

Ihara Coefficients: A Flexible Tool for Higher Order Learning

Peng Ren¹, Tatjana Aleksić², Richard C. Wilson¹, and Edwin R. Hancock¹

¹ Department of Computer Science, The University of York,
York, YO10 5DD, UK

{pengren, wilson, erh}@cs.york.ac.uk

² University of Kragujevac, Faculty of Science, 34000 Kragujevac, Serbia
taleksic@kg.ac.rs

Abstract. The aim of this paper is to seek a compact characterization of irregular unweighted hypergraphs for the purposes of clustering. To this end, we propose a novel hypergraph characterization method by using the Ihara coefficients, i.e. the characteristic polynomial coefficients extracted from the Ihara zeta function. We investigate the flexibility of the Ihara coefficients for learning relational structures with different relational orders. Furthermore, we introduce an efficient method for computing the coefficients. Our representation for hypergraphs takes into account not only the vertex connections but also the hyperedge cardinalities, and thus can distinguish different relational orders, which is prone to ambiguity in the hypergraph Laplacian. In experiments we demonstrate the effectiveness of the proposed characterization for clustering irregular unweighted hypergraphs and its advantages over the spectral characterization of the hypergraph Laplacian.

1 Introduction

Hypergraph-based methods have recently been widely used for representing and processing relational structures where the relations present are not simply pairwise. Specific applications of hypergraph related methods in visual processing include the algorithms described in [3][6]. One common feature of these methods is that they exploit domain specific and goal directed representations, and do not lend themselves to generalization. The reason for this lies in the difficulty in formulating a hypergraph in a mathematically uniform way for computation. However, to be easily manipulated, hypergraphs must be represented in a mathematically consistent way, using structures such as matrices or vectors. One possible method for establishing hypergraph matrix representations is to transform a hypergraph into a graph and then use the associated graph adjacency matrix or Laplacian matrix as the matrix representation of the hypergraph. Agarwal *et al.* [1] have made a review of the possible graph representations for a hypergraph and revealed their relationships with each other in machine learning. Each of these methods assume that there is a weight attached to each hyperedge. The edges in the graph representation are weighted in a manner determined by the corresponding hyperedge weights. As far as unweighted hypergraphs are concerned, the literature mainly focuses on using tensor representations [7][9]. The tensor representations consider all possible permutations of a subset of vertices and establish hyperedges with cardinality

consistent with the relational order. Therefore, tensors can only represent regular hypergraphs, and are not suited for irregular hypergraphs. Ren *et al.* [5] have proposed an improved hypergraph Laplacian based on developments of Zhou *et al.*'s method [10] and apply it to clustering hypergraphs. Although this method is suitable for unweighted irregular hypergraphs, it is based on a relatively impoverished spectral characterization and overlooks much of the detail of hypergraph-structure. Recently, Ren *et al.* [4] have attempted to represent hypergraphs using characteristics from the Ihara zeta function. However, this work does not indicate in which cases the characteristics are superior to spectral methods, neither does it investigate the flexibility of these features.

In this paper, we characterize irregular unweighted hypergraphs using Ihara coefficients. The proposed hypergraph representation proves to be a flexible tool in learning the structure of irregular unweighted hypergraphs with different relational orders. Our contributions are two-fold. First, we propose a vectorial representation, which naturally avoids the ambiguity induced by the matrix representations such as the hypergraph Laplacian, for irregular unweighted hypergraphs. We construct pattern vectors using the Ihara coefficients, i.e. the characteristic polynomial coefficients extracted from Ihara zeta function for hypergraphs. Second and more importantly, we propose an efficient method for computing the Ihara coefficient set, which renders the computation of the coefficients tractable. We use the pattern vectors consisting of Ihara coefficients for clustering hypergraphs extracted from images of different object views and demonstrate their effectiveness in hypergraph characterization.

