Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational and Applied Mathematics 204 (2007) 25–37

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Spectral clustering and its use in bioinformatics

Desmond J. Higham[a,*,1], Gabriela Kalna[a,2], Milla Kibble[b,3]

[a] *Department of Mathematics, University of Strathclyde, Glasgow, G1 1XH Scotland, UK*
[b] *Department of Mathematics, University of Turku, FIN-20014 Turku, Finland*

Received 15 July 2005; received in revised form 10 September 2005

Abstract

We formulate a discrete optimization problem that leads to a simple and informative derivation of a widely used class of spectral clustering algorithms. Regarding the algorithms as attempting to bi-partition a weighted graph with N vertices, our derivation indicates that they are inherently tuned to tolerate all partitions into two non-empty sets, independently of the cardinality of the two sets. This approach also helps to explain the difference in behaviour observed between methods based on the unnormalized and normalized graph Laplacian. We also give a direct explanation of why Laplacian eigenvectors beyond the Fiedler vector may contain fine-detail information of relevance to clustering. We show numerical results on synthetic data to support the analysis. Further, we provide examples where normalized and unnormalized spectral clustering is applied to microarray data—here the graph summarizes similarity of gene activity across different tissue samples, and accurate clustering of samples is a key task in bioinformatics.

MSC: 65F15; 92C37

Keywords: Balancing threshold; Gene expression; Rayleigh–Ritz Theorem; Fiedler vector; Graph Laplacian; Random graph; Maximum likelihood; Microarray; Partitioning; Scaling

1. Introduction

Suppose we are given a set of objects, labelled 1, 2, 3, ..., N, and a set of pairwise similarity weights, $\{w_{ij}\}_{i,j=1}^{N}$, with $w_{ij} = w_{ji} \geqslant 0$ and (for convenience) $w_{ii} = 0$. A large weight w_{ij} indicates that objects i and j are very similar. The associated clustering problem is to bundle the objects into groups so that objects within each group are similar and objects in distinct groups are dissimilar. Key application areas include computer imaging, pattern recognition, high-performance scientific computing, sparse matrix computation, multicasting, graph layout, datamining and bioinformatics [1,3,5,7,11,15,21–24,26,27].

Here we focus on one general approach, spectral clustering, that has been invented and re-invented in a number of disciplines, and has been implemented in publically available software, [10]. We begin with some definitions and notation.

* Corresponding author.
E-mail addresses: djh@maths.strath.ac.uk (D.J. Higham), ra.gkal@maths.strath.ac.uk (G. Kalna), milant@utu.fi (M. Kibble).

[1] Supported by Engineering and Physical Sciences Research Council Grant GR/T19100 and by Research Fellowships from The Leverhulme Trust and The Royal Society of Edinburgh/Scottish Executive Education and Lifelong Learning Department.
[2] Supported by Engineering and Physical Sciences Research Council Grant GR/T19100.
[3] Supported by the Academy of Finland, under Grant number 53441.

0377-0427/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2006.04.026

Let $W \in \mathbb{R}^{N\times N}$ denote the symmetric weight matrix, and let the diagonal matrix $D \in \mathbb{R}^{N\times N}$ have $d_i := \sum_{j=1}^{N} w_{ij}$ as its ith diagonal entry. We refer to the matrix $D - W$ as the *Laplacian*, and the matrix $D^{-1/2}(D - W)D^{-1/2}$ as the *normalized Laplacian*. We also take the standard viewpoint of regarding the objects as vertices of an undirected graph, whose edges are weighted according to W.

The Laplacian is symmetric positive semi-definite with smallest eigenvalue 0 and corresponding eigenvector $\mathbf{1}$, the vector with all elements equal to one. We suppose that the graph is connected (in the sense that any pair of vertices may be connected by a path along non-zero weighted edges). This implies that all other eigenvalues of the Laplacian are positive; see, for example, [5,24]. We also suppose that there is a unique smallest positive eigenvalue. We will order the eigenvalues so that $0 = \lambda_1 < \lambda_2 < \lambda_3 \leqslant \cdots \leqslant \lambda_N$ with corresponding mutually orthonormal eigenvectors $\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots, \mathbf{v}^{[N]}$, whence $\mathbf{v}^{[1]} = \mathbf{1}/\sqrt{N}$.

Similarly, the normalized Laplacian is symmetric positive semi-definite with smallest eigenvalue 0 and corresponding eigenvector $D^{1/2}\mathbf{1}$. We suppose the eigenvalues may be ordered so that $0 = \mu_1 < \mu_2 < \mu_3 \leqslant \cdots \leqslant \mu_N$ with corresponding mutually orthonormal eigenvectors $\mathbf{w}^{[1]}, \mathbf{w}^{[2]}, \ldots, \mathbf{w}^{[N]}$, giving $\mathbf{w}^{[1]} = D^{1/2}\mathbf{1}/\|D^{1/2}\mathbf{1}\|_2$. The eigenvalues of the normalized Laplacian satisfy $0 \leqslant \mu_i \leqslant 2$; see, for example, [24].

We refer to $\mathbf{v}^{[2]}$ as the *Fiedler vector* of the Laplacian and to $D^{-1/2}\mathbf{w}^{[2]}$ as the *normalized Fiedler vector* of the normalized Laplacian. Note that the normalized Laplacian is similar to the matrix $D^{-1}(D - W)$ that arises when the Laplacian is diagonally scaled so that the sum of the absolute values across each row is uniform. The normalized Fiedler vector corresponds to the second eigenvector of this matrix. We will be careful to distinguish between unnormalized/normalized versions of the Laplacian and Fiedler vector—across different references in the graph theory literature there is no consistency in the use of the unqualified phrases.

