

Spectral Embedding of Feature Hypergraphs

Peng Ren, Richard C. Wilson, and Edwin R. Hancock

Department of Computer Science, The University of York, York, YO10 5DD, UK
{pengren,wilson,erh}@cs.york.ac.uk

Abstract. In this paper we investigate how to establish a hypergraph model for characterizing object structures and how to embed this model into a low-dimensional pattern space. Each hyperedge of the hypergraph model is derived from a seed feature point of the object and embodies those neighbouring feature points that satisfy a similarity constraint. We show how to construct the Laplacian matrix of the hypergraph. We adopt the spectral method to construct pattern vectors from the hypergraph Laplacian. We apply principal component analysis (PCA) to the pattern vectors to embed them into a low-dimensional space. Experimental results show that the proposed scheme yields good clusters of distinct objects viewed from different directions.

1 Introduction

Graph-based methods have played an important role in machine learning due to their ability to represent relational patterns. Various applications of graph-based methods can be found in data mining, natural language processing and information retrieval. In computer vision, the attractive feature of graph representations is that they concisely capture the relational arrangement of object primitives in a manner which can be invariant to changes in object viewpoint[1].

However, in many situations the graph representation for relational patterns is incomplete. This is because in real-world problems objects and their features tend to exhibit multiple relationships rather than simple pairwise ones. Since graphs are of limited use in representing complex relational patterns, hypergraphs have been adopted as a more sophisticated alternative.

Hypergraphs have certain advantages over graphs. These include their accuracy and completeness in describing feature relations and structures. As a result, hypergraphs have recently found applications in a variety of practical problems such as partitioning circuits[7] and clustering categorical data[6]. In computer vision, hypergraphs have been applied to image segmentation[3], edge detection[4] and noise removal[4]. However, most of these applications are based on low-level approximated image analysis, and thus induce a heavy computation burden.

In addition, to be easily manipulated and hence useful in machine learning, hypergraphs must be represented in an mathematically consistent way, e.g. as matrices or vectors. Since Chung's definition on the Laplacian matrix of a k -uniform hypergraph[5], there have been several investigations of the matrix representation of hypergraphs[9][11] [14]. However, none of the proposed mathematical representations has been generally accepted as a basis for the spectral manipulation of

hypergraphs. In this paper, we develop a spectral hypergraph embedding method. The starting point is to show how to encode hyperedge structure in an improved version of the hypergraph Laplacian in [14]. The pattern vectors are then established by using the Laplacian eigenmodes, which is proved to be permutation invariant to both the vertex labels and hyperedge labels. We apply the proposed method to clustering objects based on a high-level feature-based analysis of images. The proposed method gives good object clusters.

This paper is organized as follows. In Section 2, we review the relevant background on hypergraphs. In Section 3, we develop our hypergraph model for representing the object structure. In Section 4, we introduce a Laplacian matrix representation of hypergraphs. In Section 5, we develop a spectral embedding scheme for hypergraphs by using the associated Laplacian matrix. Section 6 gives experimental results for the proposed scheme on clustering visual information. Finally, conclusions are presented in Section 7.

2 Hypergraph Fundamentals

A hypergraph is a generalization of a graph. Unlike the edge of a graph, which can connect only two vertices, the edge of a hypergraph, i.e. hyperedge, can connect any number of vertices. A hypergraph is normally defined as a pair $H(V, E)$ where V is a set of elements, called nodes or vertices, and E is a set of non-empty subsets of V called hyperedges.

An example of a hypergraph is shown in Figure 1(a), where the vertex set $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and the hyperedge set $E = \{e_1 = \{v_1, v_2, v_3\}, e_2 = \{v_3, v_4, v_5\}, e_3 = \{v_5, v_6\}\}$. The representation of a hypergraph in the form of sets, concretely captures the relationship between vertices and hyperedges. However, it is difficult to manipulate this form in a computationally uniform way. Thus one alternative representation of a hypergraph is in the form of a matrix. For a hypergraph with m hyperedges and n vertices, we establish a $m \times n$ matrix H with element $h_{i,j}$ as follows:

