications, relationships among objects of interest are more complex than pairwise. Simply approximating complex relationships as pairwise ones can lead to loss of information. An alternative for these applications is to analyze complex relationships among data directly, without the need to first represent the complex relationships into pairwise ones. A natural way to describe complex relationships is to use hypergraphs. A hypergraph is a graph in which edges can connect more than two vertices. Thus we consider learning from a hypergraph, and develop a general framework which is applicable to classification and clustering for complex relational data. We have applied our framework to real-world problems and obtained encouraging results.

1. Introduction

Typical machine learning approaches assume pairwise relationships among the objects to be investigated. For example, if we cluster a set of points in Euclidean space, then we may define an affinity matrix based on an RBF kernel on the points (Shi and Malik, 2000). If the pairwise relationships are symmetric, an undirected graph may be constructed, with a set of vertices representing the objects, and edges joining pairs of related objects. If the pairwise relationships are asymmetric, the object set may be modeled as a directed graph, such as the World Wide Web (Zhou et al., 2005).

However, for many applications, relationships among objects of interest are more complex than pairwise. Simply approximating complex relationships as pairwise ones can lead to loss of information. Let us consider classifying or clustering a collection of articles into different topics on the basis of the authors alone. We may construct an undirected graph in which the vertices represent the articles, and two articles are connected by an edge when there is at least one author in common. The edge may be weighted by the number of the authors in common. In this representation we obviously ignore the information regarding whether the same person is one of the authors of three or more papers or not.

An alternative for these applications is to analyze complex relationships among data directly, without the need to first represent complex relationships as pairwise ones. A natural solution to the complete representation of complex relationships is to use hypergraphs. A



	e_1	e_2	e_3
v_1	1	0	0
v_2	1	0	1
v_3	0	0	1
v_4	0	0	1
v_5	0	1	0
v_6	0	1	1
v_7	0	1	0

Figure 1: Left panel: a hypergraph with vertices $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and edges $E = \{e_1, e_2, e_3\}$; right panel: the incidence matrix H of the hypergraph.

hypergraph is a graph in which edges can connect more than two vertices. In other words, each edge is a subset of vertices. We illustrate a hypergraph in Fig. 1. In the above example, we may construct a hypergraph with the vertices representing the articles, and the edges the authors. Each edge contains the articles of the corresponding author. Clearly, an article might have several edges attached. We might add more relations among the articles, like the journals or proceedings where the articles are published. Then each journal or proceeding will be regarded as an edge as well. We can also associate different weights with the edges to measure their relative importance.

So, given a weighted hypergraph, in which some vertices are labeled, how to classify the remaining unlabeled vertices? Classifying a finite set of objects in which some are labeled is called *transductive inference* (Vapnik, 1998). It is natural to assign *similar* vertices (those having many incident hyperedges in common) to the same class. Thus one may simply assign a label to an unclassified vertex on the basis of the most common label among the classified neighbors of the vertex. However a preferable way is to exploit the structure of the hypergraph globally rather than locally, such that the classification is *consistent* over the whole hypergraph. The above ideas are obviously also applicable to clustering for hypergraphs, when there are no labeled vertices available.

The structure of this paper is as follows. We first introduce some basic notions on hypergraphs in Section 2. A general framework for learning on hypergraphs is presented in Section 3. In the absence of labeled instances, as shown in Section 4, this framework can be utilized as a spectral clustering approach for hypergraphs. In Section 5, we define a natural random walk over hypergraphs, such that the cut criterion and the regularization framework can be understood in terms of random walks. Experimental results on real-wold problems are shown in Section 6, and we conclude the paper in Section 7.