2 Hypergraph Laplacian

A hypergraph is a generalization of a graph. Unlike the edge of a graph, which can connect only two vertices, the hyperedge in a hypergraph can connect any number of vertices. A hypergraph is normally defined as a pair $H(V, E_H)$ where V is a set of elements, called nodes or vertices, and E_H is a set of non-empty subsets of V called hyperedges. 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 $H(V, E_H)$ with I vertices and J hyperedges, we establish an $I \times J$ matrix \mathbf{H} which is referred to as the incidence matrix of the hypergraph. \mathbf{H} has element $h_{i,j}$ 1 if $v_i \in e_j$ and 0 otherwise.

The incidence matrix can be more easily manipulated than its equivalent set representation. To obtain a vertex-to-vertex representation, we need to establish the adjacency matrix and Laplacian matrix for a hypergraph. To this end, a graph representation for the hypergraph is required. Agarwal *et al.* [1] have classified the graph representations for a hypergraph into two categories, namely a) the clique expansion and b) the star expansion. The clique expansion represents a hypergraph by constructing a graph with all the pairs of vertices within a hyperedge connecting each other. The star expansion represents a hypergraph by introducing a new vertex to every hyperedge and constructing a graph with all vertices within a hyperedge connecting the newly introduced vertex. The common feature of these methods is that each edge in a graph representation is weighted in terms of the corresponding hyperedge weight subject to certain conditions.

For example, the normalized Laplacian matrix $\hat{L}_H = I - D_v^{-1/2}HD_eH^TD_v^{-1/2}$ introduced in [10] is obtained from the star expansion of a hypergraph, and its individual edges are weighted by the quotient of the corresponding hyperedge weight and cardinality. Here D_v is the diagonal vertex degree matrix whose diagonal element $d(v_i)$ is the summation of the i th row of H , D_e is the diagonal vertex degree matrix whose diagonal element $d(e_j)$ is the summation of the j th column of H , and I is a $|V| \times |V|$ identity matrix. In this case, even edges derived from an unweighted hyperedge are assigned a nonunit weight. On the other hand, rather than attaching a weight to each edge in the graph representation, the adjacency matrix and the associated Laplacian matrix for an irregular unweighted hypergraph can be defined as $A_H = HH^T - D_v$ and $L_H = D_v - A_H = 2D_v - HH^T$ respectively [5]. In practice, these two definitions are obtained in terms of the clique expansion without attaching a weight to a graph edge. The eigenvalues of L_H are referred to as the hypergraph Laplacian spectrum and can be used in a straightforward way as hypergraph characteristics.

Although the vertex-to-vertex matrix representations for hypergraphs described above naturally reduce to those for graphs when the relational order is two, there are deficiencies for these representations in distinguishing relational structures. When relational structures have the same vertex cardinality but different relational orders, these vertex-to-vertex matrix representations become ambiguous. For example, for the graph in Fig. 1(a) and the hypergraph in Fig. 1(b), the adjacency matrices of the two hypergraphs are identical, and so are the associated Laplacian matrices. The adjacency matrix and Laplacian matrix are as follows:

$$A_H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \quad L_H = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}$$

It is clear that the unweighted adjacency matrix and Laplacian matrix can not distinguish these two hypergraphs. The reason for this deficiency is that the adjacency matrix and the Laplacian matrix only record the adjacency relationships between pairs of nodes and neglect the cardinalities of the hyperedges. In this regard they induce certain information loss in representing relational structures and can not always distinguish between pairwise relationships and high order relationships for the same set of vertices. The normalized Laplacian matrix for Fig. 1(a) and 1(b) are \hat{L}_{H1} and \hat{L}_{H2} respectively.

$$\hat{L}_{H1} = \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix} \quad \hat{L}_{H2} = \begin{pmatrix} 1/2 & -1/4 & -1/4 \\ -1/4 & 1/2 & -1/4 \\ -1/4 & -1/4 & 1/2 \end{pmatrix}$$

