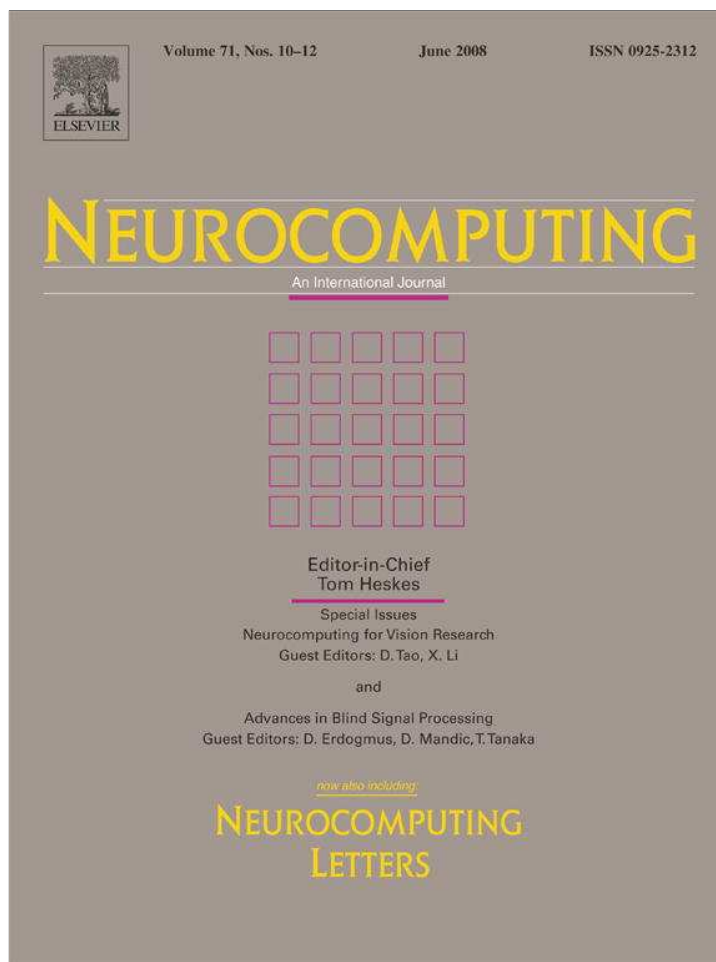


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

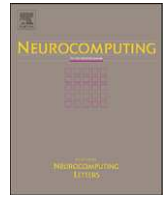
In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Isotree: Tree clustering via metric embedding

Bai Xiao^{a,*}, Andrea Torsello^b, Edwin R. Hancock^c^a Department of Computer Science, University of Bath, Bath BA2 7AF, UK^b Dipartimento di Informatica, University "Ca' Foscari" of Venice, Via Torino 155, 30172 Venezia Mestre, Italy^c Department of Computer Science, University of York, York YO10 5DD, UK

ARTICLE INFO

Available online 4 March 2008

Keywords:

Graph clustering
Metric embedding
Spectral graph theory

ABSTRACT

One of the problems that hinders the spectral analysis of trees is that they have a strong tendency to be co-spectral. As a result, structurally distinct trees possess degenerate graph-spectra, and spectral methods can be reliably used to neither compute distances between trees nor to cluster trees. The aim of this paper is to describe a method that can be used to alleviate this problem. We use the ISOMAP algorithm to embed the trees in a Euclidean space using the pattern of shortest distances between nodes. From the arrangement of nodes in this space, we compute a weighted proximity matrix, and from the proximity matrix a Laplacian matrix is computed. By transforming the graphs in this way we lift the co-spectrality of the trees. The spectrum of the Laplacian matrix for the embedded graphs may be used for purposes of comparing trees and for clustering them. Experiments on sets of shock graphs reveal the utility of the method on real-world data.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Tree structures have been used with considerable effect in computer vision to represent both object shape, scene structure and object articulation [27]. Examples include the use of shock trees to represent object boundary structure [27], the use of free-trees to represent human form [12] and the use of trees as compact image data-structures [34]. One of the problems that arises in the manipulation of large amounts of tree data is that of clustering. Although this task can be effected by applying pairwise clustering methods to the edit distance between trees, it does not allow the distribution of trees to be visualized or the effects of systematic changes in tree-structure to be assessed. Moreover, since computing the edit distance between trees relies on the availability of correspondences between nodes, and this is potentially an NP-hard problem, the computational overheads can be large.

One way to overcome the problem of computing the distance between discrete structures is to embed them in a low-dimensional space that minimizes the distortion. In this low-dimensional space, distances may be computed by taking a standard norm between the embedded pattern vectors. The problem of how to construct such an embedding has been the focus of activity in pattern recognition for several decades. For instance principal components analysis

(PCA) projects pattern vectors into a low-dimensional space that maximally preserves the variance of the original data [13]. Multidimensional scaling (MDS), on the other hand, can be used to embed non-ordinal data into a low-dimensional space which preserves the relational pattern residing in the set of pairwise distances between data-items by minimizing the stress of the data [7]. However, these pattern analysis methods can only be applied for the data which is in vectorial form, or a distance function is to hand, and hence do not extend easily to discrete structures such as trees or graphs. In the mathematics literature, on the other hand, there is a considerable body of work aimed at understanding how graphs can be embedded in manifolds. Broadly speaking there are three ways in which the problem has been addressed. First, the graph can be interpolated by a surface whose genus is determined by the number of nodes, edges and faces of the graph. Second, the graph can be interpolated by a hyperbolic surface which has the same pattern of geodesic (internode) distances as the graph [1,5]. Third, a manifold can be constructed whose triangulation is the simplicial complex of the graph [33,21]. A review of methods for efficiently computing distance via embedding is presented in the recent paper of Hjaltason and Samet [11].

In the pattern analysis community, there has recently been renewed interest in the use of embedding methods motivated by graph theory. One of the best known of these is ISOMAP [30]. Here a neighborhood ball is used to convert data-points into a graph, and Dijkstra's algorithm is used to compute the shortest (geodesic) distances between nodes. The matrix of geodesic distances is used as input to MDS. The resulting algorithm has been demonstrated to locate well-formed manifolds for a number

* Corresponding author.