$$h_{i,j} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

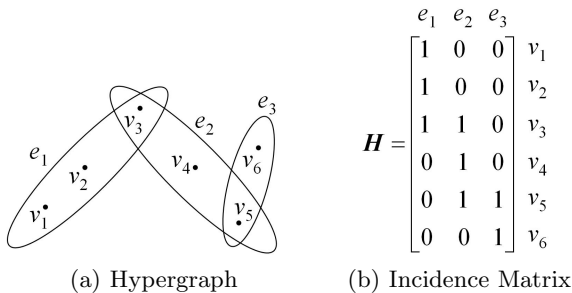


Fig. 1. An Example of Hypergraphs

Here, \mathbf{H} is referred to as the incidence matrix of the hypergraph $H(V, E)$. The incidence matrix can be more easily manipulated than its equivalent set representation. The incidence matrix of the example hypergraph is shown in Figure 1(b), where each column is associated with the corresponding hyperedge, and each row with the corresponding vertex.

3 Hypergraph Representation of Objects

There has been a considerable body of research aimed at how to represent visual information using relational graphs[10][13]. However, these methods mainly focus on the representation of the relational arrangement of object primitives but neglect the intrinsic characteristics of the feature points (such as their colors and intensities). On the other hand, the hypergraph-based image representation proposed by Bretto *et al* incorporates the intensity levels of picture elements into the neighbourhood description for edge detection and denoising[4]. However, the low-level representation is pixel-based it is computationally burdensome.

To overcome these problems, in this paper, we develop a high-level hypergraph representation, i.e. feature hypergraph, based on feature points, rather than the pixel contents of an image. We exploit the intensities of feature points to characterize the similarities between different image locations. The hypergraph associated with one object is established with m vertices and m hyperedges, where m is the cardinality of the feature point set $V = \{v_1, v_2 \cdots v_m\}$ extracted from the object. Each feature point is represented by one vertex in the hypergraph and therefore the feature point set is identical to the hypergraph vertex set (denoted by V). Each hyperedge is specific to one vertex, which we define as the seed of the hyperedge. To determine the elements of each hyperedge in addition to the seed, we measure the similarities between the intensities of the seed and of the other vertices which lie in the spatial neighbourhood of the seed to confine the extent of the hyperedge. Let $\mathbf{c}(v_i)$ denote the spatial coordinate of the feature point v_i in an image, $I(v_i)$ denote the intensity of v_i , Th_{j1} be a fixed value which represents the distance threshold for neighbourhood, and Th_{j2} be the similarity threshold between the vertex intensities. The threshold Th_{j2} could be either a fixed value or calculated in an adaptive way[4]. The association of the vertex v_i to the hyperedge $E(v_j)$ derived from the seed v_j is determined as follows:

$$v_i \begin{cases} \in E(v_j) & \text{if } \|\mathbf{c}(v_i) - \mathbf{c}(v_j)\| \leq Th_{j1} \text{ and if } |I(v_i) - I(v_j)| \leq Th_{j2} \\ \notin E(v_j) & \text{otherwise} \end{cases} \quad (2)$$

The first condition confines one set of candidate vertices to the spatial neighbourhood of the seed. The second condition selects a set of candidate vertices satisfying the similarity constraint. The intersection of the two sets constitute one hyperedge. We hereby refer to the hypergraph established in (3) as the feature hypergraph of an object.

The incidence matrix of the feature hypergraph is hence of size $m \times m$. The i th column represents the hyperedge derived from v_i , and the j th row corresponds to vertex v_j . Thus the element of incidence matrix is denoted as:

$$h(i, j) = \begin{cases} 1 & \text{if } \|c(v_i) - c(v_j)\| \leq Th_{j1} \text{ and if } |I(v_i) - I(v_j)| \leq Th_{j2} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

4 Laplacian Matrix of a Hypergraph

Although the incidence matrix can fully describe the characteristics of the constructed hypergraph, it poses some drawbacks. Firstly, the elements of the incidence matrix represent a vertex-to-hyperedge relationship rather than a vertex-to-vertex relationship. Secondly, it is not possible to perform spectral analysis on the incidence matrix of a hypergraph, since the incidence matrix is normally not square (although that of our hypergraph is square). For these reasons, we establish the hypergraph adjacency matrix and Laplacian matrix.

