Weighted Graph Characteristics from Oriented Line Graph Polynomials

Peng Ren Richard C. Wilson Edwin R. Hancock
Department of Computer Science
University of York, York, YO10 5DD, UK
{pengren, wilson, erh}@cs.york.ac.uk

Abstract

We develop a novel method for extracting graph characteristics from edge-weighted graphs, based on an extension of the Ihara zeta function from unweighted to edge-weighted graphs. This is effected by generalizing the determinant form of the Ihara zeta function. We use the set of the reciprocal polynomial coefficients of the resulting Ihara zeta function, i.e. the Ihara coefficients, to construct our characterization. We also present a spectral analysis of the edge-weighted graph Ihara coefficients and indicate their advantages over graph spectral methods. Experimental results reveal that the Ihara coefficients are effective for the purpose of clustering edge-weighted graphs.

1. Introduction

Graph-based methods were one of the earliest paradigms proposed for scene understanding [10][2], and have provided powerful tools for the analysis of shape, object and scene structures in computer vision. Examples include the analysis of shock-graphs [19], articulated shape representation [14] and object tracking [20]. Recently, the appearance based methods based on the arrangement of SIFT features have provided impressive practical results for object recognition and video analysis [15]. However, here too, the need for structural representations has been demonstrated with the SIFT features being augmented by constellations [9] and k-fans [7] to represent the arrangement of feature points.

When dealing with graph-based representations of image structures there are two ways in which the analysis of structure can be conducted. The first of these is to perform a detailed correspondence analysis and to seek matches between the nodes of the structures under consideration. There are many examples of node correspondence algorithms in computer vision, especially for feature point matching [6][13][23]. These methods exploit local connectivity information together with node and edge attributes to match relational structures by optimizing a criterion function. However, correspondence matching can become intractable when the sizes of the structures become large and when feature points tend to be less discriminative. It is for these reasons that a second approach based on comparing permutation invariant characteristics extracted from graphs is adopted as an alternative.

There are a number of alternative node permutation invariant characterizations that can be used. Perhaps the simplest of these are topological properties such as the size, edge density, the perimeter or diameter and the number of cycles of different degrees. A more sophisticated alternative is to use features extracted from a matrix characterization of a graph. Here the initial matrix representation $\boldsymbol{M}$ can be based either on the adjacency matrix, the Laplacian matrix or the signless Laplacian. The matrix can be characterized using either its eigenvalues $\mathrm{sp}(\boldsymbol{M})$ and eigenvectors (i.e. using spectral graph theory) or by the coefficients of its characteristic polynomial $\det(\lambda \boldsymbol{I} - \boldsymbol{M})$ (i.e. using algebraic graph theory). The two approaches are essentially equivalent and have led to a number of practical characterizations that can be used for both object recognition and shape clustering [21][13][22].

An alternative characterization that has received relatively little attention in the computer vision and pattern recognition community is provided by the zeta function. In number theory, the Riemann zeta function is determined by the locations of the prime numbers. There is a natural extension of the Riemann zeta function from prime numbers to graphs. For instance, the Ihara zeta function is determined by the set of prime cycles on a graph, and is detailed in [11]. Bass [4] has generalized the explicit factorizations to all finite graphs. It is interesting to note that the Ihara zeta function is computed by first transforming the graph in-hand into an oriented line graph, and then computing the characteristic polynomial of the adjacency matrix of the oriented line graph. The zeta function is determined by the reciprocal of the characteristic polynomial, and the prime cycles determine the poles of the zeta function in a manner analogous to the prime numbers. There have been a number of recent applications of zeta functions in computer vision and pattern recognition. For instance, Ren *et al.* [17]

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

have shown how to use the coefficients of the characteristic polynomial of the oriented line graph to cluster unweighted graphs. Zhao *et al.* [24] have used Savchenko's formulation of the zeta function, expressed in terms of cycles, to generate merge weights for clustering over a graph-based representation of pairwise similarity data [24]. Bai *et al.* [1] have shown that the Riemann zeta function is the moment generating function of the heat-kernel trace and have used the moments to cluster graphs.

Although the the zeta function draws on the characteristic polynomial of a graph and is hence akin to methods from algebraic graph theory, it first relies on a graph transformation. This is an interesting observation since the quest for improved alternatives to the adjacency and Laplacian matrices has been a quest in spectral graph theory. Recently, the signless Laplacian (i.e. the degree matrix plus the adjacency matrix) has been suggested. Additionally, Emms *et al.* [8] have recently shown that a unitary matrix characterization of the oriented line graph can be used to reduce or even completely lift the cospectrality of certain classes of graphs, including trees and strongly regular graphs. This points to the fact that one potentially profitable route to improving methods from spectral graph theory may reside in graph transformation.

Unfortunately, the existing definition of the Ihara zeta function applies only to unweighted graphs. The task of utilizing the Ihara zeta function as a characterization of edge-weighted graphs has yet to be investigated. The determinant (i.e. characteristic polynomial) expression of the Ihara zeta function is not available as a characterization of edge-weighted graphs due to its binary representation. In this paper, we address this shortcoming and derive a determinant expression applicable to edge-weighted graphs. This is effected with the assistance of Bartholdi zeta function. The Ihara coefficients for edge-weighted graphs are computed and the resulting pattern vectors are used to cluster edge-weighted graphs extracted from visual objects.