Since $\hat{L}_{H2} = \frac{3}{4}\hat{L}_{H1}$, the eigenvalues of \hat{L}_{H2} are found by scaling those of \hat{L}_{H1} by a factor $3/4$, and both matrices have the same eigenvectors. Thus the normalized Laplacian matrices for different hypergraphs may yield spectra that are just scaled relative to each other. This hinders the hypergraph characterization when the eigenvectors are used. One important reason for the limited usefulness of the above hypergraph matrix representations is that they result in information loss when relational orders of varying degree are present. To overcome this deficiency, we use characteristic polynomials extracted from the Ihara zeta function as a means of representing hypergraphs. In the next

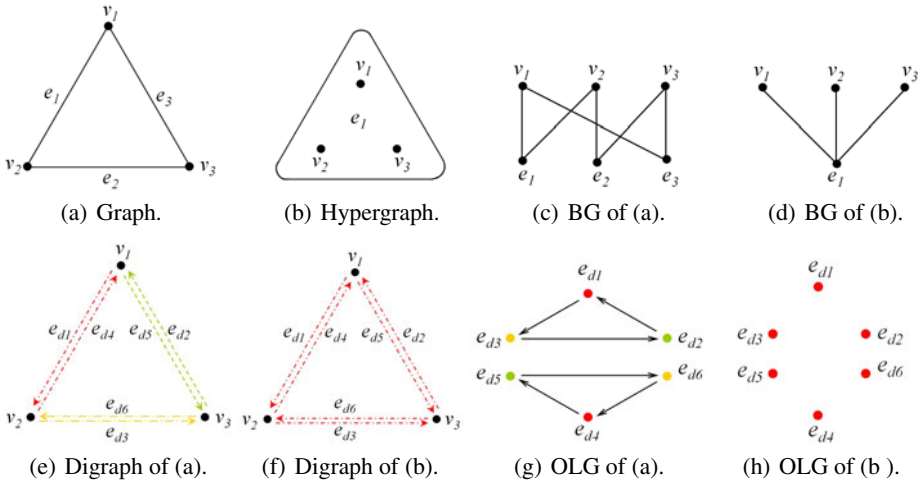


Fig. 1. Hypergraph examples and their graph representations

section, we commence by showing that the Ihara zeta function can be used to represent this type of relational structure in hypergraphs. We use the Ihara coefficients, i.e. the characteristic polynomial coefficients extracted from the Ihara zeta function, as hypergraph characteristics. We show that the Ihara coefficients not only encode the relational structural in a consistent way but also overcome the deficiencies listed above.

3 Ihara Zeta Function from Graphs to Hypergraphs

The rational expression of the Ihara zeta function for a graph is as follows [2]:

$$Z_G(u) = (1 - u^2)^{\chi(G)} \det (\mathbf{I}_{|V(G)|} - u\mathbf{A} + u^2\mathbf{Q})^{-1}, \tag{1}$$

where $\chi(G) = |V| - |E|$, \mathbf{A} is the adjacency matrix of the graph, and $\mathbf{Q} = \mathbf{D} - \mathbf{I}_{|V(G)|}$ where $\mathbf{I}_{|V(G)|}$ is the identity matrix and \mathbf{D} is the degree matrix, which can be generated by placing the column sums as the diagonal elements while setting the off-diagonal elements to zero.

To formulate the Ihara zeta function for a hypergraph in a similar form with (1), the bipartite graph representation of the hypergraph is needed. To this end, we use a dual representation in which each hyperedge is represented by a new vertex. The new vertex is incident to each of the original vertices in the corresponding hyperedge. The union of the new vertex set and the original vertex set constitute the vertex set of the associated bipartite graph. The new vertices corresponding to hyperedges are on one side and the original hypergraph vertices on the other side. Thus the bipartite graph and star expansion for a hypergraph share the same form, although they are defined for different purposes. For instance, the bipartite graphs associated with the example hypergraphs in Figs. 1(a) and 1(b) are shown in Figs. 1(c) and 1(d) respectively (BG stands for bipartite graph).