E-mail addresses: xb202@cs.bath.ac.uk (B. Xiao), torsello@dsi.unive.it (A. Torsello), erh@cs.york.ac.uk (E.R. Hancock).

of complex data sets. Related algorithms include locally linear embedding [16] which is a variant of PCA that restricts the complexity of the input data using a nearest neighbor graph, and the Laplacian eigenmap that constructs an adjacency weight matrix for the data-points and projects the data onto the principal eigenvectors of the associated Laplacian matrix (the degree matrix minus the weight matrix) [2]. Collectively, these methods are sometimes referred to as manifold learning theories.

In this paper, we are interested in the problem of embedding trees in a pattern space for the purposes of both visualization and analysis (including clustering and classification). One of the methods that has proved effective for the embedding and pattern analysis of trees is spectral graph theory [6]. For instance, Dickinson and his co-workers [25,14] have shown how graph-spectra can be used to index shock-trees. There are two criticisms that can be leveled at the spectral analysis of trees. First, graphs that are not isomorphic can be co-spectral. As demonstrated by Schwenk et al. [23,3], due to their sparse edge-structure this problem is accentuated for trees. The second problem is the distortion produced by the embedding. In [14] the metric embedding algorithm gives a distortion that is proportional to $\sqrt{\log \log |A|}$, where A is a set of points in the original metric space, $|A|$ is the number of points in that space. In [9] the distortion is $\Omega(l(T)^{1/d})$, where $l(T)$ is the number of leaves in a tree T . So, when size of the trees becomes large, then the distortion will also become large.

To overcome these problems in this paper we investigate whether methods from manifold learning theory can be combined with spectral graph theory to develop effective tools for tree analysis. The idea is to use manifold learning methods to embed the trees in a low-dimensional space, and to perform spectral analysis on the co-ordinate data for the embedded tree-nodes. We proceed as follows. We commence by using a strategy similar to ISOMAP to embed the trees in a Euclidean pattern space. This is done by computing a matrix of shortest (geodesic) distances between nodes in the tree. We then apply MDS to the distance matrix, and this embeds the individual nodes of the tree in a Euclidean space. Once embedded in this space, we construct a weighted Laplacian matrix for the nodes of the tree by exponentiating the negative squared-distance between nodes. The spectrum of eigenvalues of the Laplacian can be used for the purposes of tree clustering and visualization.

2. Metric embedding of trees

The problem of embedding finite metric space into Euclidean spaces, or other normed spaces, that approximately preserve the metric is one that has received considerable attention in recent years. A number of ways have been proposed for measuring the quality of an embedding procedure. The *distortion* has been widely accepted as a measure of the quality of the embedding. For a finite metric space (X, d) and $c \geq 1$, there is an embedding φ of X into Y where for every two points $x_1, x_2 \in X$ satisfy the condition

$$d(x_1, x_2) \geq \|\varphi(x_1) - \varphi(x_2)\| \geq \frac{1}{c} d(x_1, x_2) \quad (1)$$

Such an embedding is said to have *distortion* $\leq c$ [16]. Recently low-distortion embedding has provided powerful tools for designing efficient pattern analysis algorithms. This is because that they enable us to reduce problems defined over difficult metrics to problems over much simpler ones.

The starting point for most metric embedding methods is Bourgain's [4] Lemma:

Any finite metric (X, d) can be embedded into \mathbb{R}^p with $p < \infty$ with distortion $O(\log |X|)$.

We denote \mathbb{R}^n equipped with l_q norm by \mathbb{R}_q^n . The Euclidean norm is l_2 . The l_q norm is defined as $\|(x_1, \dots, x_n)\|_q = (\sum |x_i|^q)^{1/q}$. The original bound on p proved by Bourgain was exponential with n and too large to be of practical use. We seek to introduce an embedding with a much lower distortion.

2.1. Metric embedding of trees by using isomap

We first define a suitable metric for the trees or graphs. For a given graph $G = (A, E)$, A represents the nodes in the graph and E represents the edge relations between the nodes. Suppose that D is a metric on the graph G . The metric must satisfy the condition that for any three vertices u, v and $w \in A$, if $D(u, v) = D(w, v) \geq 0$, then $D(u, u) = 0$ and $D(u, v) \leq D(u, w) + D(w, v)$. There are many ways to define metric distances on a graph. The best known is the shortest-path metric $D(u, v) = \delta(u, v)$, which is the shortest path distance between u and v for all $u, v \in A$. In fact, if the graph G is a tree, the shortest path between any two vertices is unique, and the weights of the shortest paths between any two vertices will define a metric $D(\dots)$. Since we can treat trees as a special kind of graph, we can use the shortest-path metric for trees.

Our goal is to find a low-distortion or distortion-free embedding from the tree metric space into a normed space. Here we use Isomap (isometric feature mapping) [30] as a way to solve the low-distortion tree embedding problem. The idea behind Isomap is to apply classical MDS [7] to map data points from their high-dimensional input space to low-dimensional coordinates of a nonlinear manifold. The key contribution is hence to apply MDS to the pairwise distances not in the input Euclidean space, but in the geodesic space of the manifold.

Although the method was originally devised for dimensionality reduction, we can use it here for the low-distortion tree embedding problem. Viewed as an isometric feature mapping, Isomap is a mapping $f : X \rightarrow Y$ from the observation space X to a Euclidean feature space Y that preserves as closely as possible the intrinsic metric structure of the observations, i.e. the distances between observations as measured along geodesic (shortest) paths of X [30]. The distortion c in this embedding is nearly 1.

For trees, the embedding procedure is straightforward. We first construct the shortest path distance matrix S for each tree. Each element d_{i_1, i_2} in S is the shortest path distance between the pair of nodes i_1 and i_2 of the tree. We embed each tree in a Euclidean space by performing MDS on the matrix S .