2. The Ihara Zeta Function

The Ihara zeta function of unweighted graphs is a generalization of the Riemann zeta function from number theory. In the definition of the Ihara zeta function, the 'prime number' in the Euler product expansion of the Riemann zeta function is replaced by a 'prime cycle', i.e. cycles with no backtracking in a graph. As a result, the Ihara zeta function is generally an infinite product. However, one of its elegant features is that it can be collapsed down into a rational function, which renders it of practical utility.

2.1. Rational Expression

For a graph $G(V, E)$ with the vertex set V of cardinality $|V| = N$ and the edge set E of cardinality $|E| = M$, the rational expression of the Ihara zeta function is [11]:

$$Z_G(u) = \left(1 - u^2\right)^{\chi(G)} \det\left(\boldsymbol{I}_N - u\boldsymbol{A} + u^2\boldsymbol{Q}\right)^{-1}. \quad (1)$$

Here, $\chi(G)$ is the Euler number of the graph $G(V, E)$, which is defined as the difference between cardinalities of the vertex set and the edge set of the graph, i.e. $\chi(G) = N - M$, and $\boldsymbol{A}$ is the adjacency matrix of the graph. The degree matrix $\boldsymbol{D}$ is constructed by placing the column sums of the adjacency matrix as diagonal elements, while setting the off-diagonal elements to zero. Finally, with $\boldsymbol{I}_k$ denoting the $k \times k$ identity matrix, $\boldsymbol{Q}$ is the matrix difference of the degree matrix $\boldsymbol{D}$ and the identity matrix $\boldsymbol{I}_N$, i.e. $\boldsymbol{Q} = \boldsymbol{D} - \boldsymbol{I}_N$. From (1) it has been shown that the Ihara zeta function is permutation invariant to vertex label permutations [17]. This is because permutation matrices, which represent vertex label permutations in matrix calculation, have no effect on the determinant in (1).

2.2. Determinant Expression

For md2 graphs, i.e. the graphs with vertex degree at least 2, it is straightforward to show that (1) can be rewritten in the form of the reciprocal of a polynomial. However, it is difficult to compute the coefficients of the reciprocal of the Ihara zeta function from (1) in a uniform way, except by resorting to software for symbolic calculation. To efficiently compute these coefficients, it is more convenient to transform the rational form of the Ihara zeta function in (1) into a concise expression. The Ihara zeta function can also be written in the form of a determinant [12]:

$$Z_G(u) = \det(\boldsymbol{I}_{2M} - u\boldsymbol{T})^{-1} \quad (2)$$

where $\boldsymbol{T}$ is the Perron-Frobenius operator of the oriented line graph, and is an $2M \times 2M$ square matrix.

To obtain the Perron-Frobenius operator $\boldsymbol{T}$, we must construct the oriented line graph of the original graph from the associated symmetric digraph. The symmetric digraph $SDG(V, E_d)$ of a graph $G(V, E)$ is composed of a finite nonempty vertex set V identical to that of $G(V, E)$ and a finite multiset E_d of oriented edges called arcs, which consist of ordered pairs of vertices. For arc $e_d(u, w) \in E_d$ where u and v are elements in V, the origin of $e_d(u, w)$ is defined to be $o(e_d) = u$ and the terminus is $t(e_d) = v$. Its inverse arc, which is formed by switching the origin and terminus of $e_d(u, w)$, is denoted by $e_d(w, u)$. For the graph $G(V, E)$, we can obtain the associated symmetric digraph $SDG(V, E_d)$ by replacing each edge of $G(V, E)$ by the arc pair in which the two arcs are inverse to each other.

The oriented line graph of the original graph can be defined using the symmetric digraph. It is a dual graph of the symmetric digraph since its oriented edge set and vertex set are constructed from the vertex set and the oriented edge (arc) set of its corresponding symmetric digraph. The construction of the vertex set V_L and oriented edge set E_{dL} of the oriented line graph can be formulated as follows:

$$\begin{cases} V_L \; = E_d(SDG) \\ E_{dL} = \{(e_d(u,v), e_d(v,w)) \\ \qquad\qquad \in E_d(SDG) \times E_d(SDG);\; u \neq w\}. \end{cases} \tag{3}$$

The Perron-Frobenius operator $\boldsymbol{T}$ of the original graph is the adjacency matrix of the associated oriented line graph. For the (i,j)th entry of $\boldsymbol{T}$, $\boldsymbol{T}(i,j)$ 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, and is 0 otherwise.

Unlike the adjacency matrix of an undirected graph, the Perron-Frobenius operator is not a symmetric matrix. This is because of a constraint that arises in the construction of oriented edges. Specifically, the arc pair with two arcs that are the reverse of each other in the symmetric digraph are not allowed to establish an oriented edge in the oriented line graph. This constraint arises from the second requirement in the edge definition appearing in (3).

3. Ihara Coefficients for Weighted Graphs

Although (2) characterizes unweighted graphs in a compact way using the determinant, when it comes to edge-weighted graphs, i.e. the edges not only record the vertex connections but also have attributes attached on them, then the determinant scheme introduced in Section 2.2 is not applicable. This is because the Perron-Frobenius operator for an unweighted graph is the adjacency matrix of the associated oriented line graph, which only bears relational information and defaults the edge weights to binary values. This hinders the generalization of the determinant expression of the Ihara zeta function to edge-weighted graphs.