The Ihara zeta function of the hypergraph $H(V, E_H)$ can be expressed in a rational form as follows:

$$\zeta_H(u) = (1 - u)^{\chi(BG)} \det (\mathbf{I}_{|V(H)|+|E_H(H)|} - \sqrt{u}\mathbf{A}_{BG} + u\mathbf{Q}_{BG})^{-1}, \quad (2)$$

where $\chi(BG)$ is the Euler number of the associated bipartite graph, \mathbf{A}_{BG} is the adjacency matrix of the associated bipartite graph, and $\mathbf{Q}_{BG} = \mathbf{D}_{BG} - \mathbf{I}_{|V(H)|+|E_H(H)|}$. Further details on the arguments leading from (1) to (2) can be found in [8].

The adjacency matrix of the associated bipartite graph can be formulated using the incidence matrix \mathbf{H} of $H(V, E_H)$:

$$\mathbf{A}_{BG} = \begin{bmatrix} \mathbf{0}_{|E_H(H)| \times |E_H(H)|} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0}_{|V(H)| \times |V(H)|} \end{bmatrix}. \quad (3)$$

The hypergraph Ihara zeta function in the form of (2) provides an alternative method for the function value computation, as well as an efficient method of computing the Ihara coefficients, which will be discussed later on in Section 5.

4 Determinant Expression for Hypergraph Zeta Function

Although the Ihara zeta function can be evaluated efficiently using (2), the task of enumerating the coefficients of the polynomial appearing in the denominator of the Ihara zeta function is difficult, except by resorting to software for symbolic calculation. To efficiently compute these coefficients, a different strategy is adopted. The hypergraph is first transformed into an oriented line graph. The Ihara zeta function is then the reciprocal of the characteristic polynomial for the adjacency matrix of the oriented line graph. Our novel contribution here is to use the existing ideas from hypergraph theory to develop a new hypergraph representation, which can be used in machine learning to distinguishing hypergraphs with the same vertex set but different relational orders.

4.1 Oriented Line Graph

To establish the oriented line graph associated with the hypergraph $H(V, E_H)$, we commence by constructing a $|e_i|$ -clique, i.e. clique expansion, by connecting each pair of vertices in the hyperedge $e_i \in E_H$ through an edge. The resulting clique expansion graph is denoted by $GH(V, E_G)$. For $GH(V, E_G)$, the associated symmetric digraph $DGH(V, E_d)$ can be obtained by replacing each edge of $GH(V, E_G)$ by an arc (oriented edge) pair in which the two arcs are inverse to each other. For the example hypergraphs in Figs. 1(a) and 1(b), their $DGH(V, E_d)$ are shown in Figs. 1(e) and 1(f) respectively, where the oriented edges derived from the same hyperedge are colored the same while from different hyperedges are colored differently. Finally, the oriented line graph of the hypergraph can be established based on the symmetric digraph. The vertex set V_{ol} and edge set E_{ol} of the the oriented line graph are defined as follows [8]:

$$V_{ol} = E_d(DGH); \quad E_{ol} = \{(e_d(u, v), e_d(v, w)) \in E_d \times E_d; u, w \notin E_H\}. \quad (4)$$