2.2. Multidimensional scaling

MDS is a procedure which allows data specified in terms of a matrix of pairwise distances to be embedded in a Euclidean space. The pairwise geodesic distances between nodes d_{i_1, i_2} are used as the elements of an $N \times N$ dissimilarity matrix S , whose elements are defined as follows:

$$S_{i_1, i_2} = \begin{cases} d_{i_1, i_2} & \text{if } i_1 \neq i_2 \\ 0 & \text{if } i_1 = i_2 \end{cases} \quad (2)$$

In this paper, we use the classical MDS method. The first step of MDS is to calculate a matrix T whose element with row r and column c is given by $T_{rc} = -\frac{1}{2}[d_{rc}^2 - \hat{d}_r^2 - \hat{d}_c^2 + \hat{d}^2]$, where $\hat{d}_r = (1/N) \sum_{c=1}^N d_{rc}$ is the average dissimilarity value over the r th row, \hat{d}_c is the similarly defined average value over the c th column and $\hat{d} = (1/N^2) \sum_{r=1}^N \sum_{c=1}^N d_{r,c}$ is the average similarity value over all rows and columns of the similarity matrix T .

We subject the matrix T to an eigenvector analysis to obtain a matrix of embedding co-ordinates X . If the rank of T is $k, k \leq N$, then we will have k non-zero eigenvalues. We arrange these k non-zero eigenvalues in descending order, i.e. $l_1 \geq l_2 \geq \dots \geq l_k > 0$.

The corresponding ordered eigenvectors are denoted by u_i where l_i is the i th eigenvalue. The embedding co-ordinate system for the graphs obtained from different views is $X = [f_1, f_2, \dots, f_s]$, where $f_i = \sqrt{l_i}u_i$ are the scaled eigenvectors. For the tree-nodes indexed i , the embedded vector of co-ordinates is $x_i = (X_{i,1}, X_{i,2}, \dots, X_{i,s})^T$.

2.3. Spectral characterization

Our aim is cluster trees using a spectral characterization of the co-ordinates of the embedded nodes. To this end, we commence by computing a weighted proximity matrix W with elements

$$W_{i_1, i_2} = \begin{cases} \exp\left[\frac{-\|x_{i_1} - x_{i_2}\|_2^2}{2\sigma^2}\right] & \text{if } \|x_{i_1} - x_{i_2}\|_2 < r \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where σ is a scale constant and r is the radius of a neighbourhood hypersphere in the embedding space. Unfortunately, the matrix W may have negative eigenvalues. Hence, we turn our attention instead to the Laplacian matrix, since it is positive semi-definite and therefore has positive or zero eigenvalues. The Laplacian matrix is $L = W - \Delta$ where Δ is diagonal degree matrix with elements $\Delta(i, i) = \sum_{j \in V} W(i, j)$. The spectral decomposition of the Laplacian matrix is $L = \sum_{i=1}^n \lambda_i e_i e_i^T$, where λ_i^k is the i th eigenvalue and e_i is the corresponding eigenvector of the Laplacian matrix L . Our spectral characterization of the tree is based on the vector of N leading Laplacian eigenvalues $\vec{B} = (\lambda_1, \dots, \lambda_N)^T$. We can perform pattern analysis on sets of trees by applying clustering or dimensionality reduction techniques to the vectors of Laplacian eigenvalues.

2.4. Principal component analysis

After constructing the feature vector \vec{B} , our next aim is to construct a pattern-space for a set of graphs with pattern vectors $\vec{B}_k, k = 1, \dots, M$. There are a number of ways in which the graph pattern vectors can be analyzed. Here, for the sake of simplicity, we use PCA. We commence by constructing the matrix $S = [\vec{B}_1 | \vec{B}_2 | \dots | \vec{B}_k | \dots | \vec{B}_M]$ with the graph feature vectors as columns. Next, we compute the covariance matrix for the elements of the feature vectors by taking the matrix product $C = \hat{S}\hat{S}^T$, where \hat{S} is computed by subtracting the mean of the feature vectors from each column of the matrix S . We extract the principal components directions by performing the eigendecomposition $C = \sum_{i=1}^M l_i \vec{u}_i \vec{u}_i^T$ on the covariance matrix C , where the l_i are the eigenvalues and the \vec{u}_i are the eigenvectors. We use the first s leading eigenvectors (three in practice for visualization purposes) to represent the graphs extracted from the images. The co-ordinate system of the eigenspace is spanned by the s orthogonal vectors $\vec{U} = [\vec{u}_1, \vec{u}_2, \dots, \vec{u}_s]$. The individual graphs represented by the vectors $B_k, k = 1, 2, \dots, M$ can be projected onto this eigenspace using the

formula $\vec{B}_k = \vec{U} \vec{B}_k$. Hence each graph G_k is represented by an s -component vector B_k in the eigenspace.

3. Experiments

In this section we experiment with the application of our clustering algorithm. We tested our algorithm on two databases. The first one is shock tree graphs which composed of 150 silhouettes of 10 kinds of objects [32]. The second database is COIL [19] which is a more realistic and complex database.

3.1. Experiments on shock trees

The shock tree is a tree structure based representation of the differential structure of the boundary of a 2D shape. It is obtained by locating the shape skeleton, and examining the differential behavior of the radius of the bitangent circle from the skeleton to the object boundary, as the skeleton is traversed [15]. The idea is to evolve the boundary of an object to a canonical skeletal form using the reaction–diffusion equation. The skeleton represents the singularities in the curve evolution, where inward moving boundaries collide. With the skeleton to hand, then the next step is to devise ways of using it to characterize the shape of the original object boundary. We follow the idea [26] by labeling points on the skeleton using so-called shock-labels. The skeletons can then be abstracted as trees in which the level in the tree is determined by their time of formation [26,15]. The later the time of formation, and hence their proximity to the center of the shape, the higher the shock in the hierarchy. The shock tree extraction process has been further improved by Torsello and Hancock [31] recently. In this paper we use the methods in [31] to extract tree structure representations from the silhouettes. A representative view of each object is shown in Fig. 1.

We have followed the procedure outlined in Section 2. First we embed the trees in a Euclidean pattern space. This involves first computing the shortest (geodesic) distance between nodes and applying MDS to the shortest distance matrix S . The result is the embedding of the individual nodes of the tree in a Euclidean space. After the nodes have been embedded in this way, we construct a weighted Laplacian matrix for the nodes of the tree by exponentiating the negative squared-distance between nodes. We then extract the spectrum of eigenvalues of the Laplacian matrix and encode them as a vector for each tree. Finally, we apply the PCA to these vectors to visualize them in a two-dimensional space.