To characterize edge-weighted graphs, we generate the Perron-Frobenius operator for edge-weighted graphs from a simplified version of the Bartholdi zeta function, which is a more sophisticated zeta function with two independent variables. We then introduce a scheme to characterize the edge-weighted graphs using the polynomial coefficients of the reciprocal Ihara zeta function.

3.1. Bartholdi Zeta Function

The Bartholdi zeta function of a graph aims to generalize the zeta function using the concept of 'prime circle' where backtracking is allowable. It was first introduced and developed by Barhtholdi in [3]. For a graph $G(V,E)$, the rational expression of the Bartholdi zeta function is:

$$Z_{GB}(u,t) = \left(1-(1-t)^2u^2\right)^{\chi(G)} \times \\ \det\left(\boldsymbol{I}_N - u\boldsymbol{A} + (1-t)u^2(\boldsymbol{D}-(1-t)\boldsymbol{I}_N)\right)^{-1}. \tag{4}$$

Compared with the Ihara zeta function in its rational form (1), the rational expression for the Bartholdi zeta function involves an additional variable t, which is closely related to the cyclic bump count of a circle in a graph. One noteworthy property of the Bartholdi zeta function is that when $t=0$, it reduces to the Ihara zeta function. For more details of the Bartholdi zeta function and its relationship with the Ihara zeta function, we refer readers to [3] and [16].

3.2. Ihara Polynomial for Edge-weighted Graphs

Based on Bartholdi's work, Mizuno *et al.* [16] have developed the following determinant expression for the Bartholdi zeta function:

$$Z_{GB}(u,t) = \det\left(\boldsymbol{I}_{2M} - (\boldsymbol{B} - (1-t)\boldsymbol{J})u)\right)^{-1}. \tag{5}$$

In this form the zeta function is equivalent to (4) and is suitable for dealing with both unweighted graphs and edge-weighted graphs. There are two $2M \times 2M$ operators $\boldsymbol{B}$ and $\boldsymbol{J}$ in (5), which are both closely related to the symmetric digraph of the original graph. For a graph $G_w(V,E)$ with weighted edges, the associated symmetric digraph $SDG_w(V,E)$ can be constructed by replacing each edge of $G_w(V,E)$ by the arc pair in which the two arcs are the reverses of each other. The edge weights are then attached to each of its generating arcs. This is similar to that adopted in the case of unweighted graphs introduced in Section 2.2, except that there is the additional step of weight attachment.

Based on the symmetric digraph, the elements of the operators $\boldsymbol{J}$ and $\boldsymbol{B}$ can be computed as follows:

$$\boldsymbol{J}_{ij} = \begin{cases} 1 & \text{if } e_{di} = \bar{e}_{dj} \\ 0 & \text{otherwise,} \end{cases} \quad \boldsymbol{B}_{ij} = \begin{cases} w_{dj} & \text{if } t(e_{di}) = o(e_{dj}) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Here, $t(e_{di})$ and $o(e_{dj})$ denote the the origin and terminus of the arc e_{di} in the symmetric digraph, respectively. $\bar{e}_{dj}$ denotes the inverse arc of e_{di}. w_{dj} is the weight of the oriented edge e_{dj} in the symmetric graph. It is equal to the weight of the edge, from which the arc e_{dj} is derived, in the original graph.

In the situation of edge-weighted graphs, when $t=0$ in (5), the determinant expression of the Bartholdi zeta function reduces to the corresponding Ihara form denoted as:

$$\begin{aligned} Z_G(u) &= \det(\boldsymbol{I}_{2M} - u(\boldsymbol{B}-\boldsymbol{J}))^{-1} \\ &= \det(\boldsymbol{I}_{2M} - u\boldsymbol{T}_w)^{-1} \end{aligned} \tag{7}$$

where we define $\boldsymbol{T}_w = \boldsymbol{B} - \boldsymbol{J}$ to be the Perron-Frobenius operator for the edge-weighted graphs. $\boldsymbol{T}_w$ can also reasonably be regarded as the adjacency matrix of the oriented line graph of the original edge-weighted graph.

There are several notes needing to be made here. First, unlike in (3), there is no requirement on the exclusion of reversed arcs in the computation of the operator $\boldsymbol{B}$ in (6). Second, the operator $\boldsymbol{J}$ literally records all the reversed arc relations. Third, when it comes to unweighted graph, $\boldsymbol{T}_w$ reduces to $\boldsymbol{T}$ in (2). This is because in this case $\boldsymbol{B}$ reduces to a binary matrix representing orientations and connections only, and the operator $\boldsymbol{T}_w$, i.e. the difference of $\boldsymbol{B}$ and $\boldsymbol{J}$, naturally satisfies the all the constraints in (3). Above all, our proposed scheme establishes a generalized version of the Perron-Frobenius operator, which are both available to unweighted graphs and edge-weighted graphs.

3.3. Pattern Vectors from Ihara Coefficients

To establish pattern vectors from the Ihara zeta function for the purpose of characterizing edge-weighted graphs in machine learning, one approach is to consider taking function samples as elements. However, if this strategy is adopted, there will be dangers of sampling at poles, and these give rise to infinities. Hence, pattern vectors consisting of function samples are potentially unstable since the distribution of poles is unknown beforehand.