One observation that needs to be made here is that the adjacency matrix A_H and Laplacian matrix L_H for a hypergraph introduced in Section 2 are actually those of the graph established on the clique expansion, but without an edge-weight attachment. These matrix representations can induce ambiguity when representing relational structures with different relational orders. This point is illustrated by the two example hypergraphs in Figs. 1(a) and 1(b) which have the same clique graph and thus the same adjacency matrix and Laplacian matrix. The reason for this is that the clique expansion only records adjacency relationships between pairs of nodes and can not distinguish whether or not two edges in the clique are derived from the same hyperedge. Thus the clique graph representations for hypergraph result in loss of information concerning relational order. However, the Ihara zeta function overcomes this deficiency by avoiding the interaction between two edges derived from the same hyperedge. This is due to the constraint in (4) that the connecting oriented edge pair in the same clique of DGH can not establish an oriented edge in the oriented line graph. According to these properties, the example hypergraphs with the same adjacency matrix and Laplacian matrix in Figs. 1(a) and 1(b) produce oriented line graphs with totally different structures as shown in Figs. 1(g) and 1(h) respectively (OLG stands for oriented line graph), where the constraint in (4) prevents connections between any nodes with the same color in Figs. 1(g) and 1(h). The adjacency matrix T_H of the oriented line graph is the Perron-Frobenius operator of the original hypergraph. For the (i, j) th entry of T_H , $T_H(i, j)$ is 1 if there is one edge directed from the vertex with label i to the vertex with label j in the oriented line graph, otherwise it is 0. Unlike the adjacency matrix of an undirected graph, the Perron-Frobenius operator for a hypergraph is not a symmetric matrix. This is because of the constraint described above that arises in the construction of oriented edges. Specifically, it is the fact that the arc pair with two arcs that are derived from the same hyperedge in the original hypergraph is not allowed to establish an oriented edge in the oriented line graph that causes the asymmetry of T_H .

4.2 Characteristic Polynomial

With the oriented line graph to hand, the Ihara zeta function for a hypergraph can be written in the form of a determinant using the Perron-Frobenius operator [8]:

$$\zeta_H(u) = \det(\mathbf{I}_H - u\mathbf{T}_H)^{-1} = (c_0 + c_1u + \cdots + c_{M-1}u^{M-1} + c_Mu^M)^{-1}, \quad (5)$$

where M is the highest order of the polynomial. The polynomial coefficients c_0, c_2, \dots, c_M are referred to as the Ihara coefficients. From (5) we can see that M is the dimensionality of the square matrix \mathbf{T}_H . To establish pattern vectors from the hypergraph Ihara zeta function for the purposes of characterizing hypergraphs in machine learning, it is natural to consider taking function samples as the elements. Although the function values at most of the sampling points will perform well in distinguishing hypergraphs, there is the possibility of sampling at poles giving rise to meaningless infinities. Hence, the pattern vectors consisting of function samples are potentially unstable representations of hypergraphs, since the distribution of poles is unknown beforehand. The characteristic polynomial coefficients, i.e. the Ihara coefficients, do not give rise to infinities. From (5), it is clear that each coefficient can be derived from the

elementary symmetric polynomials of the eigenvalue set $\{\lambda_1, \lambda_2, \lambda_3 \dots\}$ of T_H as $c_r = (-1)^r \sum_{k_1 < k_2 < \dots < k_r} \lambda_{k_1} \lambda_{k_2} \dots \lambda_{k_r}$.

Furthermore, the Ihara coefficients relate strongly to the hypergraph-structure since the Ihara zeta function records information about prime cycles in the hypergraphs. We can construct pattern vectors using a dominant subset of the Ihara coefficients $v = [c_{r1} \ c_{r2} \ \dots \ c_{rN}]^T$ for a hypergraph and then apply them to clustering hypergraphs.

5 Numerical Computation

The formation of T_H and its eigen-decomposition tend to be computationally expensive for practical problems, because the matrix T_H are usually of big size. To overcome the deficiency of computing the Ihara coefficients using (5), we develop a straightforward yet efficient method which starts from the associated bipartite graph. Instead of constructing the oriented line graph for a hypergraph, we establish the oriented line graph for the bipartite graph. Considering the rational expression (2) based on the associated bipartite graph, we have:

$$\zeta_H^{-1}(u) = Z_{BG}^{-1}(\sqrt{u}) = \det(\mathbf{I}_{BG} - \sqrt{u}\mathbf{T}_{BG}), \tag{6}$$

where T_{BG} is the Perron-Frobenius operator of the associated bipartite graph, of which the Ihara zeta function (according to its original definition [2]) is represented as:

$$Z_{BG}^{-1}(u) = \prod_{p \in P_{BG}} (1 - u^{|p|}) = (1 - u^{|p_1|}) (1 - u^{|p_2|}) (1 - u^{|p_3|}) \dots \tag{7}$$

where p_i is the i th prime cycle in the set P_{BG} of prime cycle equivalence classes of the bipartite graph. Note that every cycle in a bipartite graph has an even length, i.e. $|p_i|$ is always an even number for a bipartite graph. Let $\{\tilde{c}_0, \tilde{c}_1, \tilde{c}_2, \tilde{c}_3, \tilde{c}_4, \tilde{c}_5, \tilde{c}_6 \dots\}$ denote the Ihara coefficient set of the bipartite graph. It is clear that $Z_{BG}^{-1}(u)$ is a polynomial with the odd coefficients equal to zeros:

$$\begin{aligned} Z_{BG}^{-1}(u) &= \det(\mathbf{I}_{BG} - u\mathbf{T}_{BG}) = \tilde{c}_0 + \tilde{c}_1u + \tilde{c}_2u^2 + \tilde{c}_3u^3 + \tilde{c}_4u^4 + \tilde{c}_5u^5 + \tilde{c}_6u^6 + \dots \\ &= \tilde{c}_0 + \tilde{c}_2u^2 + \tilde{c}_4u^4 + \tilde{c}_6u^6 + \dots \end{aligned} \tag{8}$$

Taking \sqrt{u} as the argument of the bipartite graph Ihara zeta function instead of u :

$$\begin{aligned} \zeta_H^{-1}(u) &= Z_{BG}^{-1}(\sqrt{u}) = \det(\mathbf{I}_{BG} - \sqrt{u}\mathbf{T}_{BG}) = (1 - (\sqrt{u})^{|p_1|}) (1 - (\sqrt{u})^{|p_2|}) \dots \\ &= \tilde{c}_0 + 0\sqrt{u} + \tilde{c}_2(\sqrt{u})^2 + 0(\sqrt{u})^3 + \tilde{c}_4(\sqrt{u})^4 + 0(\sqrt{u})^5 + \tilde{c}_6(\sqrt{u})^6 + \dots \\ &= \tilde{c}_0 + \tilde{c}_2u + \tilde{c}_4u^2 + \tilde{c}_6u^3 + \dots = c_0 + c_1u + c_2u^2 + c_3u^3 + \dots \end{aligned} \tag{9}$$

As we can see in (9), the Ihara coefficients of a hypergraph can be efficiently obtained by selecting just the even-indexed Ihara coefficients of the associated bipartite graph. This is much more efficient than the computation based on the oriented line graph of the hypergraph, because T_{BG} is much smaller in size than T_H , especially for large hypergraphs. The size of the Perron-Frobenius operator of an irregular hypergraph tends to be

difficult to enumerate. Here we thus use the K -regular hypergraph, i.e. hypergraph with every hyperedge containing K vertices, for analyzing the computational complexity of the Perron-Frobenius operators \mathbf{T}_H and \mathbf{T}_{BG} . Suppose there are in total N hyperedges in the K -regular hypergraph. To obtain \mathbf{T}_H , the clique expansion and its digraph of the K -regular hypergraph need to be established according to the transform introduced in Section 4.1. This procedure produces an oriented line graph with $K(K-1)N$ vertices and a Perron-Frobenius operator of size $(K-1)KN \times (K-1)KN$. To obtain \mathbf{T}_{BG} , the bipartite graph and its digraph of the K -regular hypergraph need to be established. This procedure produces an oriented line graph with $2KN$ vertices and a Perron-Frobenius operator of size $2KN \times 2KN$. For regular hypergraphs K is not less than 2, and the relation always holds for $2KN < (K-1)KN$. As a result, the size of \mathbf{T}_{BG} is smaller than that of \mathbf{T}_H . The computational complexity of obtaining the Ihara coefficients is governed by the eigen-decomposition of the Perron-Frobenius operator. This requires $O(n^3)$ operations where n is the size of the Perron-Frobenius operator. Therefore, the computational overheads of eigen-decomposition on \mathbf{T}_{BG} are lower than those of \mathbf{T}_H .