In the unnormalized case, the idea behind spectral clustering is to compute the second eigenvector of the Laplacian, $\mathbf{v}^{[2]}$, and perhaps other small eigenvectors, $\mathbf{v}^{[3]}, \mathbf{v}^{[4]}, \ldots$. In the normalized case, the scaled eigenvectors $D^{-1/2}\mathbf{w}^{[2]}$, $D^{-1/2}\mathbf{w}^{[3]}$, $D^{-1/2}\mathbf{w}^{[4]}, \ldots$ of the normalized Laplacian are used. The information in the eigenvectors then forms the basis for the clustering decisions. The fine details of how the eigenvector information is used vary across the clustering literature—see [2,21,24] for examples—but they are not important for this work. We are interested in the broad question of what the eigenvectors may tell us. We also mention that the Fiedler vector computation itself presents some interesting research issues. The recent code `HSL_MC73`, [14], implements an efficient multilevel algorithm for approximating the Fiedler vector of a very large graph.

To illustrate the spectral approach, we generated the data in Fig. 1. Here, the 100 points in the x, y plane are the objects to be clustered, and the weight w_{ij} is taken to be the reciprocal of the Euclidean distance between the ith and jth points. The first 50 points, marked '$*$', were generated by adding Normal(0, 1) perturbations to the point (5, 5). Similarly, the next 35 points, marked '$\circ$', were formed by adding Normal(0, 1) perturbations to (1, 1) and the final 15 points, marked '$\times$', are Normal(0, 1) shifts about $(-4, -4)$. The three types of marker thus indicate the three 'natural' clusters in this artificially generated data. In Fig. 2 we plot the eigenvectors $\mathbf{v}^{[2]}$ and $\mathbf{v}^{[3]}$ and the scaled, normalized eigenvectors $D^{-1/2}\mathbf{w}^{[2]}$ and $D^{-1/2}\mathbf{w}^{[3]}$. We have used marking consistent with that of the original data: the first 50 components of each vector are marked '$*$', the next 35 are marked '$\circ$' and the final 15 are marked '$\times$'. It is clear from the figure that (a) the eigenvectors do carry information about the existence of clusters, and (b) the unnormalized and normalized versions present this information quite differently. (An explanation of these differences will be given in Section 5.)

Because spectral partitioning has such a diverse range of applications, it is difficult to give a chronological history of its development. The Fiedler vector is so-called because its key properties were identified in [6]. An early reference from a numerical analysis/matrix reordering perspective is [3], which advocated the use of the unnormalized Fiedler vector. The normalized version was proposed in [24]. The computer science/graph theory perspective is represented in [2,16,19,21,23,26]. Clustering is a hard discrete optimization problem, and analysis that backs up spectral clustering is typically limited to

(1) semi-heuristic arguments that justify and help to explain the performance of the algorithm, [3,21,24],
(2) theoretical error analysis that guarantees good results if the data are "sufficiently clusterable", [2,16,19,23,26].

Our work contributes to 1, with emphasis on the effect of normalization, and also gives practical results on an application in bioinformatics.

The main aims of this work are as follows. Sections 2 and 3 present a simple, unified derivation of the spectral algorithms that shows naturally how the normalized and unnormalized versions arise, and explains differences in their

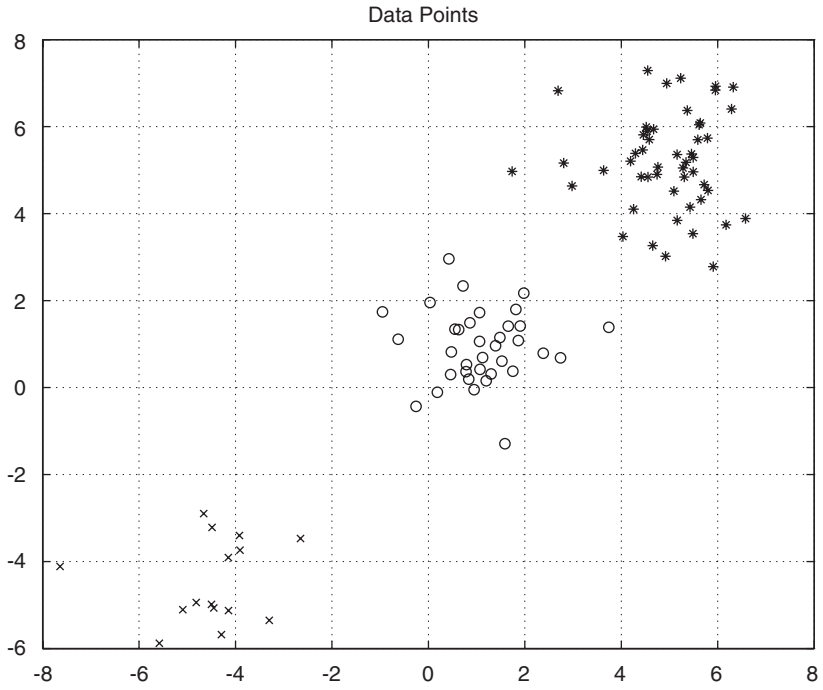

Fig. 1. Data points used for Fig. 2. Pairwise similarity weight is taken to be the reciprocal of the Euclidean distance.

