Spanning Tree Recovery via Random Walks in a Riemannian Manifold

Antonio Robles-Kelly and Edwin R. Hancock

Department of Computer Science,
University of York, York YO1 5DD, UK
{arobkell,erh}@cs.york.ac.uk

Abstract. In this paper, we describe the use of Riemannian geometry and graph-spectral methods for purposes of minimum spanning tree recovery. We commence by showing how the sectional curvature can be used to model the edge-weights of the graph as a dynamic system in a manifold governed by a Jacobi field. With this characterisation of the edge-weights at hand, we proceed to recover an approximation for the minimum spanning tree. To do this, we present a random walk approach which makes use of a probability matrix equivalent, by row-normalisation, to the matrix of edge-weights. We show the solution to be equivalent, up to scaling, to the leading eigenvector of the edge-weight matrix. We approximate the minimum spanning tree making use of a brushfire search method based upon the rank-order of the eigenvector coefficients and the set of first-order neighbourhoods for the nodes in the graph. We illustrate the utility of the method for purposes of network optimisation.

1 Introduction

The recovery of the minimum spanning tree is a classical problem in pattern recognition. For a weighted graph, the minimum spanning tree is the set of edges that connects all its vertices without cycles and with minimum total length. The minimum spanning tree finds application in a number of areas such as network optimisation, database indexing and logistic planning. The problem is clearly one of optimisation, which is traditionally solved using greedy algorithms [1, 2].

However, one of the methods that has received little attention is that of posing the problem in an energy minimisation setting and using graph-spectral techniques to recover the solution. To cast the problem in such way has a number of advantages. Firstly, it would allow the modeling of processes that occur in arbitrary dimension under non-linear constraints. It also allows the problem to be modeled as a conservative process in a manifold. Viewed in this manner, the edge-weight then becomes the energy required to move between a pair of adjacent nodes in the graph. Once the edge-weights are at hand, the apparatus of graph-spectral theory can be used to recover the tree whose cost is optimum.

Graph-spectral methods have recently proved highly effective in image processing and computer vision. By computing the eigenvalues and eigenvectors of

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 303–311, 2004.

the weight matrix, it is possible to find groups or clusters of entities. Perhaps the best known method is that of Shi and Malik [3] which has shown how to locate image regions by recursively locating the eigenvector associated with the second smallest eigenvalue of the Laplacian matrix, i.e. the degree matrix minus the affinity weight matrix. Although it is convenient to work with the Laplacian, since it is positive semi-definite, grouping and segmentation can also be performed using an edge-weight or affinity matrix. For instance, both Sarkar and Boyer [4] and Perona and Freeman [5] have developed matrix factorisation methods for line-segment grouping that use eigenvectors of an affinity matrix rather than the associated Laplacian. The Sarkar and Boyer [4] method can be understood as that which maximises the total edge weight of the clusters.

The methods described above all share the feature of using the eigenvectors of a Laplacian or an affinity matrix to define groups or clusters or objects. However, graph-spectral methods can also be used for path analysis tasks on graphs. For instance, it is well known that the path length distribution can be computed from the spectrum of eigenvalues of the adjacency matrix [6]. Ideas from spectral-graph theory have also been used to analyse the behaviour of random walks in graphs [7–9]. In addition, there are important relationships between the eigenvectors of the edge weight matrix and other quantities related to random walks. Further, the relationship between the leading eigenvector of the edge weight matrix and the steady state random walk has been exploited in a number of areas including routeing theory and information retrieval [10, 11].

The advantage of graph-spectral methods is that they can be used to find approximate or relaxed solutions without the need for parallel iterative updates at the vertex level. The method also obviates the need for complex search algorithms. However, although they have been applied to region segmentation and grouping problems, graph-spectral methods have not been applied to curve detection problems of the sort that arise in the determination of the optimal spanning tree.

2 Riemannian Geometry and Markovian Processes

In this section, we provide the theoretical basis for our minimum spanning tree approximation algorithm. We commence by showing how Riemannian geometry invariants can be used to model the edge-weights of the graph as the energy of a particle moving along a geodesic between a pair of points in a Riemannian manifold. There are clearly a number of ways in which the energy of such dynamic system can be minimised. Here, we have chosen to present a random walk approach that makes use of the apparatus of Markov chains to approximate the tree whose cost is minimum.