2. Preliminaries

Let V denote a finite set of objects v, and let E be a family of subsets e of V such that $\cup_{e\in E} = V$. Then we call $G = (V, E)$ a *hypergraph* with *vertex* set V and *hyperedge* set E. A hyperedge containing just two vertices is simply a simple graph edge. Given a set S, let $|S|$ denote the cardinality of S. Then the size of vertices is denoted by $|V|$, and the size of hyperedges is $|E|$. A *weighted hypergraph* is a hypergraph that has a positive number $w(e)$ associated with each hyperedge e, called the *weight* of hyperedge e. A hyperedge e is said to be *incident* with a vertex v when $v \in e$. For a vertex $v \in V$, the *degree* of v is defined by

$$d(v) = \sum_{\{e\in E|v\in e\}} w(e).$$

For a hyperedge $e \in E$, the degree is defined to be

$$\delta(e) = |e|.$$

We say that there is a *hyperpath* between vertices v_1 and v_k when there is an alternative sequence of distinct vertices and hyperedges $v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k$ such that $\{v_i, v_{i+1}\} \subseteq e_i$ for $1 \leq i \leq k-1$. A hypergraph is *connected* if there is a path for every pair of vertices. In the following, the hypergraphs are always assumed to be connected. A hypergraph G can be represented by a $|V| \times |E|$ matrix H with entries $h(v, e) = 1$ if $v \in e$ and 0 otherwise, called the *incidence matrix* of G. Then

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

and

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

Let D_v and D_e denote the diagonal matrices containing the vertex and hyperedge degrees respectively. Let W denote the diagonal matrix containing the weights. Then the *adjacency matrix* A of G is defined as $A = HWH^T - D_v$, where H^T is the transpose of H.

3. Regularization Framework

Given a hypergraph $G = (V, E)$, the vertices in a nonempty subset $S \subset V$ are labeled as positive or negative. Assume a classification function f over V, which classifies each vertex v as sign $f(v)$. We may think of f as a vector in Euclidean space $\mathbb{R}^{|V|}$. On the one hand, we want to assign the same labels to vertices which have many incident hyperedges in common. Thus we define a functional

$$\Omega(f) = \frac{1}{2}\sum_{e\in E}\frac{1}{\delta(e)}\sum_{\{u,v\}\subseteq e} w(e)\left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2. \tag{3.1}$$

The functional sums the changes of a function over the hyperedges of the hypergraph. On the other hand, the initial label assignment should be changed as little as possible. Let

y denote the function in $\mathcal{H}(V)$ defined by $y(v) = 1$ or -1 if vertex v has been labeled as positive or negative respectively, and 0 if it is unlabeled. Thus we may consider the optimization problem

$$\operatorname*{argmin}_{f\in\mathbb{R}^{|V|}}\{\Omega(f) + \mu\|f - y\|^2\}, \tag{3.2}$$

where $\mu > 0$ is the parameter specifying the tradeoff between the two competitive terms.

We will recover this regularizer (3.1) from a natural combinatorial optimization problem in Section 4, which can further be derived from a viewpoint of random walks in Section 5. For a simple graph, each edge is incident with only two vertices and thus $\delta(e) = 2$, so equation(3.1) reduces to

$$\Omega(f) = \frac{1}{4}\sum_{e=\{u,v\}\in E} w(u,v)\left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2,$$

which is consistent with the regularizer for transductive inference that we proposed earlier (Zhou et al., 2004) up to a factor 1/2.

Define a matrix $\Delta = I - D_v^{-1/2}HWD_e^{-1}H^TD_v^{-1/2}$, where I denotes the identity. Then

$$\Omega(f) = \langle f, \Delta f\rangle, \tag{3.3}$$

which can be shown as follows:

$$\begin{aligned}
\Omega(f) &= \sum_{e\in E}\sum_{u,v\in V}\frac{w(e)h(u,e)h(v,e)}{\delta(e)}\left(\frac{f^2(u)}{d(u)} - \frac{f(u)f(v)}{\sqrt{d(u)d(v)}}\right)\\
&= \sum_{e\in E}\sum_{u\in V}\frac{w(e)h(u,e)f^2(u)}{d(u)}\sum_{v\in V}\frac{h(v,e)}{\delta(e)} - \sum_{e\in E}\sum_{u,v\in V}\frac{w(e)h(u,e)h(v,e)}{\delta(e)}\frac{f(u)f(v)}{\sqrt{d(u)d(v)}}\\
&= \sum_{u\in V}f^2(u)\sum_{e\in E}\frac{w(e)h(u,e)}{d(u)} - \sum_{e\in E}\sum_{u,v\in V}\frac{f(u)w(e)h(u,e)h(v,e)f(v)}{\sqrt{d(u)d(v)}\delta(e)}\\
&= \sum_{u\in V}f^2(u) - \sum_{e\in E}\sum_{u,v\in V}\frac{f(u)w(e)h(u,e)h(v,e)f(v)}{\sqrt{d(u)d(v)}\delta(e)}.
\end{aligned}$$