To overcome this problem, we note that the coefficients of the reciprocal of the Ihara zeta function, which we refer to as the Ihara coefficients, do not give rise to infinities.

According to (7), the reciprocal of the Ihara zeta function for edge-weighted graphs can be rewritten as follows:

$$\begin{aligned} Z_G^{-1}(u) &= \det(\boldsymbol{I} - u\boldsymbol{T}_w) = (u)^{2M} \det\left(\frac{1}{u}\boldsymbol{I} - \boldsymbol{T}_w\right) \\ &= c_0 + c_1 u + \cdots + c_{2M-1}u^{2M-1} + c_{2M}u^{2M}. \end{aligned} \tag{8}$$

From (8), the Ihara coefficients $c_0,\ c_1,\ ...\ c_{2M-1}$ and c_{2M} are actually the coefficients of the characteristic polynomial of the matrix $\boldsymbol{T}_w$. The pattern vectors characterizing graphs are then established with Ihara coefficients as elements. This is to be contrasted with the work of Bai *et al.* [1], who sample the zeta function values. The Ihara coefficients not only avoid the hazards of infinities that are encountered if function samples are used, but also convey direct information concerning graph structure and topology.

3.4. Spectral Interpretation and Computation

For unweighted graphs, the Ihara coefficients are essentially graph structural descriptors on the circle frequencies and vertex degrees [17][18]. For edge-weighted graphs, the Ihara coefficients have no direct links with the graph structure. Here we study the Ihara coefficients for edge-weighted md2 graphs from a spectral standpoint. We then perform a comprehensive analysis on the effectiveness of the Ihara coefficients for clustering edge-weighted md2 graphs.

As stated in Section 3.2, there is always one weighted oriented line graph associated with any weighted md2 graph. For an md2 graph, the cardinality of its vertex set is not greater than that of its edge set, subject to the least vertex degree constraint. Therefore, in practice the cardinality of the vertex set of the oriented line graph is usually much greater than, or at least equal to, that of the original graph. The construction of the oriented line graph is thus a process of transforming the original graph into a version with adjacency matrix $\boldsymbol{T}_w$ in a higher dimensional space than that of the original graph. Furthermore, the Ihara coefficients have a strong relationship with the spectrum of the Perron-Frobenius operator such that each coefficient can be derived from the polynomial of the eigenvalue set $\{\lambda_1, \lambda_2\ ...\ \lambda_n\}$ of $\boldsymbol{T}_w$ as follows:

$$c_r = (-1)^r \sum_{k_1 < k_2 < \ ...\ < k_r} \lambda_{k_1}\lambda_{k_2}\ ...\ \lambda_{k_r}. \tag{9}$$

The subscript number k in (9) runs over all possible combination of the coefficient labels. The difference between (9) and the elementary symmetric polynomials adopted in [22] is that the Ihara coefficient c_r is in fact the product of the elementary symmetric polynomial and the factor $(-1)^r$, where the subscript r indicates c_r the coefficient of the variable to the power r. Equation (9) provides an efficient way to compute the Ihara coefficients by enumerating the eigenvalues of a $2M \times 2M$ matrix. This is close in spirit to the first method for computing the polynomial coefficients suggested by Brooks [5]. Moreover, it is more efficient than the third method suggested by Brooks, which is based on computing the determinant of $2M \times 2M$ matrix $\binom{2M}{2M-r}$ times to obtain the coefficient c_r.

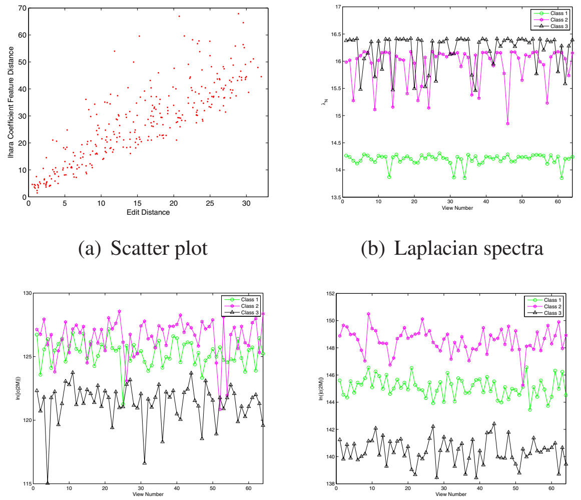
The advantages of the Ihara coefficients over the Laplacian spectral method are twofold. First, the graph is transformed to a higher dimensional feature space. Thus the characteristics from the Perron-Frobenius operator generally reflect more about the graph structure than graph matrix representations in the original domain. Second, the Ihara coefficients make use of the complete set of eigenvalues of the Perron-Frobenius operator and they do not suffer from spectral truncation, as does the pattern vectors consisting of a fixed number of leading nonzero Laplacian eigenvalues.

4. Experimental Evaluation

We evaluate our proposed scheme in two ways. We first evaluate the ability of the Ihara coefficients to distinguish between randomly generated edge-weighted graphs under controlled structural errors. Second, we focus on real-world data and assess the effectiveness of the Ihara coefficient pattern vectors in detecting object clusters.