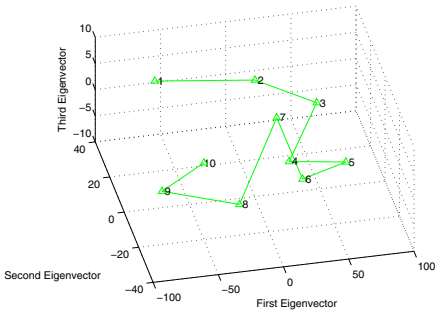
6 Experimental Evaluation

To establish hypergraphs on the visual objects, we first extract feature points using the Harris detector as the vertices of hypergraphs. Let $\mathbf{c}(v_i)$ denote the spatial coordinate of the feature point v_i in an image, and $I(v_i)$ denote the intensity of v_i . For each image, we construct the hypergraph using the method introduced in [5], where the element $H(i, j)$ of incidence matrix is 1 if $\|\mathbf{c}(v_i) - \mathbf{c}(v_j)\| \leq Th_{j1}$ and $|I(v_i) - I(v_j)| \leq Th_{j2}$, and 0 otherwise. Here Th_{j1} is the neighborhood threshold set to 1/4 the size of the image and Th_{j2} is the similarity threshold determined by the standard deviation of the intensities of neighboring feature points.

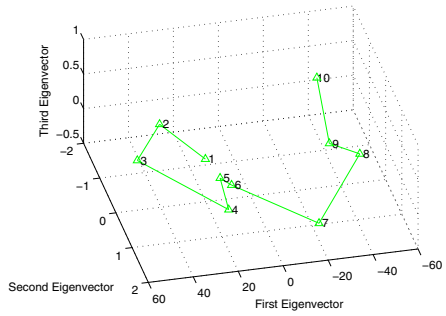
We first test the Ihara coefficient pattern vector in the form of $v_H = [c_3, c_4, \ln(|c_{M-3}|), \ln(|c_{M-2}|), \ln(|c_{M-1}|), \ln(|c_M|)]^T$ in characterizing within-class hypergraphs. We establish hypergraphs on ten images of a model house in the Chalet data set [5]. The images are taken consecutively as the camera pans around the model house in regular angular increments. Fig. 2 shows the PCA projections of the hypergraphs based on the truncated Laplacian spectrum, i.e. the leading six nonzero Laplacian eigenvalues, and the Ihara coefficients. The Laplacian spectra produce an erratic trajectory. The Ihara coefficients produce a much smoother trajectory and the neighboring images in the sequence are generally Euclidean neighbors in the eigenspace.

We then illustrate the largest Laplacian eigenvalue and the final Ihara coefficient for hypergraphs extracted from four objects in the COIL dataset [5]. The Ihara coefficients give clearer class separability than the Laplacian eigenvalues.

Finally we test the Ihara coefficients for clustering both unweighted graphs and unweighted hypergraphs. The graphs and hypergraphs are extracted from the images in the COIL dataset. We establish a Delaunay graph on the feature points of each image, and construct the pattern vectors in the form of $v_{Gs} = [c_3, c_4, \ln(|c_{2M}|)]^T$ for graphs. We evaluate the clustering performance obtained with different numbers of object classes. After performing PCA on the pattern vectors both for graphs and hypergraphs, we locate the clusters using the K -means method and calculate the Rand index, which is

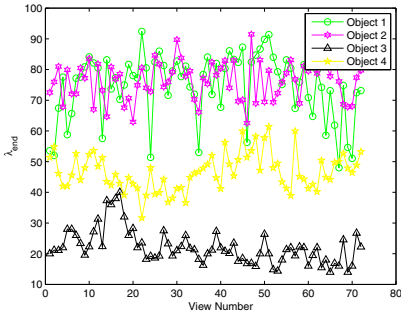


(a) Truncated Laplacian spectra.