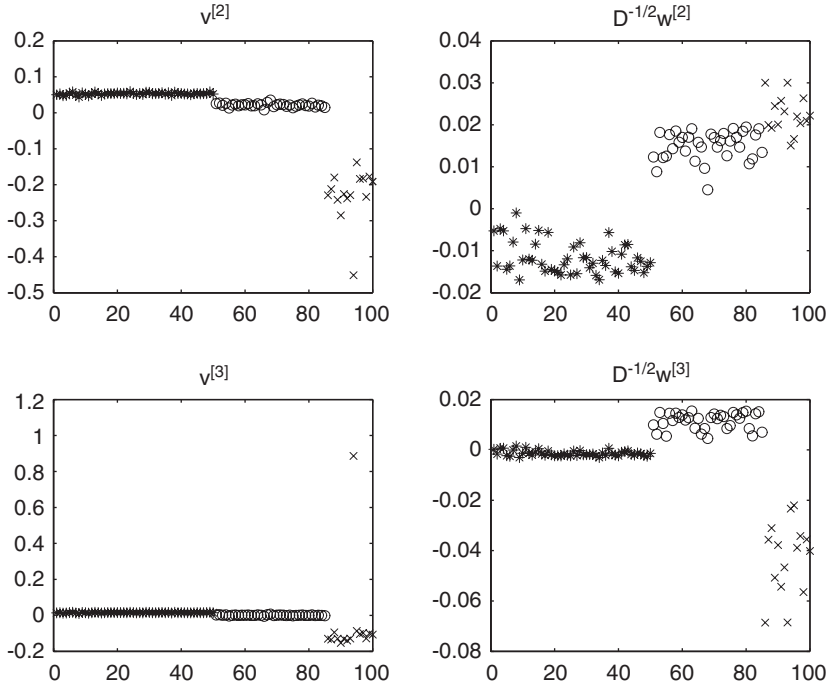

Fig. 2. Components of the second and third eigenvectors for the data from Fig. 1. Left unnormalized. Right normalized.

performance. The basic idea that we use, relaxing from a hard discrete problem to a tractable continuous one, is not new, but we believe that our particular viewpoint has merit in terms of simplicity and insight. Section 4 gives a direct argument that explains why further spectral information is relevant to the task of clustering into multiple groups.

Section 5 presents illustrative numerical experiments on artificially generated data. Section 6 applies spectral clustering to microarray datasets published by cancer researchers.

2. Discrete formulation

Suppose we partition the vertices into two disjoint sets, A and B. We will use the indicator vector $\mathbf{y}$ to denote such a partition by setting $y_i = -\frac{1}{2}$ if vertex i is in A and $y_i = \frac{1}{2}$ if vertex i is in B. A reasonable starting point for finding a good partitioning is to specify the following problem:

$$\min_{y_i \in \{-\frac{1}{2}, \frac{1}{2}\}} \sum_{i,j} (y_i - y_j)^2 w_{ij}. \tag{1}$$

The objective function in (1) contains twice the total weight of edges that span the sets A and B. The edges that span the two sets form what is known as the *cut* of the partition. As it stands, problem (1) is unsuitable because it is solved by assigning all vertices into a single cluster. Shi and Malik [21] proposed to remedy this difficulty by altering the objective function (1). We will take the alternative approach of adding an extra constraint. Such a modification has been proposed before, see, for example, [2,24], but the idea of using it to derive a spectral algorithm appears to be new.

Our next step is thus to introduce a balancing threshold β and consider the problem

$$\min_{\substack{y_i \in \{-\frac{1}{2}, \frac{1}{2}\} \\ |\mathbf{y}^{\mathrm{T}}\mathbf{1}| \leqslant \beta}} \sum_{i,j} (y_i - y_j)^2 w_{ij}. \tag{2}$$

The quantity $2\mathbf{y}^{\mathrm{T}}\mathbf{1}$ counts the difference between the number of vertices assigned to A and B and hence the constraint $|\mathbf{y}^{\mathrm{T}}\mathbf{1}| \leqslant \beta$ dictates how uneven the sizes may be. The choice $\beta = \frac{1}{2}$ forces the closest possible match. At the other extreme, $\beta = N/2$ allows all bi-partitions, including those that involve an empty set, bringing us back to (1).

An alternative to (2) is

$$\min_{\substack{y_i \in \{-\frac{1}{2}, \frac{1}{2}\} \\ |\mathbf{y}^{\mathrm{T}} D\mathbf{1}| \leqslant \beta}} \sum_{i,j} (y_i - y_j)^2 w_{ij}. \tag{3}$$

In this case the extra constraint, $|\mathbf{y}^{\mathrm{T}} D\mathbf{1}| \leqslant \beta$, controls the difference between the total weight in the two clusters—we are balancing the centre of mass. In general, the discrete problems (2) and (3) will be very sensitive to the choice of balancing threshold β. We will find, however, that natural relaxation of these problems leads to solutions that are extremely insensitive to β. In a sense, the relaxed versions will look for an appropriate balance at the same time as finding a small cut size.

3. Relaxation

To make problems (2) and (3) tractable we follow the widely used procedure of *relaxation*. More precisely, we weaken the constraint $y_i \in \{-\frac{1}{2}, \frac{1}{2}\}$ to $y_i \in \mathbb{R}$. This produces a solution $\mathbf{y} \in \mathbb{R}^N$ that must be identified with an indicator vector. The idea is that the components of $\mathbf{y}$ should fall into distinct bands, so that a clustering emerges; as was observed in Fig. 2.