4.1. Synthetic Data

We first investigate the relationship between graph edit distance and the feature distance between Ihara coefficient pattern vectors. The edit distance of two graphs G_1 and



(a) Scatter plot

(b) Laplacian spectra

(c) ESP coefficients

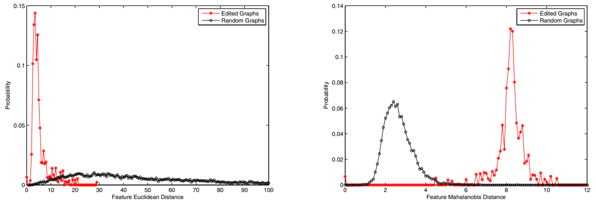
(d) Ihara coefficients

Figure 1. Feature plots

G_2 is the minimum edit cost taken over all sequences of edit operations that transform G_1 into G_2. Therefore, if we establish a new graph by deleting a certain number of edges from the seed graph and if we assign each deletion an edit cost of the deleted edge weight, the edit distance between the two graphs is equal to the overall edit cost.

We commence with a single randomly generated md2 graph as the seed graph with 100 vertices and 300 weighted edges. In this subsection, the edge weights are always generated so as to have a uniform distribution over the interval [0.5, 1.5]. We obtain edited versions of the seed graph by randomly deleting edges, with the number of deleted edges varying from 1 to 30. For each number of edge deletions, we perform 10 randomized edge deletion trials subject to the md2 constraint. We compute the Ihara coefficients using (9) and construct the pattern vector in the form of $\boldsymbol{v}_{G1} = [c_3,\, c_4,\, \ln(|c_{2M-3}|),\, \ln(|c_{2M-2}|),\, \ln(|c_{2M-1}|),\, \ln(|c_{2M}|)]^T$. The final four components of the pattern vector are scaled in a logarithmic manner to avoid unbalanced variance. The feature distance between pattern vector $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ is $d_{i,j} = \sqrt{(\boldsymbol{v}_i - \boldsymbol{v}_j)^T(\boldsymbol{v}_i - \boldsymbol{v}_j)}$, which measures the distinction between two samples in the feature space. Figure 1(a) plots the feature distances obtained using the pattern vectors composed of the Ihara coefficients, versus the corresponding graph edit distances between the seed graph and its modified variants. The main feature to note from the plot is that the Ihara coefficient distance generally follows the edit distance. Moreover, for small distances the variation of Ihara coefficient distance is approximately linear with edit distance. For large edit distance the Ihara coefficient distance becomes more scattered.

To take this study on synthetic data one step further, we study the distribution of Ihara coefficient feature distance. We investigate two sets of graphs. The first set consists of graphs which are obtained by randomly deleting one edge from a seed graph with 100 vertices and 301 weighted edges, subject to the md2 constraint. The second set are md2 graphs randomly generated with 100 vertices and 300 weighted edges. Figure 2(a) shows the distribution of the Euclidean feature distance between the pattern vectors of Ihara coefficients for the first set of graphs (red asterisk curve) and that of the second set (black circle curve). We can see that the modal distance between pattern vectors for graphs with random edge edits is much smaller than that for the structurally distinct graphs. For comparison, Figure 2(b) shows the Mahalanobis distances for the two sets of graphs. We can see that the two sets are almost totally non-overlapping. From Figures 2(a) and 2(b), the distance between pattern vectors appears to provide a scope for distinguishing between distinct graphs when there are variations in edge structure due to noise.



(a) Euclidean distance

(b) Mahalanobis distance

Figure 2. Distance distribution

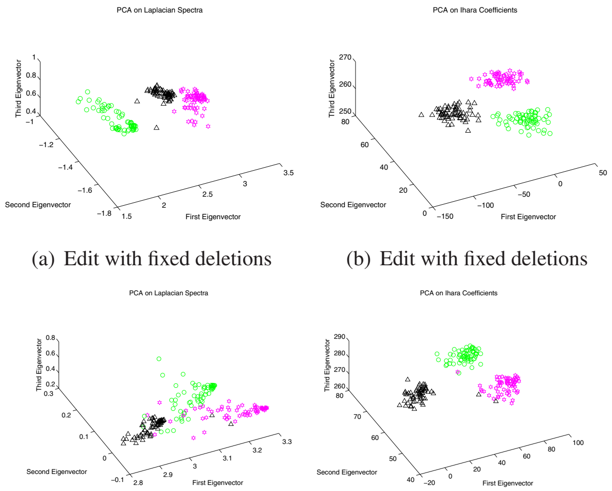
To provide an illustration and make a more comprehensive comparison with the graph spectral methods, we create two graph sets, which are established according to different types of graph edits separately, for experiments. Both sets are three classes of graphs separately derived from three seed graphs, which are again randomly generated with 100 vertices and 300 weighted edges. However, we perform two different types of edit operations on the seed graphs to establish the two graph sets separately. The first is to randomly delete eight edges at each time, and the second is with a random number of edge deletions from one to eight in each trial. We first perform the first type of edit operations on three seed graphs. Sixty four random trials of the edits are performed on each of the three seed graphs separately. Figures 1(b), 1(c) and 1(d) show the largest Laplacian eigenvalue, the final coefficient of the elementary symmetric polynomial [22] of Laplacian spectrum (ESP's) and the final Ihara coefficient as a function of trial number respectively. The main feature to note is that in the case of the Ihara coefficients, the variance is smallest and there is little overlap. The other two methods are overlapped to a more severe degree.