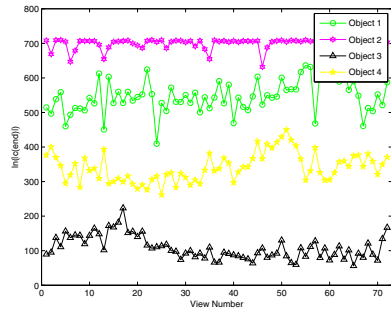


(b) Ihara coefficients.

Fig. 2. Within-class trajectory

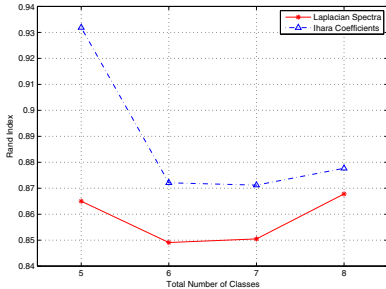


(a) Leading nonzero Laplacian eigenvalue.

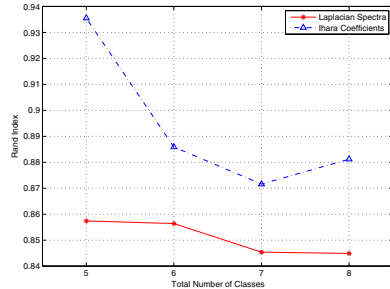


(b) Ihara coefficients.

Fig. 3. Ihara coefficient plot



(a) Graph.



(b) Hypergraph.

Fig. 4. Rand index

plotted as a function of class number in Fig. 4. We use Laplacian spectra for graphs and hypergraphs for comparison. From this set of experiments it is clarified that for both graphs and hypergraphs, the Ihara coefficients outperform the Laplacian spectra.

7 Conclusion

We have pointed out the deficiency of the vertex-to-vertex matrix representations for learning hypergraph-structure and applied the Ihara coefficients to hypergraph characterization to overcome these problems. The Ihara coefficients are a flexible tool which can be computed in a consistent manner for both graphs and hypergraphs. They can effectively overcome the ambiguity in distinguishing high order relational structures when matrix representations fail to work. Furthermore, we have proposed an efficient method for computing the Ihara coefficient set. Experimental results show that the Ihara coefficients are superior to spectral methods, both for graphs and hypergraphs.

Acknowledgments

We acknowledge the financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). Tatjana Aleksić is supported by Grant 144015G of the Serbian Ministry for Science and the British Scholarship Trust. Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

References

1. Agarwal, S., Branson, K., Belongie, S.: Higher-order learning with graphs. In: ICML (2006)
2. Bass, H.: The ihara-selberg zeta function of a tree lattice. *International Journal of Mathematics* 6, 717–797 (1992)
3. Bretto, A., Cherifi, H., Aboutajdine, D.: Hypergraph imaging: an overview. *Pattern Recognition* 35(3), 651–658 (2002)
4. Ren, P., Aleksić, T., Wilson, R.C., Hancock, E.R.: Hypergraphs, characteristic polynomials and the ihara zeta function. In: CAIP (2009)
5. Ren, P., Wilson, R.C., Hancock, E.R.: Spectral embedding of feature hypergraphs. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 308–317. Springer, Heidelberg (2008)
6. Rota-Bullo, S., Albarelli, A., Pelillo, M., Torsello, A.: A hypergraph-based approach to affine parameters estimation. In: ICPR (2008)
7. Shashua, A., Zass, R., Hazan, T.: Multi-way clustering using super-symmetric non-negative tensor factorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 595–608. Springer, Heidelberg (2006)
8. Storm, C.K.: The zeta function of a hypergraph. *Electronic Journal of Combinatorics* 13 (2006)
9. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: CVPR (2008)
10. Zhou, D., Huang, J., Scholkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: NIPS (2007)