In moving from $y_i \in \{-\frac{1}{2}, \frac{1}{2}\}$ to $y_i \in \mathbb{R}$, however, we must take care of a scaling issue. Without further constraints, the objective function $\sum_{i,j} (y_i - y_j)^2 w_{ij}$ could be made arbitrarily small by scaling all components of $\mathbf{y}$: $y_i \mapsto \varepsilon y_i$. Hence, we must normalize the size of $\mathbf{y}$. The original formulation $y_i \in \{-\frac{1}{2}, \frac{1}{2}\}$ in (2) leads to $\mathbf{y}^{\mathrm{T}}\mathbf{y} = N/4$, producing the problem

$$\min_{\substack{\mathbf{y} \in \mathbb{R}^N \\ |\mathbf{y}^{\mathrm{T}}\mathbf{1}| \leqslant \beta \\ \mathbf{y}^{\mathrm{T}}\mathbf{y} = N/4}} \sum_{i,j} (y_i - y_j)^2 w_{ij}.$$

We find it neater to scale each y_i by $2/\sqrt{N}$ and hence re-write this as

$$\min_{\substack{\mathbf{y}\in\mathbb{R}^N \\ |\mathbf{y}^T\mathbf{1}|\leqslant 2\beta/\sqrt{N} \\ \mathbf{y}^T\mathbf{y}=1}} \sum_{i,j}(y_i - y_j)^2 w_{ij}. \tag{4}$$

For the alternative discrete problem (3) it is natural to add a constraint $\mathbf{y}^T D\mathbf{y} = \theta N$, where θ is some constant. This may be interpreted as follows. If a particular vertex i has a large overall weight, d_i, then, since we are minimizing $\sum (y_i - y_j)^2 w_{ij}$, it is likely to have a large influence over the location of other y_j values. To mitigate the effect of one or two highly weighted nodes influencing the result too strongly, fixing $\mathbf{y}^T D\mathbf{y}$ encourages y_i to be close to zero when d_i is large (in other words, we avoid committing vertex i strongly to either cluster). On the other hand, if a particular vertex, i, has an unusually small overall weight, d_i, then problem (4) is likely to have a solution that takes advantage of this—making y_j small for $j \neq i$ and making $y_i = \mathrm{O}(1)$, so that all terms in $\sum (y_i - y_j)^2 w_{ij}$ are small. This type of solution is unbalanced because it separates the single "outlier" i from the rest of the pack. Replacing $\mathbf{y}^T\mathbf{y} = 1$ by $\mathbf{y}^T D\mathbf{y} = \theta N$ makes such a solution infeasible. Another viewpoint is that whilst $|\mathbf{y}^T D\mathbf{1}| \leqslant \beta$ controls the centre of mass, $\mathbf{y}^T D\mathbf{y} = \theta N$ normalizes the energy—if particles of mass d_i are located at points y_i on a massless rod with centre of gravity at the origin, which is then rotated around its origin, then the energy in the system is proportional to $\mathbf{y}^T D\mathbf{y}$. After rescaling, the resulting problem is

$$\min_{\substack{\mathbf{y}\in\mathbb{R}^N \\ |\mathbf{y}^T D\mathbf{1}|\leqslant \beta/\sqrt{\theta N} \\ \mathbf{y}^T D\mathbf{y}=1}} \sum_{i,j}(y_i - y_j)^2 w_{ij}. \tag{5}$$

Our discussion above suggests that, compared with (4), this normalized version should be less susceptible to the influence of "poorly calibrated" vertices that have abnormally large or small weights.

Problems (4) and (5) may be solved via the following theorem, which is a variation of the Rayleigh–Ritz Theorem, [13, Theorem 4.2.2]. For completeness, we give a proof.

Theorem 3.1. *Let $A \in \mathbb{R}^{N\times N}$ be a symmetric matrix with eigenvalues ordered $v_1 < v_2 \leqslant \cdots \leqslant v_N$ and corresponding mutually orthonormal eigenvectors $\mathbf{x}^{[1]}, \mathbf{x}^{[2]}, \ldots, \mathbf{x}^{[N]}$. Then, for fixed $0 \leqslant \alpha < 1$, the problem*

$$\min_{\substack{\mathbf{y}\in\mathbb{R}^N \\ |\mathbf{y}^T\mathbf{x}^{[1]}|\leqslant \alpha \\ \mathbf{y}^T\mathbf{y}=1}} \mathbf{y}^T A\mathbf{y}$$

is solved by $\mathbf{y} = \alpha\mathbf{x}^{[1]} + \sqrt{1-\alpha^2}\mathbf{x}^{[2]}$.

Proof. We may write $A = X\Sigma X^T$, where $\Sigma = \mathrm{diag}(v_i)$ and X has jth column $\mathbf{x}^{[j]}$ with $X^T X = I$. Setting $\mathbf{z} = X^T\mathbf{y}$, we can rewrite the problem as

$$\min_{\substack{\mathbf{z}\in\mathbb{R}^N \\ |\mathbf{z}^T X^T\mathbf{x}^{[1]}|\leqslant \alpha \\ \mathbf{z}^T\mathbf{z}=1}} \mathbf{z}^T\Sigma\mathbf{z},$$

which simplifies to

$$\min_{\substack{\mathbf{z}\in\mathbb{R}^N \\ |z_1|\leqslant \alpha \\ \mathbf{z}^T\mathbf{z}=1}} \sum_{i=1}^{N} v_i z_i^2.$$

This problem is clearly solved by taking $z_1 = \alpha$, $z_2 = \sqrt{1-\alpha^2}$ and $z_i = 0$ for $i > 2$. This corresponds to $\mathbf{y} = \alpha\mathbf{x}^{[1]} + \sqrt{1-\alpha^2}\mathbf{x}^{[2]}$. □

The following corollary is immediate.