In our experiments, we will compare the results obtained by using our algorithm with those obtained by using spectral analysis of trees. We compare two different spectral methods. The first method is so-called “eigenvalue” method [17,22]. We commence by first constructing the Laplacian matrix L for the tree. Then we use the leading Laplacian eigenvalues λ_i of the matrix L to

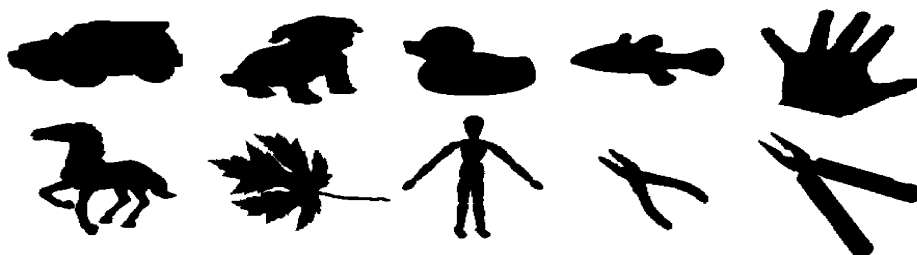


Fig. 1. Sample views of the 10 objects.

construct the spectral feature vector $\mathbf{B} = (\lambda_1, \dots, \lambda_N)^T$. After the spectral feature vectors have been extracted from the trees, we also apply PCA as dimension reduction techniques for the purposes of visualisation.

The second spectral method which has been recently proposed by Wilson et al. [35]. In their recent paper, Wilson et al. [35] have reported a family of invariants that can be computed by applying symmetric polynomials to the elements from the graph spectral matrix. The spectral matrix is obtained by performing eigendecomposition on the Laplacian matrix L of the graph, $L = \sum_{i=1}^n \lambda_i e_i e_i^T$, where λ_i is the i th eigenvalue e_i is the corresponding eigenvector of the symmetric Laplacian L . The spectral matrix S_M is defined as

$$S_M = (\sqrt{l_1}u_1, \sqrt{l_2}u_2, \dots, \sqrt{l_n}u_n)^T \quad (4)$$

It can also be written as $S_M = \sqrt{A_L} \Phi_L$, where A_L and Φ_L are the eigen matrices of the Laplacian matrix corresponding to the eigenvalues and eigenvectors. There are two sets of symmetric polynomials defined in the paper [35], the elementary symmetric polynomials and power-sum symmetric polynomials. For a set of variables $\{v_1, v_2 \dots v_n\}$ the elementary symmetric polynomials are defined as

$$\begin{aligned} E_1(v_1, \dots, v_n) &= \sum_{i=1}^n v_i \\ E_2(v_1, \dots, v_n) &= \sum_{i=1}^n \sum_{j=i+1}^n v_i v_j \\ &\vdots \\ E_r(v_1, \dots, v_n) &= \sum_{i_1 < i_2 < \dots < i_r} v_{i_1} v_{i_2} \dots v_{i_r} \\ &\vdots \\ E_n(v_1, \dots, v_n) &= \prod_{i=1}^n v_i \end{aligned}$$

The output for the n set input variable $\{v_1, v_2 \dots v_n\}$ is E_1, E_2, \dots, E_n .

The power-sum symmetric polynomial functions, which are defined as below:

$$\begin{aligned} P_1(v_1, \dots, v_n) &= \sum_{i=1}^n v_i \\ P_2(v_1, \dots, v_n) &= \sum_{i=1}^n v_i^2 \end{aligned}$$

$$\begin{aligned} P_r(v_1, \dots, v_n) &= \sum_{i=1}^n v_i^r \\ &\vdots \\ P_n(v_1, \dots, v_n) &= \sum_{i=1}^n v_i^n \end{aligned}$$

also form a basis set over the set of symmetric polynomials. Any polynomial function which is invariant to permutation of the variable indices (v_1, v_2, \dots, v_n) can be expressed in terms of one of these sets of polynomials. The two sets of polynomials are related to one another by the Newton-Girard formula:

$$E_r = \frac{(-1)^{r+1}}{r} \sum_{k=1}^r (-1)^{k+r} P_k E_{r-k} \quad (5)$$

where we have used the shorthand E_r for $E_r(v_1, \dots, v_n)$ and P_r for $P_r(v_1, \dots, v_n)$. As a consequence, the elementary symmetric polynomials S_r can be efficiently computed using the power-sum symmetric polynomials. In [35] Wilson et al. proposed to use the elements of the spectral matrix $\sqrt{A_L} \Phi_L$ as input for the symmetric polynomial to compute the invariants as features for graph characterization. In our experiment, the arguments of the polynomials are the column elements of the spectral matrix $\sqrt{A_L} \Phi_L$. We choose six different symmetric polynomials invariants, which are $E_1, E_2, \dots, E_5, E_6$, to construct a feature vector for our experiment.

Our first experiment compares our algorithm with the two spectral analysis of the trees from the entire database of 150 shock trees. We extracted feature vectors for the shock trees by using the methods mentioned above, then we perform PCA on the feature vectors to project them into two-dimensional spaces. In Fig. 2, the left-hand plot shows the results of the spectral eigenvalue analysis of the shock trees, the middle one is the result of spectral polynomial method, and the right-hand figure is the result by using our algorithm. In the different plots, the different shapes of a particular class are represented by symbols of different color. There is a legend in the top left-hand corner of each plot that explains the shape correspondence of each of the symbols.