4.1 Adjacency Matrix

For a hypergraph with a incidence matrix \mathbf{H} , the associated adjacency matrix \mathbf{A} is:

$$\mathbf{A} = \mathbf{H}\mathbf{H}^T - \mathbf{D}_v \quad (4)$$

where \mathbf{H}^T is the transpose of \mathbf{H} and \mathbf{D}_v is the diagonal vertex degree matrix whose diagonal element $d(v_i)$ is the summation of the i th row of \mathbf{H} . The (i, j) th entry of $\mathbf{H}\mathbf{H}^T$ records the number of hyperedges which contain both v_i and v_j .

An interpretation of (4) can be given in terms of the associated bipartite graph of the hypergraph[12]. To establish the associated bipartite graph, we use a dual representation in which each hyperedge is represented by a new vertex. The new vertex is incident to every original vertex in the corresponding hyperedge. The new vertex set and the original vertex set therefore constitute the associated bipartite graph with the new vertices corresponding to hyperedges on one side and the original vertices on the other side. The (i, j) th entry of the adjacency matrix \mathbf{A} is the number of directed paths with length 2 from v_i to v_j in the bipartite graph.

The adjacency matrix of a hypergraph defined in (4) characterizes it in a way that is essentially analogous to that of a graph, since it encodes the incidence

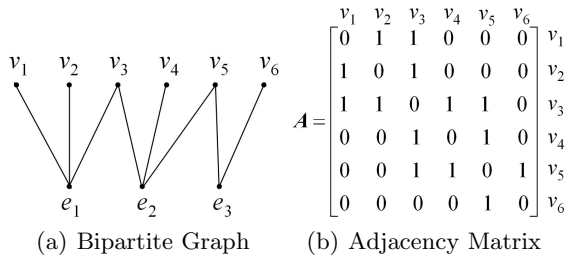


Fig. 2. The Bipartite Graph and The Adjacency Matrix of Figure 1(a)

relations between vertices. In the extreme situation of all hyperedges with cardinality 2, the adjacency matrix of a hypergraph reduces to the adjacency matrix of a graph.

The associated bipartite graph representation of the example hypergraph is shown in Figure 2, together with its associated adjacency matrix. The adjacency matrix can be computed either from (4) or by using the bipartite graph method. As an example of the bipartite graph method, the (1, 3)th entry of the adjacency matrix is 1 because there is only one directed path $v_1e_1v_3$ with length 2 from v_1 to v_3 .

4.2 Laplacian Matrix

Recently, Zhou *et al* have detailed a hypergraph Laplacian that is suitable for spectral analysis[14]. However, there are some drawbacks with their Laplacian when it is used to deal with vision problems. One of these is that the definition involves the diagonal hyperedge degree matrix to guarantee the positive semi-definiteness of the hypergraph Laplacian. This is necessary to ensure that the embedding coordinates of vertices obtained from the spectrum of the Laplacian is real. However, the diagonal hyperedge degree matrix contains redundant information. Another problem with the definition of Zhou *et al* is that it involves normalizing the hypergraph Laplacian in order to ensure that the Fiedler vector has the properties that lend it to bi-partitioning. However, in our preliminary experiments we found that the normalized Laplacian is not effective in distinguishing relational data[15]. To overcome these drawbacks, we define the hypergraph Laplacian by analogy with the graph Laplacian, as follows:

$$\mathbf{L} = \mathbf{D}_v - \mathbf{A} = 2\mathbf{D}_v - \mathbf{H}\mathbf{H}^T \quad (5)$$

This definition does not involve the diagonal hyperedge degree matrix, which as noted above contains redundant information. Furthermore, we do not normalize our Laplacian since the unnormalized Laplacian has been shown to be superior in distinguishing relational data[15]. Therefore, the hypergraph Laplacian in (5) can be regarded as an improvement over Zhou's hypergraph Laplacian, which is more concise in characterizing the structures of hypergraphs. It is also can be regarded as a star expansion representation of a hypergraph according to [2].