Corollary 1. *For $0 \leqslant \beta < N/2$ the relaxed problem* (4) *has solution*

$$\mathbf{y} = \frac{2\beta}{N\sqrt{N}}\,\mathbf{1} + \sqrt{1 - 4\,\frac{\beta^2}{N^2}}\,\mathbf{v}^{[2]}$$

and for $0 \leqslant \beta < \sqrt{\theta N}\,\|D^{1/2}\mathbf{1}\|_2$ the relaxed problem (5) *has solution*

$$\mathbf{y} = \frac{\beta}{\sqrt{\theta N}\,\|D^{1/2}\mathbf{1}\|_2^2}\,\mathbf{1} + \sqrt{1 - \frac{\beta^2}{\theta N\,\|D^{1/2}\mathbf{1}\|_2^2}}\,D^{-1/2}\mathbf{w}^{[2]}.$$

Both solutions in Corollary 1 contain a term that is a multiple of $\mathbf{1}$. These terms have no relevance to the clustering issue, they simply shift all components uniformly. Hence, for the purpose of using $\mathbf{y}$ as a basis for clustering, the relaxed problem reduces to using $\mathbf{v}^{[2]}$ or $D^{-1/2}\mathbf{w}^{[2]}$ in the unnormalized and normalized cases, respectively.

In the unnormalized case, the corollary shows that effectively the same solution arises for all values $0 \leqslant \beta < N/2$ of the balancing threshold. Now, recall that in the original discrete formulation (2), taking β just less than $N/2$ corresponds to allowing all possible bi-partitions except the pathological case where one set is empty. We conclude that, after relaxation, the algorithm is completely insensitive to the particular choice of β in (2)—it is willing to tolerate all non-trivial cluster sizes.

4. Third eigenvector

It is natural to expect that the eigenvectors corresponding to larger eigenvalues of the Laplacian may have some relevance for clustering. One way to see this is to note that, for example, minimizing $\sum_{i,j}(y_i - y_j)^2 w_{ij}$ over $\mathbf{y} \in \mathbb{R}^N$ subject to $\mathbf{y}^{\mathrm{T}}\mathbf{y} = 1$, $|\mathbf{y}^{\mathrm{T}}\mathbf{v}^{[1]}| \leqslant \alpha_1$ and $|\mathbf{y}^{\mathrm{T}}\mathbf{v}^{[2]}| \leqslant \alpha_2$, with $\alpha_1^2 + \alpha_2^2 < 1$, produces the vector $\alpha_1\mathbf{1}/\sqrt{N} + \alpha_2\mathbf{v}^{[2]} + \sqrt{1 - \alpha_1^2 - \alpha_2^2}\,\mathbf{v}^{[3]}$. (This can be proved along the lines of the proof of Theorem 3.1.) In this sense, $\mathbf{v}^{[3]}$ is the "next best direction, after the Fiedler vector," in which to search.

However, a more intuitive reasoning is possible. To see this, we begin with the following lemma.

Lemma 4.1. *If the weights are altered so that*

$$w_{ij} \mapsto w_{ij} - v v_i^{[2]} v_j^{[2]}, \tag{6}$$

for any fixed $v > \lambda_3 - \lambda_2$, then the new weight matrix has Fiedler vector $\mathbf{v}^{[3]}$. Similarly, if

$$w_{ij} \mapsto w_{ij} - v\sqrt{d_i}\,w_i^{[2]}\sqrt{d_j}\,w_j^{[2]}, \tag{7}$$

for any fixed $v > \mu_3 - \mu_2$, then the new weight matrix has normalized Fiedler vector $D^{-1/2}\mathbf{w}^{[3]}$.

Proof. Perturbation (6) does not change the diagonal matrix D. The new Laplacian $D - W + v\mathbf{v}^{[2]}\mathbf{v}^{[2]\mathrm{T}}$ has eigenvalues $\lambda_1, \lambda_2 + v, \lambda_3, \ldots, \lambda_N$ and corresponding eigenvectors $\mathbf{v}^{[1]}, \mathbf{v}^{[2]}, \ldots, \mathbf{v}^{[N]}$. By construction, λ_3 is now the second smallest eigenvalue, so $\mathbf{v}^{[3]}$ is the Fiedler vector.

The normalized case may be proved similarly. □

To interpret the lemma in the unnormalized case, recall that the original Fiedler vector, $\mathbf{v}^{[2]}$, may be regarded as an attempt to split the vertices into two sets. Since, $|\mathbf{y}^{\mathrm{T}}\mathbf{1}| \leqslant 2\beta/\sqrt{N}$ in (4), we would expect $v_i^{[2]}$ and $v_j^{[2]}$ to have

- the same sign if i and j are assigned to the same cluster,
- opposite signs if i and j are assigned to different clusters.

Perturbation (6) then

- decreases the weight w_{ij} if i and j were placed in the same cluster by $\mathbf{v}^{[2]}$,
- increases the weight w_{ij} if i and j were placed in different clusters by $\mathbf{v}^{[2]}$.

Hence, we may regard $\mathbf{v}^{[3]}$ as solving a new problem where we have attempted to remove the automatic bias toward bi-partition. Thus $\mathbf{v}^{[3]}$ may reveal some of the finer structure in the pairwise affinity data.

The idea in Lemma 4.1 extends to higher eigenvalues. Generally, subtracting sufficiently large multiples of $\mathbf{v}^{[2]}\mathbf{v}^{[2]\mathrm{T}}$, $\mathbf{v}^{[3]}\mathbf{v}^{[3]\mathrm{T}}, \ldots, \mathbf{v}^{[k-1]}\mathbf{v}^{[k-1]\mathrm{T}}$ promotes $\mathbf{v}^{[k]}$ to the position of Fiedler vector.

Similar arguments apply in the normalized case. In (7) the perturbation also takes account of the total weight of each node.

We also remark that perturbations (6) and (7) will, in general, introduce negative weights. This does not invalidate the arguments in Sections 2 and 3. Making w_{ij} more negative corresponds to specifying that i and j are *more dissimilar* and encourages y_i and y_j to be placed further apart in the minimization of the objective function $\sum_{i,j}(y_i - y_j)^2 w_{ij}$.