We then embed the pattern vectors for the two sets of edited graphs separately into a three-dimensional space using principal component analysis(PCA) to evaluate their clustering performances. We applied this procedure both to the Laplacian spectral pattern vector consisting of the second through to the seventh Laplacian eigenvalues, which is one of the most effective representations for graphs, and to the Ihara coefficient vector v_{G1}. Figure 3 shows the experimental results. Figures 3(a) and 3(b) show the clusters generated by the Laplacian spectra and the Ihara coefficients respectively, subject to the first type of graph edits. Although the spectral method appears to produce 'good' clusters in Figure 3(a), there are some deficiencies. First, the right two clusters are so close that they almost merge into one. Second, the left cluster has a non-compact ring shape and there are no graphs near the cluster center. However, the clusters in Figure 3(b) produced by the Ihara coefficient method are both more compact and more separable. Figures 3(c) and 3(d) show the results for the second type of graph edits. In this more complex situation, the Laplacian spectral method yields clusters with considerable scattering, as illustrated in Figure 3(c). However, for the Ihara coefficients, although there are a small number of outlier samples (two black triangles in the pink star cluster and one pink star in the green circle cluster), the overall performance is much better and provides the basis for a usable clustering technique.

4.2. View-based Object Recognition

We apply the pattern vectors composed of Ihara coefficients to two graph datasets. The first set of graphs are extracted from three sequences of images of model houses, referred to as the CMU, MOVI and Chalet sequences (samples shown in Figure 4(a)). The second set of graphs are extracted from images of objects in the COIL database (samples shown in Figure 4(b)). To establish graphs we first extract corner points using the Harris detector. Then we establish Delaunay graphs based on these corner points as vertices. The established Delaunay graphs are by construction md2 graphs. The edges are weighted with the exponential of the negative distance between two connected vertices, i.e. $w_{ij} = \exp[-k\,||\boldsymbol{x}_i - \boldsymbol{x}_j||]$ where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are coordinates of corner points i and j in an image and k is a scalar scaling factor. The graphs extracted from sample objects are superimposed upon the sample images in Figure 4.

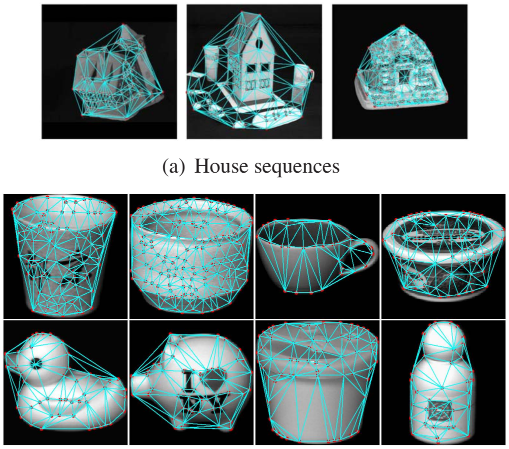
We first explore which coefficients provide the strongest discrimination between the graphs for the different object classes. Although the full set of coefficients associated with a graph can be used to construct a pattern vector, only a subset of the coefficients contribute significantly. Some coefficients may be redundant. Some others may reduce the effectiveness of the clustering algorithm. We thus need to select the subset of salient coefficients, i.e. those that take on distinct values for different classes and exhibit small within class variance. To do this, we compute the between-class scatter $S_b = \sum_{i=1}^{M} N_i(\bar{c}_{k,i} - \bar{c}_k)^2$ and the within-class scatter $S_w = \sum_{i=1}^{M} \sum_{c_{k,i,j} \in C_i} (c_{k,i,j} - \bar{c}_{k,i})^2$ of the individual coefficients, where $\bar{c}_k$ is the mean of the c_k samples, $\bar{c}_{k,i}$ is the mean of the c_k samples in class C_i, N_i is the number of the c_k samples in class C_i and M is the total number of classes. We then use the criterion function $J = (S_b + S_w)/S_w$ to evaluate the performance of individ-



(a) Edit with fixed deletions (b) Edit with fixed deletions

(c) Edit with random deletions (d) Edit with random deletions

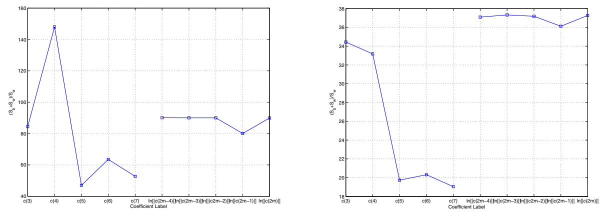
Figure 3. Clusters for three classes of graphs