2.1 Riemannian Manifolds

In this section, we aim at providing a means of characterising the step between two adjacent nodes in the graph as a geodesic in a Riemannian manifold. To do this, we pose the problem in a graph-based setting. Let $G = (V, E, W)$ denote a weighted graph with index-set V, edge-set $E = \{(i,j)|(i,j) \in V \times V, i \neq j\}$ and edge-weight set $W = \{W(i,j)|(i,j) \in V \times V \wedge W(i,j) \in [0,1]\}$. If the nodes in the graph are viewed as points on the manifold, the edge weight between a pair of nodes can then be interpreted as the energy of a particle moving along the geodesic under the influence of a vector field. With these ingredients, the energy $\mathcal{E}_{p_i,p_j}$ of the dynamic system can be used to define the elements of the edge-weight matrix such that $W(i,j) = \exp[-\mathcal{E}_{p_i,p_j}]$.

To express the energy $\mathcal{E}_{p_i,p_j}$ in terms of geometric invariants, we employ the theory of Jacobi vector fields and their relation to the curvature tensor to characterise the sectional curvature of the manifold. The reasons for using the curvature tensor are twofold. Firstly, the curvature tensor is natural, i.e. invariant under isometries. Secondly, the curvature tensor can be defined intrinsically through coordinate changes and it appears as the second differential form of the metric. Hence, the curvature tensor is one of the main invariants in Riemannian geometry. Here, unless noted otherwise, we use tensor notation in order to provide a framework which is both, consistent with the material available from mathematics and compatible with concepts drawn from physics. We consider a function f to be differentiable if it is of class C^∞, i.e. all its partial derivatives, of all orders, exist and are continuous.

Consider a n-dimensional differentiable manifold M. For any point $p \in M$, let M_p denote the tangent space of M at p. Further, let Y be a differentiable vector field in $\Re^n$ such that $Y = \sum_{i=1}^n \eta^i \partial_i$, where η^i is the ith coordinate of the vector $\eta = \sum_{i=1}^n \eta^i \partial_i|_p$ and $\mathfrak{e} = \{\mathfrak{e}_1, \mathfrak{e}_2, \ldots, \mathfrak{e}_n\}$ is the natural basis $(\Re^n)_p$, i.e. the natural basis of $\Re^n$ at $p \in M$. In the equations above, the symbol ∂_i has been defined so as to be consistent with both, the notion of chart in Riemannian geometry and the natural basis $\mathfrak{e}$. To provide a definition of ∂_i, we turn our attention to the natural identification $\Im_p : \Re^n \mapsto (\Re^n)_p$ of the tangent space at p, i.e. $(\Re^n)_p$, onto $\Re^n$. For the natural basis, the chart is then given by the identity map such that $\partial_i|_p = \Im_p \mathfrak{e}_i$.

In order to take our analysis further, we require a well-defined method for differentiating vector fields. Hence, for a collection of vector fields $\wp^1$ of class C^1 and a differentiable vector $\xi \in M_p$, the connection $\nabla : M_p \times \wp^1(M_p) \mapsto M_p$ is given by $\nabla_\xi Y = \sum_{i=1}^n (\xi \eta^i) \partial_i$. This definition implies that the vector $\nabla_\xi Y$ is in the same tangent space as ξ. Furthermore, the connection expresses the covariant derivatives of the vector field Y in terms of the vector ξ. This is, $\nabla_\xi Y$ describes the rate of change of the vector field Y in the direction ξ in terms of ξ itself.

In this section, we aim to characterise the transition between two nodes in the graph as a dynamic system in a manifold. To provide a characterisation invariant over isometric transformations, we use the notion of connection provided above to define the curvature tensor. Consider the vector fields Y, X and Z to be extensions in a neighbourhood of p of the vectors $\eta, \xi, \zeta \in M_p$. The curvature tensor, which is quadrilinear in nature [12], is then denoted by $R(\xi, \eta)\zeta$. Here, we are interested in the sectional curvature, which is bilinear in nature. To obtain a bilinear form, i.e. the sectional curvature, from the curvature tensor we use two linearly independent vectors $\eta, \xi \in M_p$ and write

$$\mathcal{K}(\xi,\eta) = \frac{\langle R(\xi,\eta)\xi,\eta\rangle}{\mid \xi \mid^2 \mid \eta \mid^2 - \langle \xi,\eta\rangle} \tag{1}$$

As mentioned earlier, we are interested in modeling the edges in the graph as geodesics in a manifold. Consider the parameterised curve $\gamma : t \in [\alpha, \beta] \mapsto M$. From Riemannian geometry, we know that for γ to be a geodesic, it must satisfy the condition $\nabla_{\gamma'}\gamma' = 0$. It can be shown that the connection ∇ for geodesics is, in fact, a Levi-Civita connection [12]. Further, Levi-Civita connections are metric preserving, unique and are guaranteed to exist.