5. Numerical illustrations

We now examine the behaviour of the algorithms in practice, in the light of the preceding analysis.

To begin, we return to the example in Section 1. In the top left picture of Fig. 2 we see that the unnormalized Fiedler vector, $\mathbf{v}^{[2]}$, is able to identify the three natural clusters. The smallest cluster of 15 '×' points gives rise to the largest

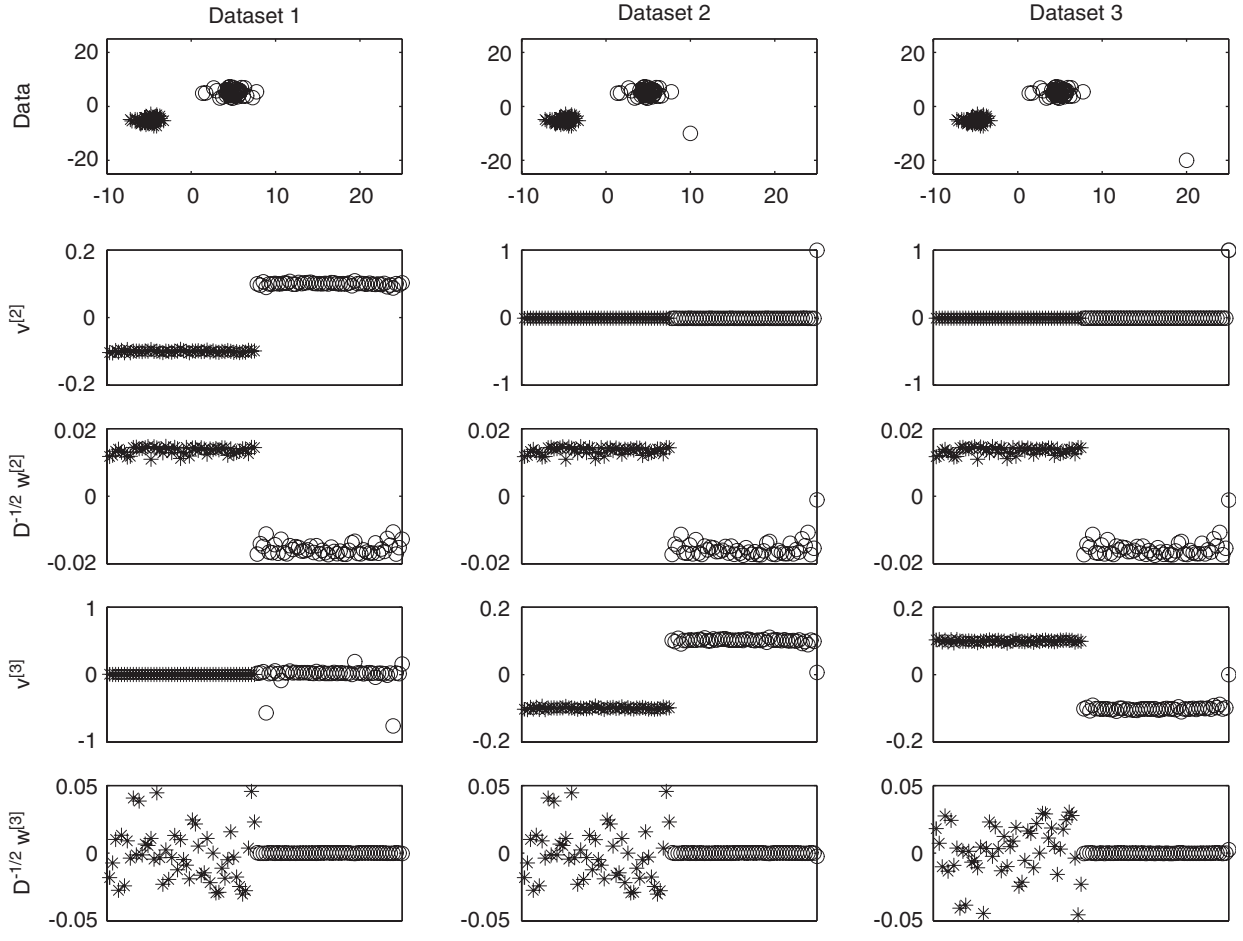

Fig. 3. Top row: three datasets. Similarity weights are given by the reciprocal of the Euclidean distance. In the top left picture, the first 50 points, marked '∗', form one obvious cluster and the second 50, marked '∘', form another. In the middle picture the final (100th) point is moved away, and in the right picture it is moved further away. Second row: components of $\mathbf{v}^{[2]}$ for the unnormalized Laplacian. First 50 components are marked '∗' and second 50 components are marked '∘'. Third row: components of $D^{-1/2}\mathbf{w}^{[2]}$ for the normalized Laplacian. Fourth row: components of $\mathbf{v}^{[3]}$ for the unnormalized Laplacian. Fifth row: components of $D^{-1/2}\mathbf{w}^{[3]}$ for the normalized Laplacian.