5 Spectral Embedding of Hypergraphs

The structure of a hypergraph is invariant under permutations of the node and hyperedge labels. However, the incidence matrix is modified by the node order and hyperedge order since the rows and columns are respectively indexed by the node order and hyperedge order. If we relabel the nodes, the incidence matrix undergoes a permutation of rows, while if we relabel the hyperedges, the incidence matrix undergoes a permutation of columns. The associated adjacency matrix and Laplacian matrix, which represent the incident relationship between nodes, might thus also undergo the permutation of rows and columns when there

is a reordering of nodes or hyperedges. Hence, the problems to be solved here are a) how to establish a set of feature vectors that are permutation invariant to the change of the node and hyperedge order and b) how to embed them into a low-dimensional space.

5.1 Pattern Vectors

Suppose that \mathbf{H} and \mathbf{H}_P are the incidence matrices associated with two isomorphic hypergraphs $H(V, E)$ and $H_P(V, E)$ respectively. Let the matrix \mathbf{P}_V be the node permutation matrix representing the change in node order between $H(V, E)$ and $H_P(V, E)$ and \mathbf{P}_E the hyperedge permutation matrix representing the change in hyperedge order. Both \mathbf{P}_V and \mathbf{P}_E are orthogonal matrices and as a result $\mathbf{P}_V \mathbf{P}_V^T = \mathbf{I}_V$ and $\mathbf{P}_E \mathbf{P}_E^T = \mathbf{I}_E$, where \mathbf{I}_V and \mathbf{I}_E are identity matrices of the same size with \mathbf{P}_V and \mathbf{P}_E respectively. The permutation relationship between the two incidence matrices can be denoted as $\mathbf{H}_P = \mathbf{P}_V \mathbf{H} \mathbf{P}_E$. Thus, the relationship between the adjacency matrices of $H_P(V, E)$ and of $H(V, E)$ is:

$$\begin{aligned} \mathbf{A}_P &= \mathbf{H}_P \mathbf{H}_P^T - \mathbf{D}_{vP} = \mathbf{P}_V \mathbf{H} \mathbf{P}_E (\mathbf{P}_V \mathbf{H} \mathbf{P}_E)^T - \mathbf{P}_V \mathbf{D}_v \mathbf{P}_V^T \\ &= \mathbf{P}_V \mathbf{H} \mathbf{P}_E \mathbf{P}_E^T \mathbf{H}^T \mathbf{P}_V^T - \mathbf{P}_V \mathbf{D}_v \mathbf{P}_V^T = \mathbf{P}_V (\mathbf{H} \mathbf{H}^T - \mathbf{D}_v) \mathbf{P}_V^T = \mathbf{P}_V \mathbf{A} \mathbf{P}_V^T \end{aligned} \quad (6)$$

where \mathbf{D}_{vP} is the diagonal vertex degree matrix of $H_P(V, E)$. The relationship between the Laplacian matrix of $H_P(V, E)$ and of $H(V, E)$ is:

$$\begin{aligned} \mathbf{L}_P &= 2\mathbf{D}_{vP} - \mathbf{H}_P \mathbf{H}_P^T = 2\mathbf{P}_V \mathbf{D}_v \mathbf{P}_V^T - \mathbf{P}_V \mathbf{H} \mathbf{P}_E (\mathbf{P}_V \mathbf{H} \mathbf{P}_E)^T \\ &= 2\mathbf{P}_V \mathbf{D}_v \mathbf{P}_V^T - \mathbf{P}_V \mathbf{H} \mathbf{P}_E \mathbf{P}_E^T \mathbf{H}^T \mathbf{P}_V^T = \mathbf{P}_V (2\mathbf{D}_v - \mathbf{H} \mathbf{H}^T) \mathbf{P}_V^T = \mathbf{P}_V \mathbf{L} \mathbf{P}_V^T \end{aligned} \quad (7)$$

From (6) and (7) it is clear that the hyperedge order has no effect on either the permutation of the adjacency matrix or the permutation of the Laplacian matrix, which are both solely controlled by the node permutation matrix.