There are a number of points that can be drawn from these plots. First, in the case of the spectral eigenvalue analysis the data distribute themselves along a trajectory in the embedding space. This implies that the tree-spectra are nearly uni-dimensional in terms of the distribution of both their variance and dissimilarity. This underlines the problem of co-spectrality mentioned in the introduction. Moreover, in the case of the spectral eigenvalue

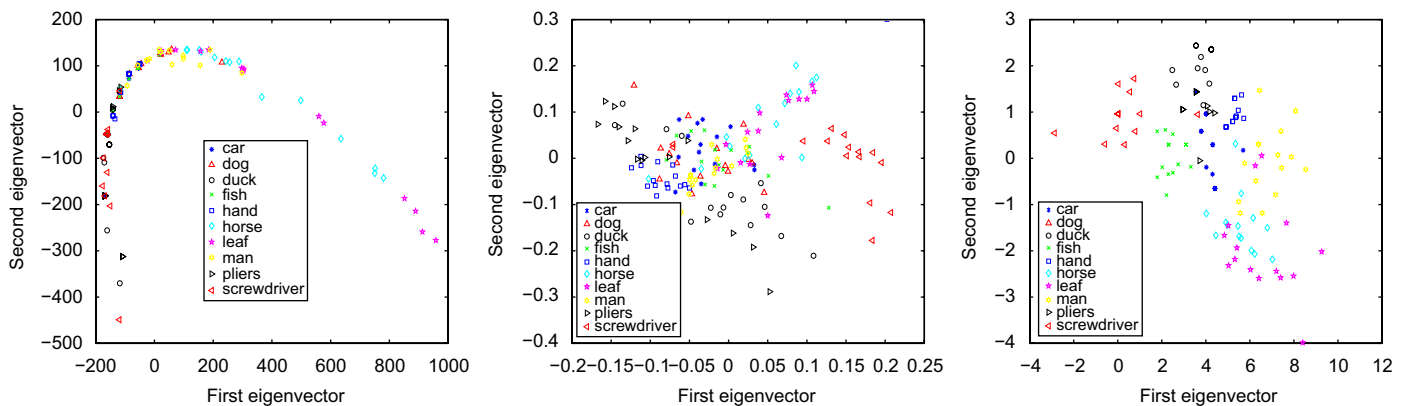


Fig. 2. Shock tree clustering—left (spectral eigenvalue method), middle (spectral polynomial method) and right (our method).

analysis the different shapes are interspersed along the trajectory. It is hence not possible to allocate the shapes to reliably assign shapes to classes on the basis of their position in the plots. The spectral polynomial result looks better than the spectral eigenvalue result. But there is considerable overlap near the center of the plot. In the case of the Isomap embedding, the trees could be better separated.

3.2. Qualitative analysis

To give these three methods a qualitative evaluation, we introduce the idea of Rand index [20]. The Rand index is defined as

$$R_I = \frac{C_A}{C_A + W_D} \quad (6)$$

where C_A is the number of agreements and W_D is the number of disagreements in cluster assignment. If two objects are in the same cluster in both the ground truth clustering and the clustering from our experiment, this counts as an agreement. If two objects are in the same cluster in the ground truth clustering but in different clustering from our experiment, this counts as a disagreement. The index is hence the fraction of views of a particular class that are closer to an object of the same class than to one of another class. The Rand index takes a value in the interval $[0, 1]$, where 1 corresponds to a perfect clustering. Since we have already known the ground truth clustering before the experiments, finding out the clustering results from the low-dimensional embedded coordinates can use the K-nearest algorithm. The idea is for each embedded coordinate we search its K nearest-neighbors. We then check the ground truth cluster numbers of these K-neighbors to see if over half of these neighbors belong to the same category as its ground truth one. In our experiment we set K to 5. For example, a tree where in the ground truth it belongs to the first object class. In the embedded coordinate space, we check its five nearest-neighbors. If three or more of its neighbors also belong to the first object class in their ground truth clustering, then we say it is an agreement. Otherwise it is a disagreement. We repeat this process for shock trees used in the experiments and compute the Rand Index value according to Eq. (6). In our experiment the spectral eigenvalue method's rand index value is 0.58, the spectral polynomial is 0.76 and our method is 0.89 which is the highest.

We have repeated the procedure above for a smaller database which contains only three representative shapes. The three shapes used for test are the hands, the leaves and the men. For each shape there are 15 different views corresponding to different viewing

directions. The left-hand plot of Fig. 3 shows the results of spectral eigenvalue analysis. The middle one is from the spectral polynomial method. The right-hand plot shows the result of applying Isomap embedding to the three shapes. From the results, it can be concluded that the Isomap embedding gives the best result in separation.

To investigate the role of distance in more detail, Fig. 4 shows the scatter plots of different distances estimate for the trees with the edit distance computed from [32]. In the paper [32], Torsello et al. propose a robust and stable way to compute edit distance between pairs of shock trees. For the whole 150 data sets, we compute a 150×150 size pairwise distance matrix by using the methods from [32]. For the other three methods including ours, we also compute the pairwise dissimilarity distance matrices. The left-hand of Fig. 4 shows scatter plot between the distance matrix of the spectral eigenvalue (x -axis) and the distance matrix computed from edit distance (y -axis). The middle plot shows the scatter plot between the distance matrix computed from spectral polynomial method (x -axis) and the distance matrix from edit distance (y -axis). The right-hand plot shows the scatter plot between the distance matrix computed from our methods (x -axis) and the distance matrix from edit distance (y -axis). In the first (left-hand) scatter plot there is a regression trend in the data, while in the last (right-hand) plot there is no trend. Hence, the dissimilarity distance computed for the embedding method better reflect the edit distances between the trees.

3.3. Experiments on COIL databases

In the second experiment, we experimented our algorithms on COIL [19] database. The database contains views of 3D objects under controlled viewer and lighting conditions. For each object in the database there are 72 equally spaced views, which are obtained as the camera circumscribes the object. We study the images from eight example objects. A sample view of each object is shown in Fig. 5. For each image of each object we extract feature points using the method of [10]. We have extracted graphs from the images by computing the Voronoi tessellations of the feature points, and constructing the region adjacency graph, i.e. the Delaunay triangulation, of the Voronoi regions. Our embedding procedure has been applied to the resulting graph structures.