To take our analysis further, we define the Jacobi field along γ as the differentiable vector field $Y \in M_p$, orthogonal to γ, satisfying Jacobi's equation $\nabla_t^2 Y + R(\gamma', Y)\gamma' = 0$, where ∇ is a Levi-Civita connection.

With this ingredients, we can substitute ξ and η with γ' and Y in Equation 1 and write $\mathcal{K}(\gamma', Y) = \frac{\langle R(\gamma',Y)\gamma',Y\rangle}{|\gamma'|^2|Y|^2-\langle\gamma',Y\rangle}$. But, because Y is orthogonal to γ', the equation above becomes

$$\mathcal{K}(\gamma',Y) = \frac{\langle R(\gamma',Y)\gamma',Y\rangle}{\mid \gamma' \mid^2 \mid Y \mid^2} \tag{2}$$

To simplify the expression for the sectional curvature further, we make use of the fact that, since Y is a Jacobi field, it must satisfy the condition $\nabla_t^2 Y = -R(\gamma', Y)\gamma'$. Hence, we write $\mathcal{K}(\gamma', Y) = \frac{\langle -\nabla_t^2 Y, Y\rangle}{\langle Y, Y\rangle}$, where we have substituted $\mid Y \mid^2$ with $\langle Y, Y\rangle$ and set $\mid \gamma' \mid = 1$. As a result, it follows that $\nabla_t^2 Y = -\mathcal{K}(\gamma', Y)Y$.

This suggests a way of formulating the energy along the geodesic $\gamma \in M$ connecting the pair of points indexed i and j. Consider a particle of mass ρ moving along the geodesic γ subject to the Jacobi field Y. The energy of the particle can be expressed making use of the equations above as

$$\mathcal{E}_{p_i,p_j} = \rho\bigg(\int_\gamma \mid \gamma' + \nabla_t^2 Y \mid^2 dt\bigg) = \rho\bigg(\int_\gamma \mid \gamma' - \mathcal{K}(\gamma',Y)Y \mid^2 dt\bigg) \tag{3}$$

where p_i is the point indexed i in M. This is, we have expressed the energy of the particle moving from the point indexed i to the point indexed j as the sum of its kinetic energy and the potential contributed by the Jacobi field along γ. Hence, the edge-weight is small if a pair of points are far from each other or the curvature along the geodesic between them is large.

2.2 Random Walks

To take our analysis further and make the relationship to the sectional curvature more explicit, we cast the problem into a random walk setting. In order to profit from a Markov chain approach to the problem, we commence by row-normalising the weight matrix W so its rows sum to unity. To do this, we compute the degree of each node $deg(i) = \sum_{j=1}^{|V|} W(i,j)$. With the diagonal degree matrix $D = diag(deg(1), deg(2), \ldots, deg(|V|))$ at hand, the transition probability matrix is given by $P = D^{-1}W$. The elements of the transition matrix are hence given by $P_{i,j} = \frac{1}{deg(i)} W_{i,j}$. It is interesting to note that the transition matrix P is a row stochastic matrix. Moreover, it is related to the normalised symmetric positive definite matrix $\hat{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = D^{\frac{1}{2}} P D^{-\frac{1}{2}}$. As a result, we can write $P = D^{-\frac{1}{2}} \hat{W} D^{\frac{1}{2}}$. It is worth noting in passing that the matrix $\hat{W}$ is related to the normalised Laplacian $L = D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = I - \hat{W}$.