Equation(3.3) also shows that the matrix Δ is positive semi-definite. It is easy to verify that the smallest eigenvalue of Δ is 0, and the corresponding eigenvector is $\sqrt{d}$.

For an undirected simple graph, the edge degree matrix D_e reduces to $2I$. Thus

$$\Delta = I - \frac{1}{2}D_v^{-1/2}HWH^TD_v^{-1/2} = I - \frac{1}{2}D_v^{-1/2}\left(D_v + A\right)D_v^{-1/2} = \frac{1}{2}\left(I - D_v^{-1/2}AD_v^{-1/2}\right),$$

which coincides with the usual definition of the graph Laplacian for an undirected simple graph (Chung, 1997) up to a factor of 1/2. Hence we call the matrix Δ defined by (3.3) as the Laplacian for a hypergraph. It is might be interesting to compare our hypergraph Laplacian with an alternative that we found from mathematical literature (Rodríguez, 2002). Given an unweighted hypergraph $G = (V, E)$, and two distinct vertices $u, v \in V$, Rodríguez (2002) defined each entry $a(u, v)$ of the adjacency matrix A as the number of

hyperedges containing both u and v. Then he defined a diagonal matrix with diagonal elements $D(v,v) = \sum_{u,v} a(u,v)$, and the hypergraph Laplacian $\Delta = D - A$. Note that $D(v)$ is generally not equal to the number of the hyperedges attached with vertex v. In the following sections, we will further show that our hypergraph Laplacian is well motivated.

Let f^* denote the solution of (3.2). From (3.3), differentiating (3.2) with respect to function f, we have $\Delta f^* + \mu(f^* - y) = 0$. It is a linear equation. As in (Zhou et al., 2004, 2005), we can obtain a closed-form solution

$$f^* = (1-\alpha)(I - \alpha\Theta)^{-1} y, \tag{3.4}$$

where $\alpha = 1/(1+\mu)$ and $\Theta = D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}$. For computing the involved matrix inverse, we may refer the reader to (Spielman and Teng, 2003) for a nearly linear time numerical technique.

We may generalize this regularization framework to *directed hypergraphs* (Ausiello et al., 2001) as the regularization framework for undirected simple graphs (Zhou et al., 2004) is generalized to directed simple graphs (Zhou et al., 2005). A directed hypergraph is a hypergraph in which each hyperedge e is an ordered pair (X, Y) where $X \subseteq V$ is the *tail* of e and $Y \subseteq V \setminus X$ is its *head*.

4. Hyperspectral Clustering

In the absence of labeled instances, this framework can be utilized in an unsupervised setting as a spectral clustering approach for hypergraphs, which generalizes the powerful spectral partitioning methodology for undirected simple graphs (Shi and Malik, 2000).

For a vertex subset $S \subset V$, let S^c denote the compliment of S. A cut of a hypergraph $G = (V, E)$ is a partition of V into two parts S and S^c. We say that a hyperedge e is cut if it is incident with the vertices in S and S^c simultaneously. We define the *hyperedge boundary* ∂S of S by $\partial S := \{e \in E | e \cap S \neq \emptyset, e \cap S^c \neq \emptyset\}$. For a vertex subset $S \subset V$, we define the *volume* $\operatorname{vol} S$ of S by

$$\operatorname{vol} S := \sum_{v \in S} d(v),$$

and the volume of ∂S by

$$\operatorname{vol} \partial S := \sum_{e \in \partial S} w(e) \frac{|e \cap S|\,|e \cap S^c|}{\delta(e)}.$$