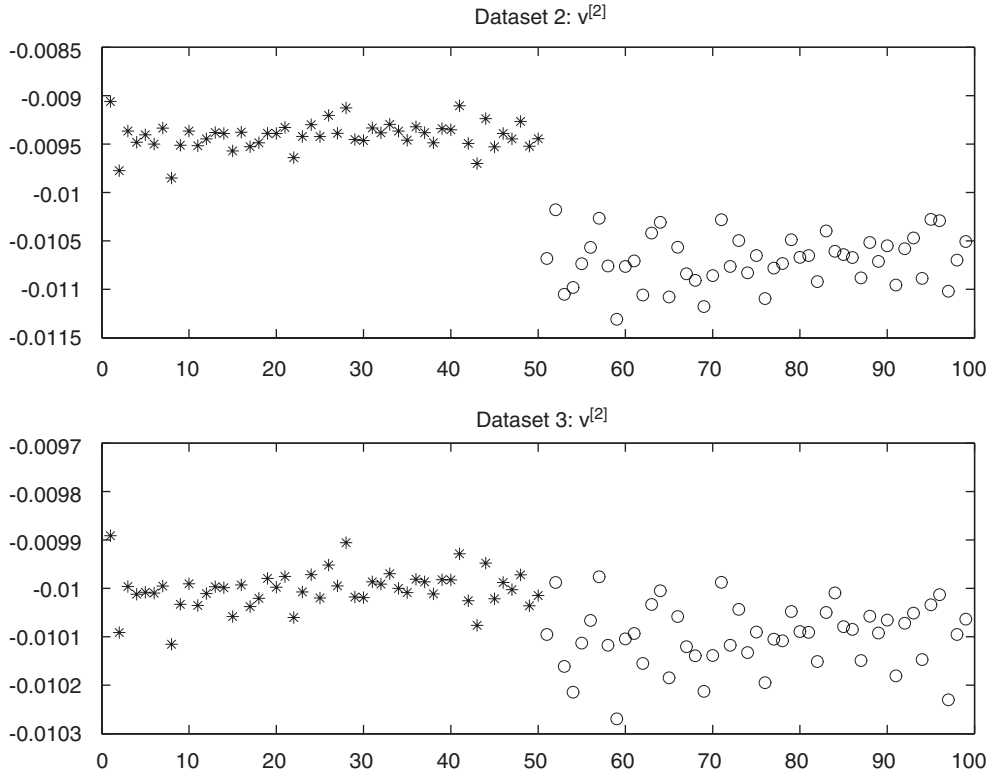

Fig. 4. Zoom in of the second and third pictures on the second row of Fig. 3. This shows the first 99 components of $\mathbf{v}^{[2]}$ for the unnormalized Laplacian with the Dataset 2 (upper picture) and Dataset 3 (lower picture).

components $|v_i^{[2]}|$. This can be understood from an extension of the argument in Section 3 concerning outliers. The '×' data produces relatively small weights and hence the two-sum $\sum_{ij}(y_i - y_j)^2 w_{ij}$ can be minimized subject to $\mathbf{y}^{\mathrm{T}}\mathbf{y} = 1$ and $|\mathbf{y}^{\mathrm{T}}\mathbf{1}| \leqslant 2\beta/\sqrt{N}$ by assigning quite large values to the corresponding y_i components and small values, with opposite sign, to the '∘' and '∗' data points. Further evidence of this phenomenon is seen by the stand-out value of ≈ -0.45 assigned to the 94th data point. This point is the '×' towards the lower left corner of Fig. 1 with coordinates $(x, y) \approx (-7.6, -4.1)$. This "outlier within the outlier group" has been given special prominence. Because $\mathbf{v}^{[2]}$ was able to separate the three main clusters, the third eigenvector, $\mathbf{v}^{[3]}$, gives no further information in that respect—instead it focuses on the outlying point.

By contrast, the top right picture in Fig. 2 shows that the normalized Fiedler vector, $D^{-1/2}\mathbf{w}^{[2]}$, separates the data into two clusters, bundling together the '×' and '∘' points so that the clusters are of equal size. From the point of view of Section 3, this makes sense because the normalized version is insensitive to the relatively small weights present in the '×' set. The third scaled eigenvector of the normalized Laplacian, $D^{-1/2}\mathbf{w}^{[3]}$, successfully breaks down the '∘' and '×' points, as we would expect from Section 4.

Overall, although both algorithms correctly identify the three main clusters, it is interesting to note from Fig. 2 that only the unnormalized case was able to highlight the unplanned outlier that turned up. Whether this extra information is of value depends, of course, on the opinion of the user and the context in which the algorithm is applied.

In Fig. 3 we illustrate further the outlier effect. The top left picture shows Dataset 1, given by 50 clustered '∗' points and 50 '∘' points. As before, similarity weights are given by the reciprocal of the Euclidean distance. For Dataset 2, shown in the upper middle picture, we move the 100th data point away from its cluster, and in Dataset 3 we move it further away. For each dataset, the unnormalized Fiedler vector, $\mathbf{v}^{[2]}$, is shown in the second row. We see that for Dataset 1, $\mathbf{v}^{[2]}$ perfectly separates the two clusters, whereas for Datasets 2 and 3, it appears that $\mathbf{v}^{[2]}$ has distinguished only between the outlier and the rest of the pack. Fig. 4 zooms in on components 1–99 of $\mathbf{v}^{[2]}$ for Datasets 2 and 3. It shows that, for Dataset 2, a generous observer may regard $\mathbf{v}^{[2]}$ as revealing the two large clusters, but for the