Our aim is to use the steady state random walk on the graph G for purposes of recovering the spanning tree whose cost is minimum. The walk commences at the node j_1 and proceeds via the sequence of nodes $\Upsilon = \{j_1, j_2, j_3, ...\}$. If the random walk can be represented by a Markov chain with transition matrix P, then the probability of visiting the nodes in the sequence above is

$$P_\Upsilon = P(j_1) \prod_{l \in \Upsilon} P_{j_{l+1}, j_l} = \prod_{l \in \Upsilon} \frac{W_{j_{l+1}, j_l}}{deg(l)} \tag{4}$$

Substituting for the path energy, we have that

$$P_\Upsilon = \frac{\exp\left[-\rho \sum_{l \in \Upsilon} \left(\int_\gamma \mid \gamma' - \mathcal{K}(\gamma', Y) Y \mid^2 dt \right)\right]}{\prod_{l \in \Upsilon} deg(l)} = \frac{1}{Z_\Upsilon} \exp[-\mathcal{E}_\Upsilon] \tag{5}$$

where $\mathcal{E}_\Upsilon = \rho \sum_{l \in \Upsilon} \left(\int_\gamma \mid \gamma' - \mathcal{K}(\gamma', Y) Y \mid^2 dt \right)$ and $Z_\Upsilon = \prod_{l \in \Upsilon} deg(l)$.

Hence, the path is a Markov chain with energy function $\mathcal{E}_\Upsilon$ and partition function Z_Υ. Further, let $Q_k(i)$ be the probability of visiting the node indexed i after k-steps of the random walk and let $Q_k = [Q_k(1), Q_k(2), \ldots]^T$ be the vector whose components are the probabilities of visiting the nodes at step k. After k steps we have that $Q_k = P^k Q_0$. If $\hat{W}^k$ is the result of multiplying the symmetric positive definite matrix $\hat{W}$ by itself k times, then $P^k = D^{-\frac{1}{2}} \hat{W}^k D^{\frac{1}{2}}$.

To develop a spectral method for locating the steady state random walk, we turn to the spectral decomposition of the normalised affinity matrix $\hat{W}$ $\hat{W} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = \sum_{i=1}^{|V|} \lambda_i \phi_i \phi_i^T$, where the λ_i are the eigenvalues of $\hat{W}$ and the ϕ_i are the corresponding eigenvectors. By constructing the matrix $\Phi = (\phi_1 | \phi_2 | \ldots | \phi_{|V|})$ with the eigenvectors of $\hat{W}$ as columns and the matrix $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ with the eigenvalues as diagonal elements, we can write the spectral decomposition in the more compact form $\hat{W} = \Phi \Lambda \Phi^T$. Since, the eigenvectors of $\hat{W}$ are orthonormal, i.e. $\Phi \Phi^T = I$, we have that $\hat{W}^k = \Phi \Lambda^k \Phi^T$.

Recall that the leading eigenvalue of $\hat{W}$ is unity. Furthermore, from spectral graph theory [8], provided that the graph G is not a bipartite graph, then the smallest eigenvalue $\lambda_{|V|} > -1$. As a result, when the Markov chain approaches its steady state, i.e. $k \to \infty$, then all but the first term in the above series become negligible. Hence, the steady state random walk is given by $Q_s = \lim_{k \to \infty} Q_k = D^{\frac{1}{2}} \phi_* \phi_*^T D^{-\frac{1}{2}} Q_0$. This establishes that the leading eigenvector of the normalised affinity matrix $\hat{W}$ determines the steady state of the random walk. It is also important to note that the equilibrium equation for the Markov process is $Q_s = PQ_s$, where Q_s is the vector of steady-state site visitation probabilities. Hence, since the leading eigenvalue of P is unity, then it follows that Q_s is the leading eigenvector of P. For a more complete proof of this result see the book by Varga [13] or the review of Lovasz [7].

We aim to visit the points in the manifold in the order of their steady-state state probabilities. Suppose that the initial state vector for the sites is uniform, i.e. $Q_0 = (\frac{1}{|V|}, \ldots, \frac{1}{|V|})^T$. As a result, the steady-state probability of visiting the node indexed i is

$$Q_s(i) = \frac{1}{|V|} \sum_{j=1}^{|V|} \sqrt{\frac{deg(j)}{deg(i)}} \phi_*(i)\phi_*(j) = \frac{1}{|V|} \frac{\phi_*(i)}{\sqrt{deg(i)}} \sum_{j=1}^{|V|} \sqrt{deg(j)} \phi_*(j) \tag{6}$$

Since the summation appearing above is the same for all the vertices in the graph, the probability rank order is determined by the quantity $\psi_*(i) = \frac{\phi_*(i)}{\sqrt{deg(i)}}$.