Clearly, $\operatorname{vol} \partial S = \operatorname{vol} \partial S^c$. Intuitively, a good partition is to cut as few hyperedges as possible to disconnect the hypergraph into two subgraphs such that each subgraph is as dense as possible. Thus we may consider the problem

$$\operatorname*{argmin}_{\emptyset \neq S \subset V} \operatorname{Ncut}(S) := \operatorname{vol} \partial S \left(\frac{1}{\operatorname{vol} S} + \frac{1}{\operatorname{vol} S^c} \right). \tag{4.5}$$

For an undirected simple graph, $|e \cap S| = |e \cap S^c| = 1$, and $\delta(e) = 2$, so the right-hand side of equation (5) reduces to the usual normalized cut (Shi and Malik, 2000) up to a factor $1/2$.

In the following, we relax (4.5) into a real-valued optimization problem to obtain an approximate solution of (4.5). Let r be an indicator function with $r(v) = 1$ if $v \in S$ and -1 if $v \in S^c$. For a cut edge $e \in \partial S$, and the vertices $\{u, v\} \subseteq e, u \in S, v \in S^c$, $(r(u)-r(v))^2 = 4$. Otherwise, $(r(u) - r(v))^2 = 0$. Let γ denote the ratio $\operatorname{vol} S / \operatorname{vol} V$. Then the cut criterion may be written

$$\operatorname{Ncut}(S) = \frac{\sum_{e\in E} \frac{1}{\delta(e)} \sum_{\{u,v\}\subseteq e} w(e)\,(r(u) - r(v))^2}{8\gamma(1-\gamma) \sum_{v\in V} r^2(v)d(v)}.$$

Define another function s with $s(v) = 2(1 - \gamma)$ if $v \in S$, and -2γ if $v \in S^c$. Clearly, for all $u, v \in V$, $\operatorname{sign} s(v) = \operatorname{sign} r(v)$, and $s(u) - s(v) = r(u) - r(v)$. In addition,

$$\sum_{v\in V} d(v)s(v) = \sum_{v\in V} d(v)(r(v) + (1 - 2\gamma)) = 0,$$

and

$$\sum_{v\in V} s^2(v)d(v) = 4\gamma(1-\gamma) \sum_{v\in V} r^2(v)d(v).$$

Thus

$$\begin{aligned}
\operatorname{Ncut}(S) &= \frac{\sum_{e\in E} \frac{w(e)}{\delta(e)} \sum_{\{u,v\}\subseteq e} (s(u) - s(v))^2}{2 \sum_{v\in V} s^2(v)d(v)} \\
&= \frac{\sum_{e\in E} \frac{w(e)}{\delta(e)} \sum_{\{u,v\}\subseteq e} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2}{2 \sum_{v\in V} f^2(v)},
\end{aligned}$$

where $f = \sqrt{d}s$. Clearly, $\operatorname{sign} f(v) = \operatorname{sign} s(v)$ for all $v \in V$. Let the elements of f take any continuous values. Then the combinatorial optimization problem (4.5) may be relaxed into

$$\begin{aligned}
&\operatorname*{argmin}_{f\in\mathbb{R}^{|V|}} \Omega(f) \\
&\text{subject to} \quad \|f\| = 1,\ \langle f, \sqrt{d}\rangle = 0.
\end{aligned} \tag{4.6}$$

Note that $\sqrt{d}$ is an eigenvector of Δ with the smallest eigenvalue 0. Therefore the solution of (4.6) is an eigenvector Φ with the second smallest eigenvalue. Consequently the vertices is partitioned into the two parts $S = \{v \in V | \Phi(v) \geq 0\}$ and $S^c = \{v \in V | \Phi(v) < 0\}$.

It is easy to extend the hyperspectral clustering approach to k-partition. Assume a k-partition to be $V = V_1 \cup V_2 \cup \cdots \cup V_k$, where $V_i \cap V_j = \emptyset$ for all $1 \leq i, j \leq k$. Let P_k denote a k-partition. Then we may obtain a k-partition by minimizing

$$\operatorname{Ncut}(P_k) = \sum_{1\leq i\leq k} \frac{\operatorname{vol} \partial V_i}{\operatorname{vol} V_i}. \tag{4.7}$$

Similarly, it can be shown that the solution of the corresponding relaxed optimization problem of (4.7) can be any orthonormal basis of the linear space spanned by the eigenvectors of Δ associated with the k smallest eigenvalues.