(a) House sequences

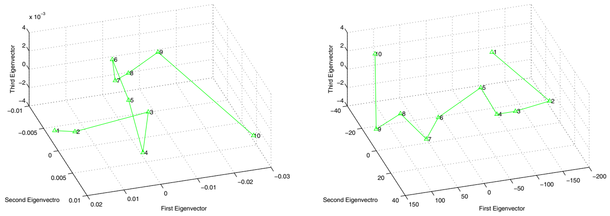
(b) COIL datasets

Figure 4. Datasets for Experiments



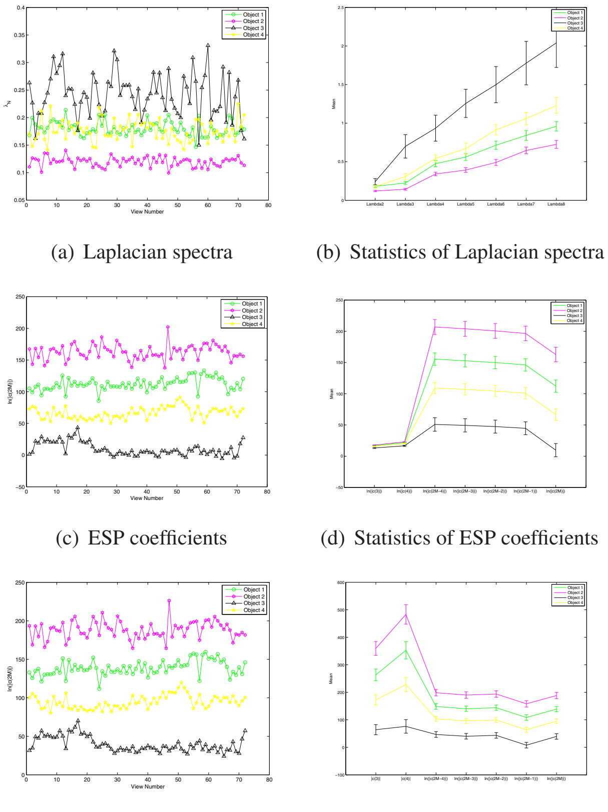
(a) Houses (b) COIL

Figure 5. Criterion function value



(a) Laplacian spectra (b) Ihara coefficients

Figure 6. Eigenprojection of graphs from Chalet houses



(a) Laplacian spectra (b) Statistics of Laplacian spectra

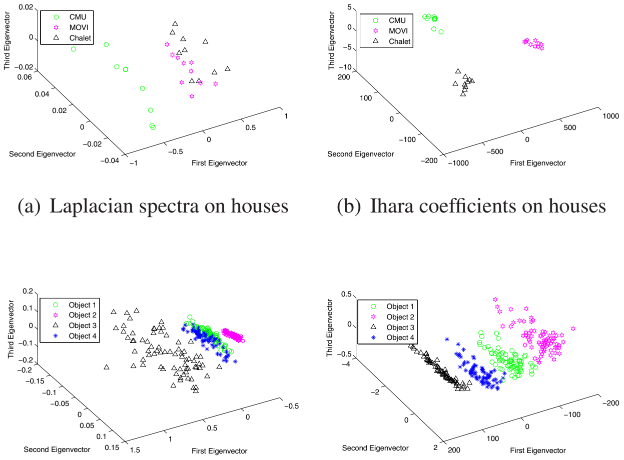
(c) ESP coefficients (d) Statistics of ESP coefficients

(e) Ihara coefficients (f) Statistics of Ihara coefficients

Figure 7. Coefficient data for the COIL dataset

ual coefficients. We select the coefficients according to the criterion that the individual coefficients making the largest contributions to the criterion function are the most significant ones. For each of three selected objects, ten sample images are used as training data to compute the criterion function value. Figures 5(a) and 5(b) show the criterion function values for the coefficients extracted from the house dataset and the COIL dataset respectively. It is clear that the leading few and trailing coefficients contribute more to distinguishing the objects than the intermediate coefficients.

To investigate whether the proposed scheme can learn the structural variation within a graph class, we project the pattern vectors onto the leading three eigenvectors of the class covariance matrix and thus embed the graphs in a pattern space. We use $\boldsymbol{v}_{G2} = [c_3,\ c_4,\ \ln(|c_{2M}|)]^T$ as the pattern vector characterizing the edge-weighted graphs. Figures 6(a) and 6(b) show the projections of the graphs from the Chalet images, based on the Laplacian spectra and $\boldsymbol{v}_{G2}$ respectively. Each point in the pattern space is marked with a view number which corresponds to the camera angle. We can see that the spectral method yields a trajectory hardly tractable. However, the Ihara coefficients produce a clear trajectory and the neighboring images in the sequence are generally close together in the eigenspace.



(a) Laplacian spectra on houses (b) Ihara coefficients on houses

(c) Laplacian spectra on COIL (d) Ihara coefficients on COIL

Figure 8. Clusters for real-world object classes

Next we evaluate the performance of the pattern vectors in distinguishing real-world graph classes. Figure 7 provides some details of the variation of the Laplacian eigenvalues, the ESP's and the Ihara coefficients for graphs extracted from the first four COIL objects in Figure 4(b). In these plots different colored lines correspond to different COIL objects. In the left hand column of Figure 7, we show coefficient values plotted as a function of view numbers for the four objects. In the right hand column of Figure 7, we show the coefficient mean values and standard errors for the four objects over different views as a function of coefficient indices. The main features to note from these plots are that a) both the Ihara coefficients and the ESP's are better separated than the Laplacian eigenvalues, b) there is now little difference between the Ihara coefficients and the ESP's. This latter point is attributable to the highly regular nature of the Delaunay triangulation.