We can make the relationship to the energy of the path more explicit by expanding the expressions above and substituting the energy functional for every element of the matrix W. Further, if the mass of the particle is small, we can make use of the Maclaurin expansion for the exponential weighting function and write $\hat{W}(i,j) \simeq \frac{1}{\sqrt{deg(i)deg(j)}}(1 - \rho[\int_\gamma \mid \gamma' - \mathcal{K}(\gamma', Y)Y \mid^2 dt])$. Hence, the path can be shown to be the one that satisfies the condition

$$\phi_* = \arg\min_{\Phi} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \left(\frac{\phi(i)\phi(j)}{\sqrt{deg(i)deg(j)}} \int_\gamma \mid \gamma' - \mathcal{K}(\gamma', Y)Y \mid^2 dt \right) \tag{7}$$

As a result, the integration path will minimise both, the sectional curvature along the geodesic and the length of the geodesic itself.

Our aim is to use the probability rank order to recover the spanning tree for which the cost of traversing its branches is minimum. If we visit the nodes in the order defined by the magnitudes of the coefficients of the leading eigenvector of the normalised affinity matrix, then the path is the steady state of the Markov chain. Unfortunately, the path followed by the steady state random walk is not edge-connected. Hence, we need a means of placing the nodes in an order in which neighbourhood connectivity constraints are preserved using the elements of the scaled leading eigenvector ψ_*.

3 Recovering the Spanning Tree

The idea underpinning our spanning tree recovery algorithm is to use the rank-order provided by the components of the leading eigenvector to locate those graph-edges that correspond to branches of the spanning tree. We pose this as a brushfire search which is driven from the rank order of the nodes in the data. In a nutshell, the idea is to traverse the rank-ordered list of graph nodes, commencing with the node of largest coefficient and terminating with the node of

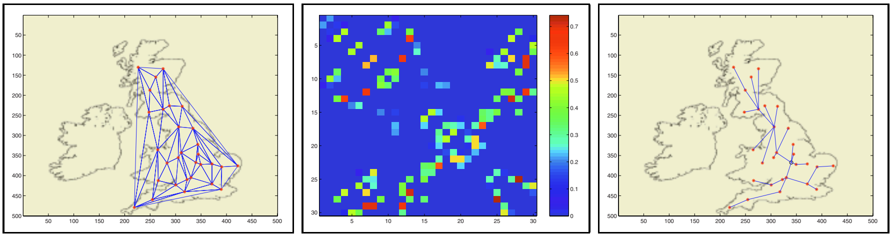

Fig. 1. From left-to-right: Delaunay triangulation corresponding to 30 cities in the UK; Corresponding matrix W; Approximation to the minimum spanning tree delivered by our algorithm.

smallest coefficient. To do this, we use the rank-order of the nodes in the graph, which is given by the list of sorted node-indices $O_D = (j_1, j_2, j_3, ..., j_{|V_D|})$ where $\psi_*(1) > \psi_*(2) > \psi_*(3) > ... > \psi_*(|V|)$. The subscript i of the node-index $j_i \in V$ is hence the rank-order of the coefficient $\psi_*(i)$.

We commence by placing the first ranked node as the root of the spanning tree, which we label j_1. We proceed with our brush-fire search by considering the list of first-neighbours $\mathcal{N}_{j_1}$ for the root node j_1. The candidates which may be assigned to the node j_1 must satisfy the edge-connectivity constraints provided by the graph-edges. These candidates are the first-neighbours of the root j_1. We rank the nodes in the list $\mathcal{N}_{j_1}$ according to the coefficients of the vector ϕ_* and propagate this procedure by visiting each node in the graph in the order specified by the ranked-list O_D. This is an iterative process which spreads like a brush-fire from the root node j_1.