5. Random Walks

We first define a natural random walk over a hypergraph. Then the combinational cut criterion is interpreted in terms of random walks.

The transition rule of the random walk is as follows: given that the current position is vertex $u \in V$, first choose a hyperedge e over the hyperedges incident with u with the probability proportional to the weight of e, and then a vertex $v \in e$ is selected uniformly at random. Let P denote the transition probability matrix of the random walk. Then each entry of P is

$$p(u,v) = \sum_{e \in E} w(e) \frac{h(u,e)}{d(u)} \frac{h(v,e)}{\delta(e)}. \tag{5.8}$$

In the matrix notation, we have $P = D_v^{-1} H W D_e^{-1} H^T$. The stationary distribution π of the random walk is

$$\pi(v) = \frac{d(v)}{\operatorname{vol} V},$$

which follows from that

$$\begin{aligned}
\sum_{u \in V} \pi(u) p(u,v) &= \sum_{u \in V} \frac{d(u)}{\operatorname{vol} V} \sum_{e \in E} \frac{w(e) h(u,e) h(v,e)}{d(u)\delta(e)} \\
&= \frac{1}{\operatorname{vol} V} \sum_{u \in V} \sum_{e \in E} \frac{w(e) h(u,e) h(v,e)}{\delta(e)} \\
&= \frac{1}{\operatorname{vol} V} \sum_{e \in E} w(e) \sum_{u \in V} h(u,e) \frac{h(v,e)}{\delta(e)} \\
&= \frac{1}{\operatorname{vol} V} \sum_{e \in E} w(e) h(v,e) \\
&= \frac{d(v)}{\operatorname{vol} V} \\
&= \pi(v).
\end{aligned}$$

Now we show how to understand the combinatorial cut criterion in terms of random walks. The cut criterion may be transformed into

$$\operatorname{Ncut}(S) = \frac{\operatorname{vol} \partial S}{\operatorname{vol} V} \left(\frac{1}{\operatorname{vol} S / \operatorname{vol} V} + \frac{1}{\operatorname{vol} S^c / \operatorname{vol} V} \right).$$

From the closed-form expression of the stationary distribution, we have

$$\frac{\operatorname{vol} S}{\operatorname{vol} V} = \sum_{v \in S} \frac{d(v)}{\operatorname{vol} V} = \sum_{v \in V} \pi(v),$$

which is the probability of the random walk occupying some vertex in S. Moreover,

$$\begin{aligned}
\frac{\operatorname{vol}\partial S}{\operatorname{vol} V} &= \sum_{e\in\partial S}\frac{w(e)}{\operatorname{vol} V}\frac{|e\cap S|\,|e\cap S^c|}{\delta(e)} \\
&= \sum_{e\in\partial S}\sum_{u\in e\cap S}\sum_{v\in e\cap S^c}\frac{w(e)}{\operatorname{vol} V}\frac{h(u,e)h(v,e)}{\delta(e)} \\
&= \sum_{e\in\partial S}\sum_{u\in e\cap S}\sum_{v\in e\cap S^c} w(e)\frac{d(u)}{\operatorname{vol} V}\frac{h(u,e)}{d(u)}\frac{h(v,e)}{\delta(e)} \\
&= \sum_{u\in S}\sum_{v\in S^c}\frac{d(u)}{\operatorname{vol} V}\sum_{e\in S} w(e)\frac{h(u,e)}{d(u)}\frac{h(v,e)}{\delta(e)} \\
&= \sum_{u\in S}\sum_{v\in S^c}\pi(u)p(u,v).
\end{aligned}$$

This is the probability with which one sees a jump of the random walk from S to S^c. Therefore the combinatorial cut criterion may be intuitively understood as follows: looking for a cut such that, under the stationary distribution, the probability of transition from one cluster to another is as small as possible, while the probability of remaining in the same cluster is as large as possible. It is worth mentioning that there is a similar random walk view for the spectral clustering approach for undirected simple graphs (Meilă and Shi, 2001).