In Fig. 6, the left-hand plot is the spectral eigenvalue clustering results, the middle is spectral polynomial clustering result while the right-hand plot is the result by using our method. The main qualitative feature is that the different views of the eight objects are more overlapped by using the spectral eigenvalue and polynomial methods than our method. This time for the

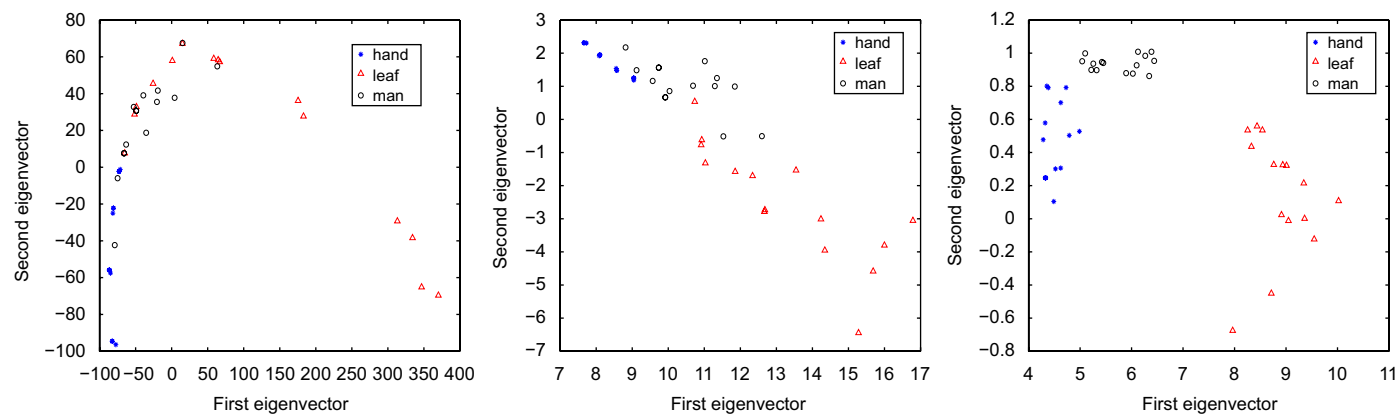


Fig. 3. Shock tree clustering—left (spectral eigenvalue method), middle (spectral polynomial method) and right (our method).

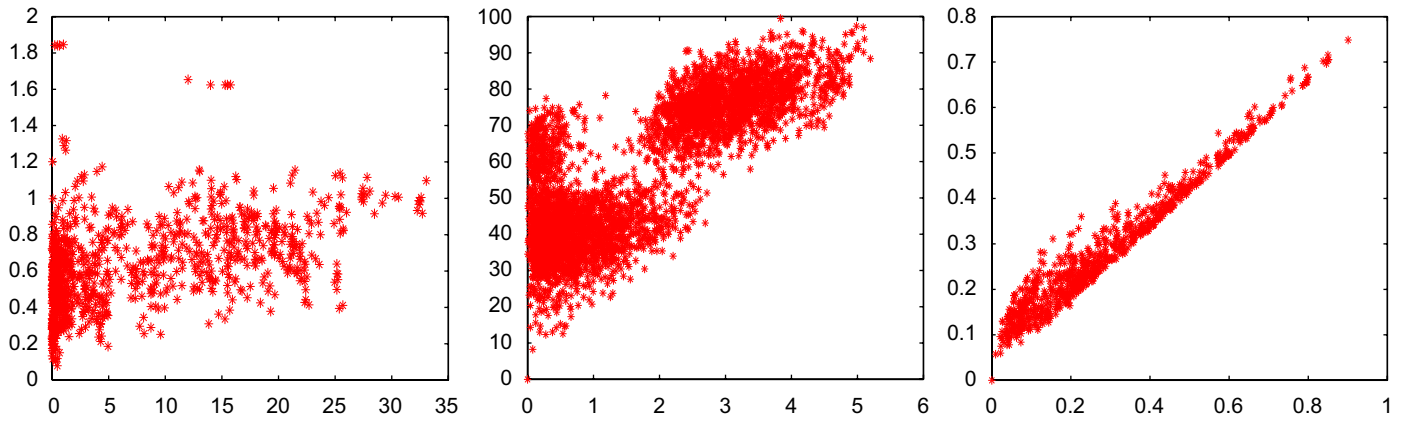


Fig. 4. Scatter plot for different clustering method against the edit distance (y-axis)—left (spectral eigenvalue method), middle (spectral polynomial) and right (our method).