Furthermore, let \mathbf{e} be a normalized eigenvector of \mathbf{L} with associated eigenvalue λ and let $\mathbf{e}_P = \mathbf{P}_V \mathbf{e}$. With these ingredients, we have that:

$$\mathbf{L}_P \mathbf{e}_P = \mathbf{P}_V \mathbf{L} \mathbf{P}_V^T \mathbf{P}_V \mathbf{e} = \mathbf{P}_V \mathbf{L} \mathbf{e} = \lambda \mathbf{P}_V \mathbf{e} = \lambda \mathbf{e}_P \quad (8)$$

From (8) \mathbf{e}_P is an eigenvector of \mathbf{L}_P with eigenvalue λ . Hence, the isomorphic hypergraphs $H(V, E)$ and $H_P(V, E)$ with changes in hyperedge order and node order both share the common Laplacian spectrum, i.e. the same set of eigenvalues of the Laplacian. Spectra from the adjacency matrices exhibit the same property and this can be proved in a similar way as (8).

We construct our pattern vectors by using the hypergraph Laplacian spectrum. The pattern vector of a hypergraph which is permutation invariant to node labels and hyperedge labels can thus be established by sorting the eigenvalues of the hypergraph Laplacian in an ascending order and taking the second

smallest up to the $(N + 1)$ th smallest ones as vector components. Although the hypergraphs studied might be of different sizes, thus inducing different dimensionalities of the eigenspaces, we can fix the number of eigenvalues used to restrict the dimension of pattern vectors to N . As a result, the pattern vectors of hypergraphs with different sizes can be formulated in the common form of $\mathbf{v} = [\lambda_2, \lambda_3 \cdots \lambda_{N+1}]^T$ without distinctions in length.

5.2 Embedding Using PCA

In Section 5.1, we construct the pattern vectors which both characterize the structures of the hypergraphs and that are invariant to label permutations. However, it is inconvenient to apply these pattern vectors to clustering objects and visualization when the vector dimension N is high. For this reason, we embed the pattern vector into a low-dimensional space by using principal component analysis (PCA). The purpose of this technique is to find a linear projection which describes the class differences rather than the overall variance of the data. Therefore the coordinates representing the original individual hypergraphs in the embedding space tend to be massed into clusters according to the class labels.

6 Experiments

In this section, we experiment with our new spectral representation of hypergraphs for the problem of clustering object views. All the experiments are undertaken based on the feature points extracted from objects by Harris detector[8]. We commence by comparing the proposed scheme with the traditional graph-spectral method. We embark on the experiments on a set of house images previously used in [10] and [13]. There are three classes of houses in the image set,