We then use $\boldsymbol{v}_{G2} = [c_3,\ c_4,\ \ln(|c_{2M}|)]^T$ and $\boldsymbol{v}_{G3} = [\ln(|c_{2M-2}|),\ \ln(|c_{2M-1}|),\ \ln(|c_{2M}|)]^T$ as the pattern vectors characterizing the edge-weighted graphs extracted from house sequences and COIL dataset respectively, and embed them into a three-dimensional space using PCA. For

Pattern Vector	Number of Object Classes				
	4	5	6	7	8
Laplacian Spectra	0.94	0.87	0.87	0.86	0.87
Ihara Coefficients	0.99	0.95	0.90	0.88	0.89

Table 1. Rand Indices

comparison, we use the leading three non-zero Laplacian eigenvalues as the spectral pattern vector for the two dataset. Figures 8(a) and 8(b) show the clusters of graphs extracted from the house sequences, produced by the Laplacian spectral method and the Ihara coefficients respectively. Figures 8(c) and 8(d) show the clusters of graphs extracted from the first four COIL objects in Figure 4(b), produced by the Laplacian spectral method and the Ihara coefficients respectively. From Figure 8 we can see that the Ihara coefficients outperform the Laplacian spectral method in producing good clusters.

To take the quantitative evaluation of the pattern vectors, we concentrate our attention on the COIL dataset, and evaluate the clustering performance obtained with different numbers of object classes. After performing PCA on the pattern vectors, we locate the clusters using the K-means method and calculate the Rand index for the resulting clusters. The Rand index is defined as $R_I = Z/(Z+Y)$, where Z is the number of agreements and Y is the number of disagreements in cluster assignment. It takes a value in the interval [0,1], where 1 corresponds to a perfect clustering. The Rand indices for the Laplacian spectral method and for the Ihara coefficients are listed in Table 1. From Table 1 it is clear that the Ihara coefficients outperform Laplacian spectra for all numbers of object classes studied.

5. Conclusions

We have studied how to extract characteristics from edge-weighted graphs using the Ihara zeta function, and have exploited the resulting characterization for the purposes of clustering edge-weighted graphs. We use Ihara coefficients to construct pattern vectors, rather than sampling the zeta function which associates the potential pitfall of encountering infinities. The main contribution in this paper is that we provide a route that allows the Ihara coefficients to be extended from unweighted to edge-weighted graphs. This is achieved by establishing the Perron-Frobenius operator for edge-weighted graphs with the assistance of the Bartholdi zeta function. We perform a spectral analysis that explains why the Ihara coefficients are more effective in distinguishing graph classes than the graph spectral methods. Experiments were conducted on both synthetic and real-world data, and reveal not only that the Ihara coefficients are effective for graph clustering but that they also outperform the Laplacian spectra in graph characterization.

Acknowledgments

We acknowledge the financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

[1] X. Bai, E. R. Hancock, and R. C. Wilson. Graph characteristics from the heat kernel trace. *Pattern Recognition*, 42(11):2589–2606, 2009.

[2] H. G. Barrow and R. M. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4:83–84, 1976.

[3] L. Bartholdi. Counting paths in graphs. *L' Enseignement Mathmatique*, 45:83–131, 1999.

[4] H. Bass. The ihara-selberg zeta function of a tree lattice. *International Journal of Mathematics*, 6:717–797, 1992.

[5] B. P. Brooks. The coefficients of the characteristic polynomial in terms of the eigenvalues and the elements of an n$\times$n matrix. *Applied Mathematics Letters*, 19:511–515, 2006.

[6] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[7] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. *In CVPR*, 2005.

[8] D. Emms, S. Severini, R. C. Wilson, and E. R. Hancock. Coined quantum walks lift the cospectrality of graphs and trees. *Pattern Recognition*, 42(9):1988–2002, 2009.

[9] R. Fergus, P. Perona, and A. Zisserman. Weakly supervised scale-invariant learning of models for visual recognition. *IJCV*, 71(3):273–303, 2007.

[10] M. Fischler and R. Elschlager. The representation and matching of pictorical structures. *IEEE Trans. Computer*, 22(1):67–92, 1973.

[11] Y. Ihara. On discrete subgroups of the two by two projective linear group over p-adic fields. *Journal of Mathematics Society Japan*, 18:219–235, 1966.

[12] M. Kotani and T. Sunada. Zeta functions of finite graphs. *Journal of Mathematics University of Tokyo*, 7(1):7–25, 2000.

[13] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. *In ICCV*, 2005.

[14] T. Liu and D. Geiger. Approximate tree matching and shape similarity. *In ICCV*, 1999.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[16] H. Mizuno and I. Sato. Bartholdi zeta function of graph coverings. *Journal of Combinatorial Theory, Series B*, 89:27–41, 2003.

[17] P. Ren, R. C. Wilson, and E. R. Hancock. Pattern vectors from the ihara zeta function. *In ICPR*, 2008.

[18] G. Scott and C. K. Storm. The coefficients of the ihara zeta function. *Involve - A Journal of Mathematics*, 1(2):217–233, 2008.

[19] T. S. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. *In ICCV*, 2001.

[20] L. S. Shapiro and J. M. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10:283–288, 1992.

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[22] R. C. Wilson, B. Luo, and E. R. Hancock. Pattern vectors from algebraic graph theory. *IEEE TPAMI*, 27(7):1112–1124, 2005.

[23] R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. *In CVPR*, 2008.

[24] D. Zhao and X. Tang. Cyclizing clusters via zeta function of a graph. *In NIPS*, 2008.