6. Experiments

We apply the hypergraph based approach to four real-world problems, and compare it with the method in (Zhou et al., 2004), which is a powerful transductive approach operating on data with pairwise relationships, and built on the spectral clustering algorithm of Shi and Malik (Shi and Malik, 2000). All dataset except the 20-newsgroup one are from the UCI depository. The instances in the datasets are described as one or more attributes. For the hypergraph based approach, each attribute value is thought of as a hyperedge. For instance, in the 20-newsgroup dataset, each word is regarded as a hyperedge, and each document a vertex. The weights of the hyperedges are simply set to 1. For the baseline, the pairwise relationships is naturally defined to be the adjacency matrix of the hypergraph.

The first task is clustering on the zoo dataset. It contains 100 animals with 17 Boolean-valued attributes. The attributes indicate whether the animals fly, milk, live aquatically and so on. The animals are classified into 7 classes. We embed the dataset into Euclidean space using the eigenvectors of the hypergraph Laplacian (Fig. 2). Clearly, the animals are well separated by the first three eigenvectors.

The second task is classification on the mushroom dataset. It contains information of physical characteristics of mushrooms. The data contains 8124 instances described by 22 categorical attributes, such as shape, color, etc. Each attribute has several categorical values. We remove the 11th attribute which has missing values. Each instance is labeled as *edible* or *poisonous*. The two classes have have 4208 and 3916 instances separately.

The third task is text categorization. We use the modified 20-newsgroup dataset with binary occurrence value for 100 words across 16242 text news.[1] It is classified into 4 different classes corresponding the highest level of the original 20 newsgroups, which contains $4605, 3519, 2657$ and 5461 articles respectively.

The final task is letter recognition. The letter dataset records images of capital letter (from A to Z) based on 20 different fonts and they are randomly distorted. Each data point is associated with 16 integer attributes extracted from raster scan image of the letter. We use a subset of the dataset containing letters from A to E with $789, 766, 736, 805$ and 768 letters for the five classes respectively.

The experimental results of the last three tasks are shown in Fig. 3(a)-3(c). The regularization parameter α for both approaches is fixed at 0.1. Each test error is averaged over 20 trails. In each trail, we randomly select some points to get labeled. For each class, we have to have at least one labeled point. Otherwise, we sample again. The results clearly show that the hypergraph based method is consistently better than the baseline. The influence of the α for the letter recognition task is shown in Fig. 3(d). It is interesting that the α influences the performance of the baseline much more than the hypergraph based approach.

7. Conclusion

We have proposed a general framework for learning from a hypergraph which naturally models complex relationships among data. In the absence of labeled instances, this framework reduces to a spectral clustering approach for hypergraphs. Moreover, we defined a natural random walk on hypergraphs to interpret this framework in an intuitive fashion. Experiments on real-world datasets demonstrated that the hypergraph based approach is significantly better than that based on a pairwise description. In the side of mathematics, our work essentially extended the usual spectral graph theory (Chung, 1997) to the context of hypergraphs.

It is interesting to consider applying the same framework to a broader range of practical problems. For instance, the interactions among proteins are generally modeled as a protein-protein interaction network. This representation does not take into account multi-protein complexes however.

References

G. Ausiello, P.G. Franciosa, and D. Frigioni. Directed hypergraphs: problems, algorithmic results, and a novel decremental approach. In *Proc. of the 7th Italian Conference on Theoretical Computer Science*, volume 2202 of *Lecture Notes In Computer Science*, pages 312–327. Springer-Verlag, London, UK, 2001.

F. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, Providence, RI, 1997.

M. Meilă and J. Shi. A random walks view of spectral segmentation. In *Proc. 8th International Workshop on Artificial Intelligence and Statistics*, 2001.

1. See http://www.cs.toronto.edu/~roweis/data/20news_w100.mat.

J. A. Rodríguez. On the Laplacian eigenvalues and metric parameters of hypergraphs. *Linear and Multilinear Algebra*, 50(1):1–14, 2002.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

D. Spielman and S. Teng. Solving sparse, symmetric, diagonally-dominant linear systems in time $o(m^{1.31})$. In *Proc. 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.