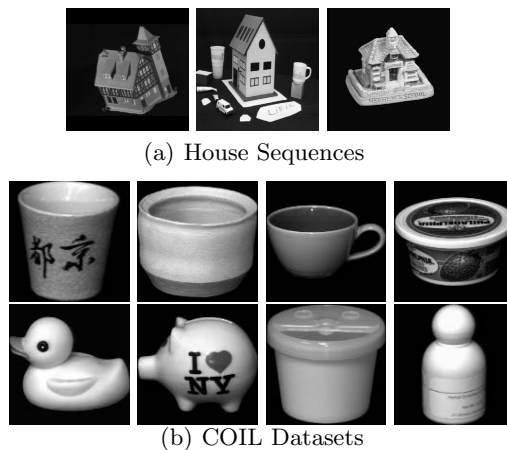


Fig. 3. Datasets for Experiments

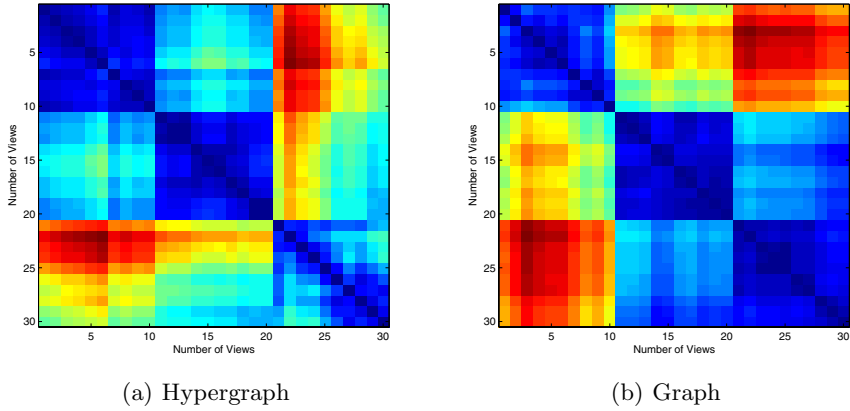


Fig. 4. Distance Map of Houses

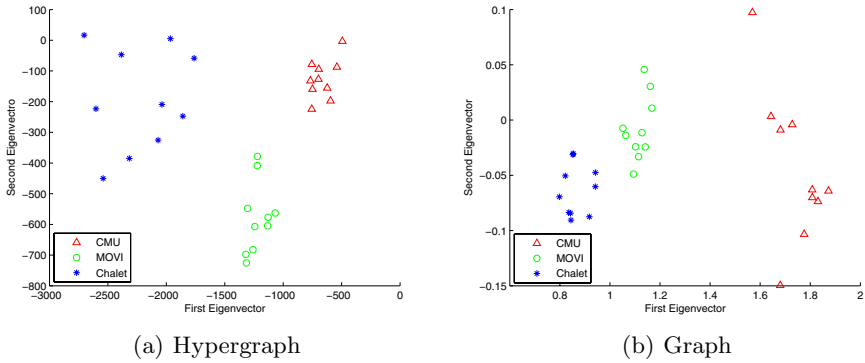


Fig. 5. Results of Clustering Houses

which are from the CMU database, the MOVI database and the chalet sequence. For each object, we have a sequence with different camera viewing directions. From the left to right, Figure 3(a) shows the sample image from the CMU database, the MOVI database and the chalet sequence. We establish the feature hypergraphs of objects as reported in Section 3 based on the feature points as vertices. The neighbourhood threshold Th_{j1} is set to $1/4$ the size of the image. The similarity threshold Th_{j2} is computed in the adaptive way. Then we establish the pattern vector for each feature hypergraph as presented in Section 5.1. Here we set the dimensionality of the pattern space $N = 6$. Figure 4(a) is the distance map of the proposed pattern vectors, and Figure 5(a) is the clustering results of the proposed scheme in the space spanned by the first two principal linear projections in PCA. The distance map and clustering results from the graph-spectral method are illustrated in Figure 4(b) and Figure 5(b) respectively. The graphs here used are Delaunay graphs of which the feature points

are the vertices. We can see from the experimental results that the proposed hypergraph-spectral method outperforms the traditional graph-spectral method in clustering the house sequences. We then applied the proposed method and the traditional graph-spectral method to the COIL database. There are 72 views for each object in the COIL database, acquired by rotating the object under study about a vertical axis. The sample images of different objects in COIL database are illustrated in Figure 3(b). The hypergraphs and graphs and their corresponding pattern vectors are established in the same way as the experiments on the house sequences. Figure 6 shows the first component of the pattern vectors, i.e. λ_2 , from the proposed hypergraph Laplacian and the graph Laplacian separately as a function of view number for the first five objects. The different curves correspond to different objects. The main feature to note is that the curves in the top plot, which is the result of our hypergraph-based method, is better separated than those in the bottom plot, which is the result of the graph-based method. All the above experiments imply that the hypergraph-spectral method is generally superior to the graph-spectral method in clustering visual information. This is because the graph based method mainly focuses on the relational arrangement of objects and neglects the similarity between feature points. The proposed hypergraph method, on other hand, incorporates a similarity measure between feature points as well as considering the arrangement of feature points in terms of neighbourhood relationship. Therefore, the hypergraph model represents richer sources of scene information about objects than the graph model.