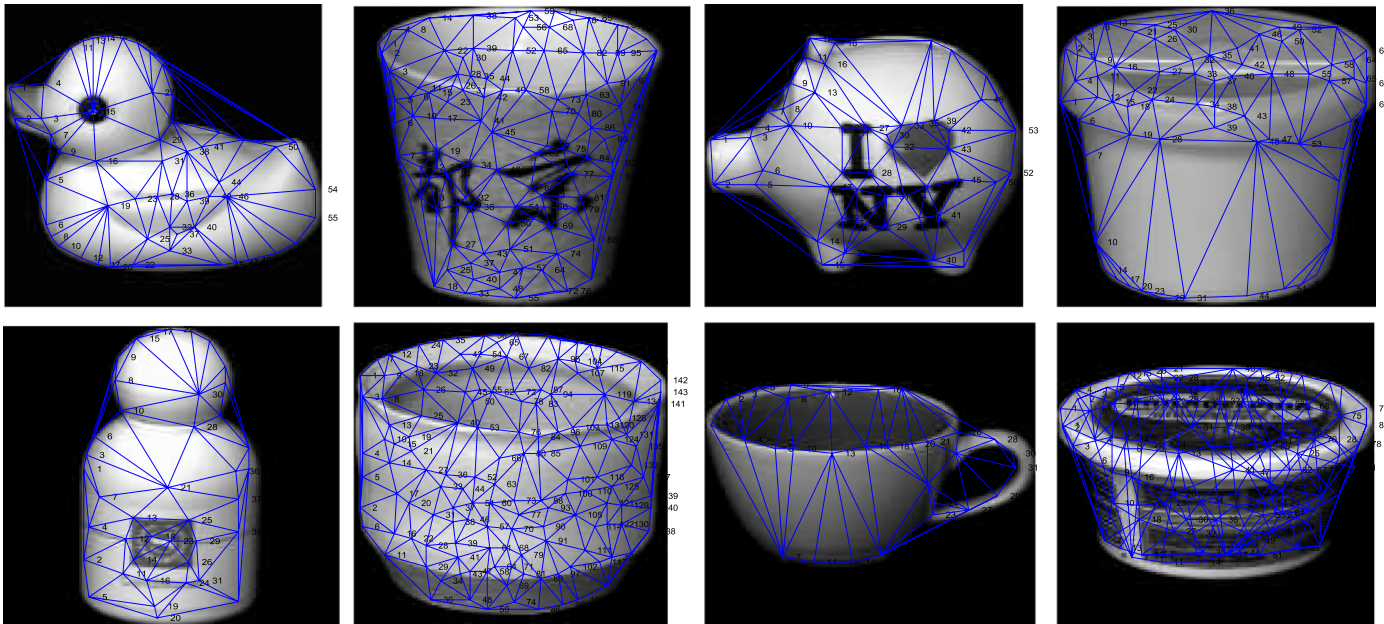


Fig. 5. Eight objects with their Delaunay graphs overlaid.

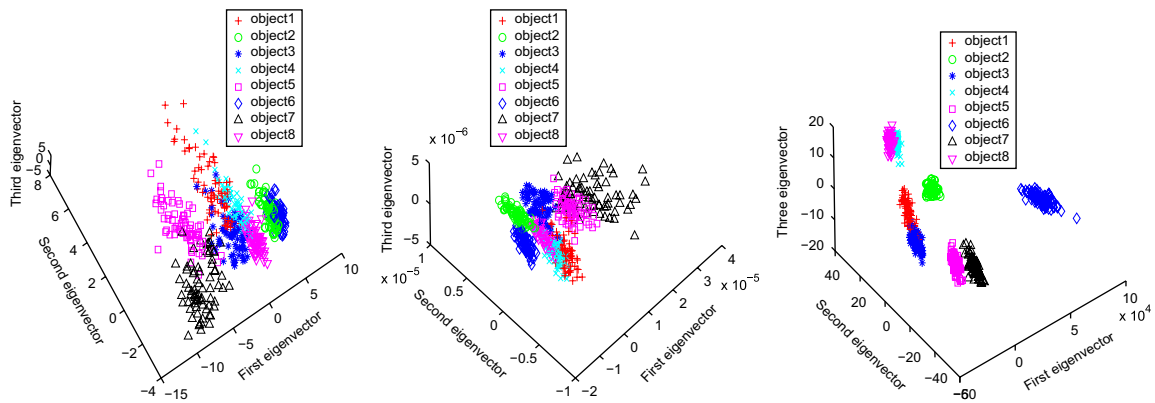


Fig. 6. COIL database clustering—left (direct spectral method) and right (our method).

visualization purpose we project the graph into a three-dimensional space. In the plots, the different shapes of a particular class are represented by symbols of different color. We also computed the

Rand index value for these three methods. The spectral eigenvalue method for COIL database clustering rand index value is 0.78, the spectral polynomial method is 0.86 and our method is 0.96.

4. Conclusion and future work

This paper has presented an efficient approach for tree (graph) clustering. The approach first use the Isomap algorithm to embed the trees in a Euclidean space by using the geodesic distance between nodes. From the arrangement of the nodes in this space, we compute a weighted proximity matrix, and from the proximity matrix a Laplacian matrix is computed. The spectrum of the Laplacian matrix for the embedded graphs may be used for purposes of comparing trees and for clustering them. Our algorithm can lift the co-spectrality of the trees and can keep the lowest distortion produced by the metric embedding.

In our experiments we have proved the feasibility of our algorithm and applied our algorithm to shock trees and COIL database. Our algorithm can also be applied to other types of graphs. We plan to extend our work in several directions. These are: first to experiment with different shock representations, second to apply this algorithm to the graphs and trees for solving the matching problems and third to exploit the feasibility of using the embedded vector representations for indexing purposes.

Another interesting extension of the work reported in this paper would be to explore the use of tensor representations for graphs. The apparatus for this kind of representation has been developed in the mathematics literature [18], and the spectra of tensor products have been studied in some detail [8]. Recently, tensor factorization has been used for multiway-clustering [24], while Tao et al. [29,28] have developed a tensor framework for discriminative learning.