To keep track of the nodes visited we maintain a list $\mathcal{L}$ of the nodes that have already been assigned to a branch of the tree. Suppose that we have reached the n^{th} ranked node which is not in the list $\mathcal{L}$, i.e. $j_n \notin \mathcal{L}$, such that $\mathcal{L} \cap \mathcal{N}_{j_n} \neq \emptyset$. The algorithm proceeds as follows. We find the set of first-neighbours of the node j_n which are in the list $\mathcal{L}$. We would like to preserve edge-connectivity constraints while assigning the node ranked n as a leaf of the node $j_i \in \mathcal{L}$ on the basis of the rank-order of the coefficients of the scaled leading eigenvector ψ_*. Hence, we assign the node j_n as a leaf of the node j_i so as to satisfy the condition $j_i = \{j_l \mid \psi_*(l) = \max_{j_l \in \{\mathcal{N}_{j_n} \cap \mathcal{L}\}} \{\psi_*(l)\}\}$. This process is repeated until all of the nodes in the graph have been assigned to a branch of the spanning tree, i.e. $\mathcal{L} = V$.

4 Experiments

In this section, we illustrate the utility of our method for purposes of network optimisation. Our experimental vehicle is a distribution network in the United Kingdom (UK). For this purpose, we have used a set of 30 points drawn from city locations in the UK. We do this by making use of the postcodes for 30 cities to locate points, by longitude and latitude, on a map of the UK. Our graph is then given by the Delaunay triangulation of these points.

To compute the edge-weights, we have done the following. Since the earth can be considered to be a sphere, we have set the sectional curvature to the constant κ, i.e. $\mathcal{K}(\gamma', Y) \equiv \kappa$. It can be shown that, for the special case of constant, positive sectional curvature, Jacobi's equation becomes $\nabla_t^2 Y = -\kappa Y$ and its solution, given $Y(0) = 0$ and $\mid \nabla_t Y(0) \mid = 1$, is $Y(t) = \frac{\sin(\sqrt{\kappa}t)}{\sqrt{\kappa}}\eta$, where the vector η is in the tangent space of M at p_i and is orthogonal to γ' at the point indexed i, i.e. $\eta \in M_{p_i}$ and $\langle \eta, \gamma' \mid_{p_i} \rangle = 0$.

With these ingredients, and by rescaling the parameter t so that $\mid \gamma' \mid = \tau$, we can express the cost of the step between nodes indexed i and j as follows

$$\mathcal{E}_{p_i,p_j} = \int_0^a \left((\tau)^2 + \kappa \left(\sin(\sqrt{\kappa}t) \right)^2 \right) dt \tag{8}$$

In our experiments, we have set κ to unity and rescaled τ so as to take into account the circumference of the earth and satisfy the condition $\mid \gamma' \mid = \tau$. As a result, the value of a is given by the arc length of the geodesic between the points p_i and p_j.

In Figure 1, we show the results obtained by our algorithm. In the left-hand panel, we show the Delaunay graph for the 30 cities used in our experiments. The middle panel shows the edge-weight matrix, i.e. the matrix W for the Delaunay graph in the left-hand panel. Finally, the right-hand panel shows the spanning tree recovered by our algorithm. Here, we have indicated the root of the tree with a circle, while the rest of the points have been plotted using asterisks.

5 Conclusions

In this paper, we have described how geometric invariants and random walks can be used to pose the recovery of the minimum spanning tree in an energy minimisation setting. The work described here can be further extended and improved in a number of different ways. Firstly, there is clearly scope for developing a more sophisticated method for the brushfire search process. It may also be interesting to investigate whether the ideas presented here can be applied to 2D curve enhancement problems.

References

1. J. B. Kruskal. On the shortest spanning subtree of a graph and the travelling salesman problem. *Prof. of the American Mathematical Society*, 7:48–50, 1956.
2. R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.
3. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
4. S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71(1):110–136, 1998.

5. P. Perona and W. T. Freeman. Factorization approach to grouping. In *Proc. ECCV*, pages 655–670, 1998.
6. N. L. Biggs. *Algebraic Graph Theory.* Cambridge University Press, 1993.
7. L. Lovász. Random walks on graphs: a survey. *Bolyai Society Mathematical Studies*, 2(2):1–46, 1993.
8. Fan R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997.
9. D. Cvetković, M. Doob, and H. Sachs. *Spectra of Graphs:Theory and Application.* Academic Press, 1980.
10. Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *ACM Symposium on Theory of Computing*, pages 619–626, 2000.
11. J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. ACM-SIAM symposium on discrete algorithms*, pages 668–677, 1998.
12. I. Chavel. *Riemannian Geometry: A Modern Introduction.* Cambridge University Press, 1995.
13. R. S. Varga. *Matrix Iterative Analysis.* Springer, second edition, 2000.