Finally, we compare the performance of our hypergraph Laplacian and of Zhou’s hypergraph Laplacian[14] in clustering views of objects. We first establish the Laplacian matrices of the feature hypergraphs as reported in Section 4.2 and as reported in [14] separately. We then perform PCA on the spectral pattern vectors from the two kinds of Laplacian and embed them into a 3-dimensional space. We locate the clusters by using the K -means method and calculate the Rand index to evaluate the clustering performance. The Rand indices, about the two methods, of the clustered objects with different numbers of classes are illustrated in Figure 6. From the point of view by using Rand index for evaluation, the proposed hypergraph Laplacian is superior to Zhou’s hypergraph Laplacian

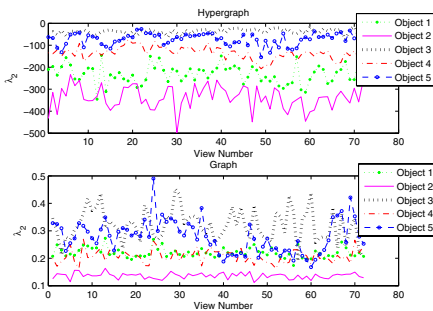


Fig. 6. Eigenvalue λ_2

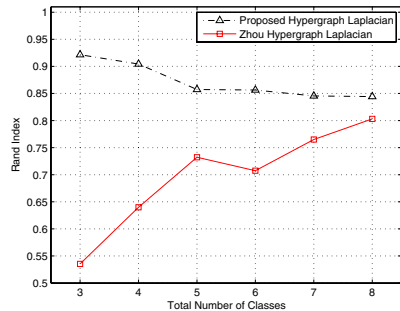


Fig. 7. Rand Index

in clustering visual information. This indicates that although Zhou's hypergraph Laplacian performs well in classifying categorical data, it is not suitable for visual clustering problems.

7 Conclusion

In this paper we have investigated how spectral pattern vectors extracted from the hypergraph Laplacian can be embedded in low-dimensional spaces for the purpose of clustering. There are two main contributions in this paper. The first is that we develop a hypergraph representation based on the locations of attributed feature points, i.e. feature hypergraph, to represent objects. The second is that we introduce an improved hypergraph Laplacian and develop one embedding method by applying PCA to the pattern vectors generated from it. The hypergraphs studied in our analysis are of different sizes and we do not need to locate correspondences. Experimental results indicate that the proposed scheme can provide useful pattern vectors for objects which perform well the in the domain of object view clustering.

References

1. IEEE TPAMI: Special Section on Graph Algorithms and Computer Vision, 23 10401151.1 (2001)
2. Agarwal, S., Branson, K., Belongie, S.: Higher-Order Learning with Graphs. In: ICML, pp. 17–23 (2006)
3. Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., Belongie, S.: Beyond pairwise clustering. CVPR 2, 838–845 (2005)
4. Bretto, A., Cherifi, H., Aboutajdine, D.: Hypergraph imaging: an overview. Pattern Recognition 35(3), 651–658 (2002)
5. Chung, F.K.: The Laplacian of a Hypergraph. AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science 10, 21–36 (1993)
6. Gibson, D., Kleinberg, J., Raghavan, P.: Clustering categorical data: An approach based on dynamical systems. The VLDB Journal 8(3-4), 222–236 (2000)
7. Hagen, L., Kahng, A.B.: New spectral methods for ratio cut partitioning and clustering. IEEE TCAD 11(9), 1074–1085 (1992)
8. Harris, C.G., Stephens, M.J.: A combined corner and edge detector. In: Proceedings of Fourth Alvey Vision Conference, pp. 147–151 (1994)
9. Li, W., Sole, P.: Spectra of Regular Graphs and Hypergraphs and Orthogonal Polynomials. European Journal of Combinatorics 17, 461–477 (1996)
10. Luo, B., Wilson, R.C., Hancock, E.R.: Higher-Order Learning with Graphs. Pattern Recognition 36(10), 2213–2223 (2003)
11. Rodriguez, J.A.: On The Laplacian Eigenvalues and Metric Parameters of Hypergraphs. Linear and Multilinear Algebra 51, 285–297 (2003)
12. Storm, C.K.: The Zeta Function of a Hypergraph. EJC 13 (2006)
13. Wilson, R.C., Hancock, E.R., Luo, B.: Pattern Vectors from Algebraic Graph Theory. IEEE TPAMI 27(7), 1112–1124 (2005)
14. Zhou, D., Huang, J., Scholkopf, B.: Learning with Hypergraphs: Clustering, Classification, and Embedding. NIPS 19, 17–24 (2007)
15. Zhu, P., Wilson, R.C.: A Study of Graph Spectra for Comparing Graphs and Trees. Pattern Recognition (to appear, 2008)