D. Zhou, O. Bousquet, T.N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proc. 22nd International Conference on Machine Learning*, 2005.

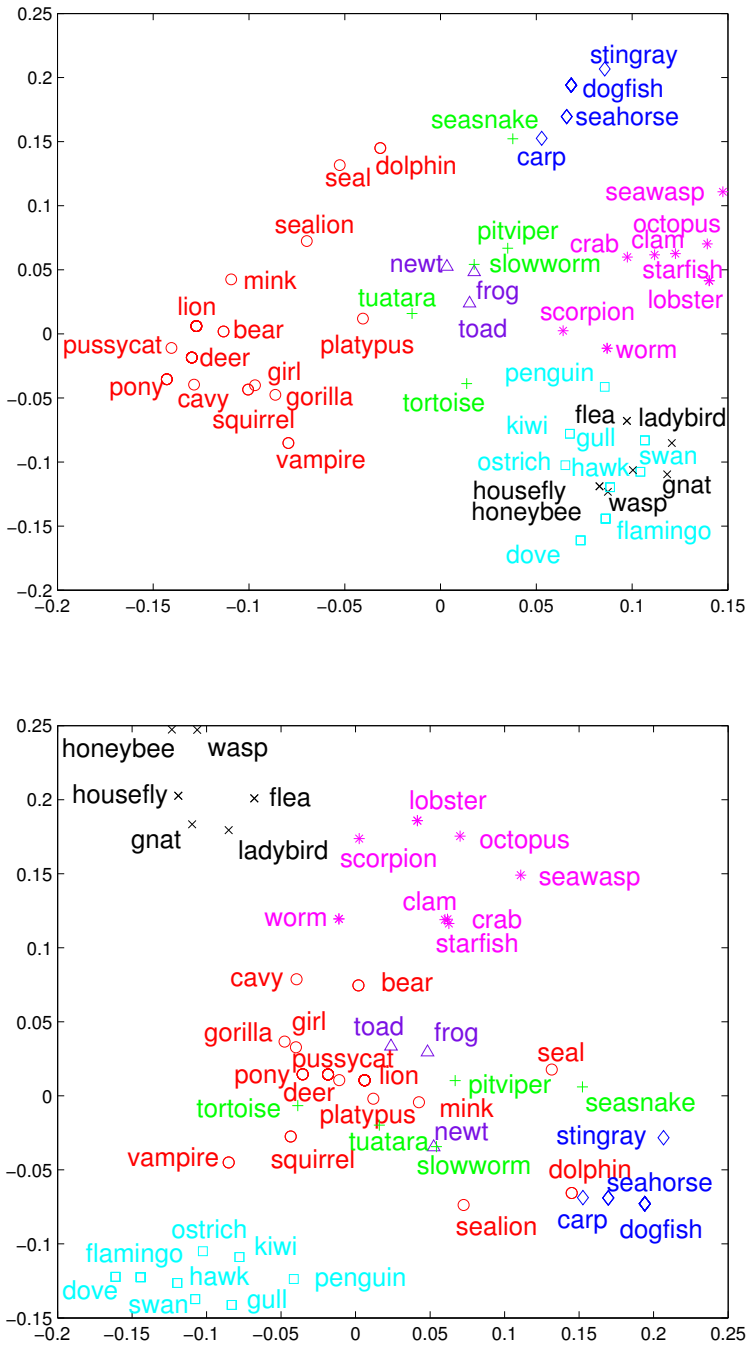
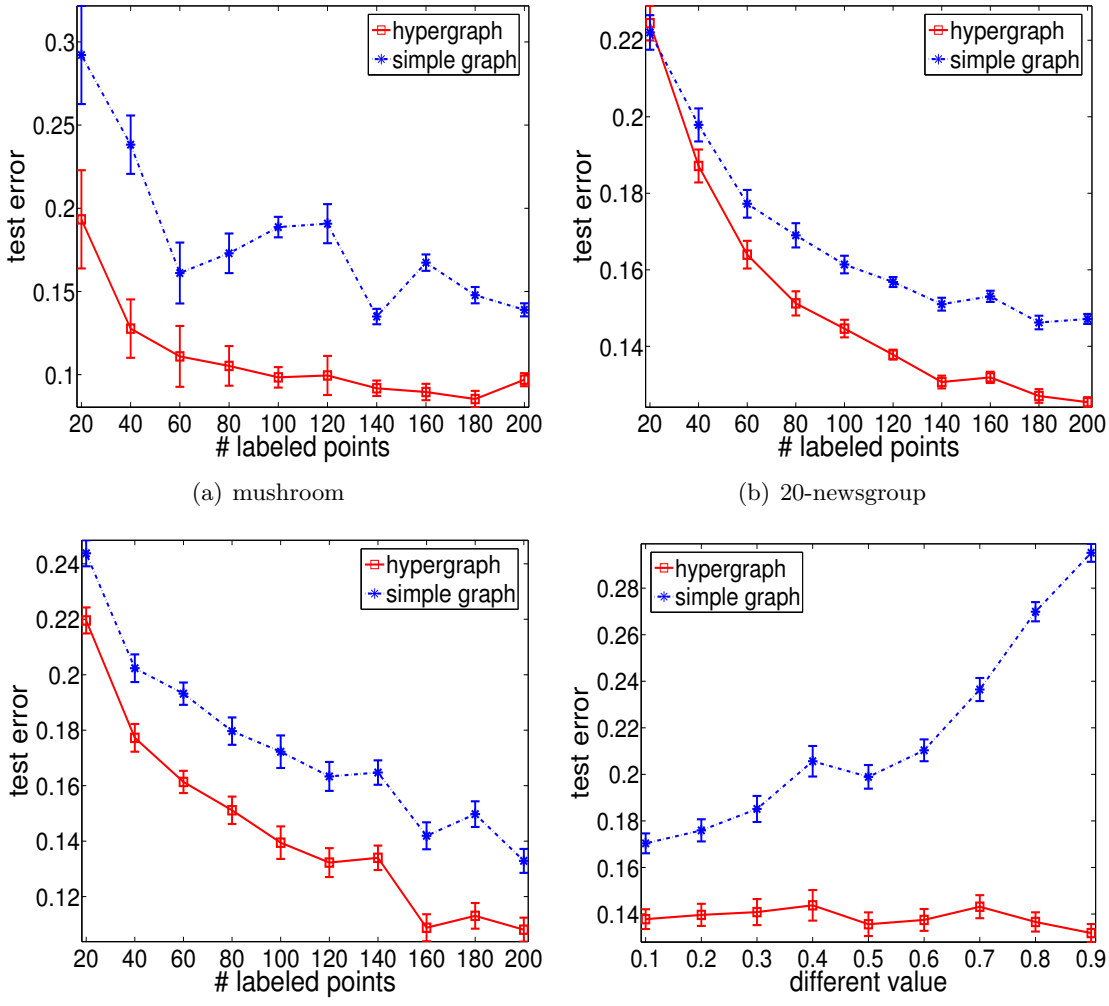


Figure 2: Embedding the zoo dataset into Euclidean space. Left panel: the two eigenvectors of the hypergraph Laplacian corresponding to the second and third smallest eigenvalues; right panel: the two eigenvectors of the hypergraph Laplacian corresponding to the third and fourth smallest eigenvalues. For animals having the same attributes, we randomly choose one as their representative to put in the figures. It is worth noticing that the animals like dolphin are between class 1 (denoted by $\circ$) containing the animals mostly milking and living on land, and class 4 (denoted by $\diamond$) containing the animals living in sea.



(a) mushroom

(b) 20-newsgroup

(c) letter

(d) α (letter)

Figure 3: Classification on the datasets with complex relationships. Fig. (a)-(c) depict the test errors of the hypergraph based approach and the baseline on three different datasets. The number of the labeled instances for each dataset is increased from 20 to 200. Fig. (d) illustrates the influence of the regularization parameter α in the letter recognition task with 100 labeled instances.