References

- [1] A.D. Alexandrov, V.A. Zalgaller, Intrinsic geometry of surfaces, Transl. Math. Monogr (1967).
- [2] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Neural Inf. Process. Systems 14 (2002) 634–640.
- [3] P. Botti, R. Merris, Almost all trees share a complete set of inmanantal polynomials, J. Graph Theory 17 (1993) 467–476.
- [4] J. Bourgain, On Lipschitz embedding of finite metric spaces into hilbert space, Isreal J. Math. 52 (1985) 46–52.
- [5] H. Busemann, The Geometry of Geodesics, Academic Press, New York, 1955.
- [6] F.R.K. Chung, Spectral Graph Theory, American Mathematical Society, Providence, RI, 1997.
- [7] T. Cox, M. Cox, Multidimensional Scaling, Chapman and Hall, London, 1994.
- [8] C.D. Godsil, B.D. McKay, Constructing cospectral graphs, Aequationes Mathematicae 25 (1982) 257–268.
- [9] A. Gupta, Embedding tree metrics into low dimensionality euclidean spaces, in: Annual ACM Symposium on Theory of Computing, vol. 24 (1), 1999, pp. 106–116.
- [10] C.G. Harris, M.J. Stephens, A combined corner and edge detector, in: Fourth Alvey Vision Conference, 1994, pp. 147–151.
- [11] G.R. Hjaltason, H. Samet, Properties of embedding methods for similarity searching in metric spaces, Pattern Anal. Mach. Intell. 25 (2003) 530–549.
- [12] S. Ioffe, D.A. Forsyth, Human tracking with mixtures of trees, ICCV (2001) 690–695.
- [13] I.T. Jolliffe, Principal Component Analysis, Springer, Berlin, 1986.
- [14] Y. Keselman, A. Shokoufandeh, M. Fatih Demirci, S. Dickinson, Many-to-many graph matching via metric embedding, CVPR 1 (2003) 850–857.
- [15] B.B. Kimia, A.R. Tannenbaum, S.W. Zucker, Shapes, shocks, and deformations, Int. J. Comput. Vision 15 (1995) 189–224.
- [16] N. Linial, E. London, Y. Rabinovich, The geometry of graphs and some its algorithmic application, Combinatorica 15 (1995) 215–245.
- [17] B. Luo, R.C. Wilson, E.R. Hancock, Spectral embedding of graphs, Pattern Recognition 36 (2003) 2213–2230.
- [18] R. Merris, K. Rebman, W. Watkins, Permanent polynomials of graphs, Linear Algebra Appl. 38 (1981) 273–288.
- [19] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (coil 100), Columbia University, 1996.
- [20] W.M. Rand, Objective criteria for the evaluation of clustering, J. Am. Statist. Assoc. 66 (1971) 846–850.
- [21] A. Ranicki, Algebraic I-Theory and Topological Manifolds, Cambridge University Press, Cambridge, 1992.
- [22] S. Sarkar, K.L. Boyer, Quantitative measures of change based on feature organization, in: Proceedings of the International Conference on Computer Vision, 1996, pp. 478–486.
- [23] A. Schwenk, Almost all Trees are Cospectral, Academic Press, New York, 1973, pp. 275–307.
- [24] A. Shashua, R. Zass, T. Hazan, Multi-way clustering using super-symmetric non-negative tensor factorization, in: Proceedings of the European Conference on Computer Vision, 2006, pp. 595–608.
- [25] A. Shokoufandeh, S. Dickinson, K. Siddiqi, S. Zucker, Indexing using a spectral encoding of topological structure, CVPR 20 (1999) 491–497.
- [26] A. Shokoufandeh, S. Dickinson, K. Siddiqi, S. Zucker, Indexing using a spectral encoding of topological structure, in: International Conference on Computer Vision and Pattern Recognition, 1999, pp. 491–497.
- [27] K. Siddiqi, A. Shokoufandeh, S. Dickinson, S. Zucker, Shock graphs and shape matching, Int. J. Comput. Vision 30 (1999) 1–24.
- [28] D. Tao, X. Li, X. Wu, S.J. Maybank, Human carrying status in visual surveillance, in: International Conference on Computer Vision and Pattern Recognition, 2006, pp. 1670–1677.
- [29] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1700–1715.
- [30] J.B. Tenenbaum, V.D. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 586–591.
- [31] A. Torsello, E.R. Hancock, A skeletal measure of 2d shape similarity, Comput. Vision Image Understanding 95 (1) (2004) 1–29.
- [32] A. Torsello, A. Robles-Kelly, E.R. Hancock, Discovering shape classes using tree edit-distance and pairwise clustering, Int. J. Comput. Vision 72 (3) (2007) 259–285.
- [33] S. Weinberger, Review of algebraic I-theory and topological manifolds by A. Ranicki, BAMS 33 (1996) 93–99.
- [34] C.K.I. Williams, N.J. Adams, Dynamic trees, Neural Inf. Process. Systems 11 (1999) 634–640.
- [35] R.C. Wilson, B. Luo, E.R. Hancock, Pattern vectors from algebraic graph theory, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2003) 2220–2237.



Xiao Bai received the B.Eng. degree in Computer Science from Beihang University of China in 2001. From 2002 to 2006, he was a Ph.D. student at Computer Science Department, University of York, UK under the supervision of Professor Edwin Hancock.

He is now a research officer (Fellow, Scientist) in Computer Science Department, University of Bath. He has published more than 10 papers in journals and refereed conferences. His current research interests include graph spectral analysis, image segmentation and recognition, statistical pattern recognition, and feature extraction.



Andrea Torsello received the “Laurea” degree with honors in Computer Science from Ca’ Foscari University of Venice, Italy, in 1997. In 2004 he received his Ph.D. in Computer Science at the University of York, UK. Currently he is an assistant professor at “Ca’ Foscari” University of Venice, Italy.

His research interests are in the area of computer vision and pattern recognition, in particular, the application of stochastic and structural approaches to shape analysis. Recently, he co-edited a special issue of Pattern Recognition on “Similarity-based pattern recognition.”



Edwin R. Hancock studied physics as an undergraduate at the University of Durham and graduated with honours in 1977. He remained at Durham to complete a Ph.D. in the area of high energy physics in 1981. Following this he worked for 10 years as a researcher in the fields of high-energy nuclear physics and pattern recognition at the Rutherford-Appleton Laboratory (now the Central Research Laboratory of the Research Councils). During this period he also held adjunct teaching posts at the University of Surrey and the Open University. In 1991 he moved to the University of York as a lecturer in the Department of Computer Science. He was promoted to Senior Lecturer

in 1997 and to Reader in 1998. In 1998 he was appointed to a Chair in Computer Vision. Professor Hancock now leads a group of some 20 faculty, research staff and Ph.D. students working in the areas of computer vision and pattern recognition. He has supervised about 30 successful PhD theses. His main research interests are in the use of optimization and probabilistic methods for high and intermediate level vision. He is also interested in the methodology of structural and statistical pattern recognition. He is currently working on graph-matching, shape-from-X, image databases, and statistical learning theory. His work has found applications in areas such as radar terrain analysis, seismic section analysis, remote sensing, and medical imaging. Professor Hancock has published some 120 journal papers and 400 refereed conference publications on these topics. He was awarded the Pattern Recognition Society medal in 1991 and an outstanding paper award in 1997 by the journal Pattern Recognition. He also has best paper prizes from CAIP 2001, ACCV 2001, ICPR 2006 and BMVC 2007. In 1998 he became a fellow of the International Association for Pattern Recognition. He is also a Fellow of the Institute of Physics,

the IET and the British Computer Society. Professor Hancock has been a member of the Editorial Boards of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, and Pattern Recognition. He has also been a guest editor for special editions of the journals Image and Vision Computing and Pattern Recognition. He has been on the programme committees for numerous national

and international meetings. In 1997 with Marcello Pelillo, he established a new series of international meetings on energy minimization methods in computer vision and pattern recognition. He has chaired BMVC, been track chair at ICPR and area chair for ECCV and CVPR.