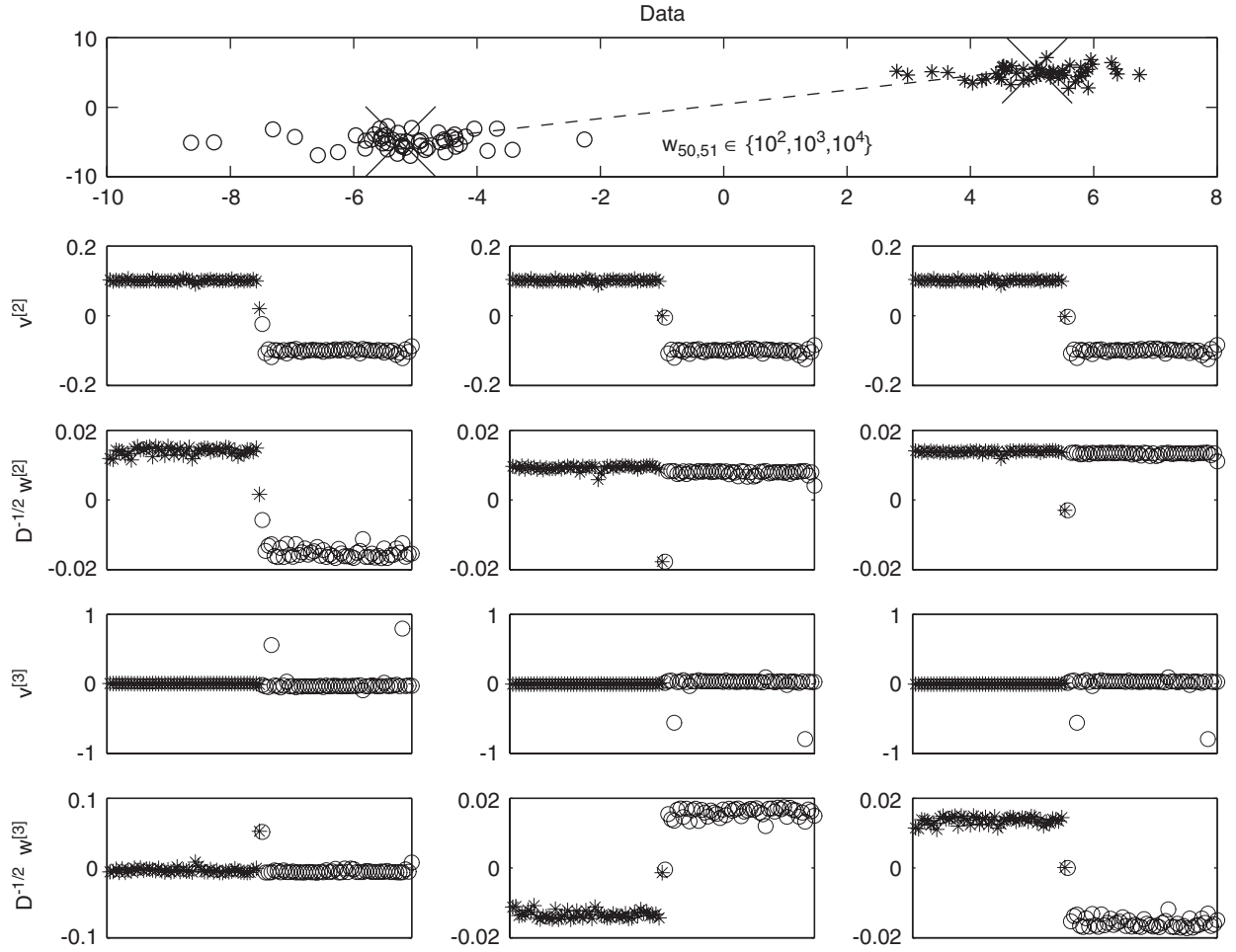

Fig. 5. Top row: the dataset. Similarity weights are given by the reciprocal of the Euclidean distance, with the exception that the weight between the 50th and 51st objects has been re-set. (These two points are marked with large crosses and joined by a dashed line.) Left column uses $w_{50,51} = 10^2$, middle column uses $w_{50,51} = 10^3$, right column uses $w_{50,51} = 10^4$. Second row: components of $\mathbf{v}^{[2]}$ for the unnormalized Laplacian. Third row: components of $D^{-1/2}\mathbf{w}^{[2]}$ for the normalized Laplacian. Fourth row: components of $\mathbf{v}^{[3]}$ for the unnormalized Laplacian. Fifth row: components of $D^{-1/2}\mathbf{w}^{[3]}$ for the normalized Laplacian.

Dataset 3 all information has been lost. For Datasets 2 and 3, the fourth row of Fig. 3 shows that, as we would expect from Section 4, the third eigenvector of the unnormalized Laplacian, $\mathbf{v}^{[3]}$, is able to distinguish between the '$*$' set and the non-outlying '$\circ$' points—these were regarded as a single set by $\mathbf{v}^{[2]}$.

The third row in Fig. 3 shows the normalized Fiedler vector, $D^{-1/2}\mathbf{w}^{[2]}$. Here, as we would expect from the discussion in Section 3, the results are relatively insensitive to the presence of the outlier. For Datasets 2 and 3, $D^{-1/2}\mathbf{w}^{[2]}$ clearly separates the two big clusters and also isolates the outlying point. Because the second eigenvector has captured all relevant information, the third eigenvector, $D^{-1/2}\mathbf{w}^{[3]}$, shown in the fifth row, gives no added value.

Fig. 3 shows what happens when a node is given relatively small weights. In Fig. 5 we create an opposite effect. The data are displayed at the top of Fig. 5. As in Fig. 3, there are two clusters of 50 points, marked '$\circ$' and '$*$'. However, the weight $w_{50,51}$ that associates the final '$\circ$' with the first '$*$' is artificially inflated. The left column uses $w_{50,51} = 10^2$, the middle column uses $w_{50,51} = 10^3$ and the right column uses $w_{50,51} = 10^4$. (The maximum weight w_{ij} over all other i and j was 12.1.) In all three cases, the unnormalized Fiedler vector, $\mathbf{v}^{[2]}$, was able to distinguish the two large clusters (1–49 and 51–100) and isolate the extra points (50 and 51). Here, because $i, j \in \{50, 51\}$ have large weights in the two-sum $\sum_{ij} (y_i - y_j)^2 w_{ij}$, the optimal solution is likely to have $y_{50} \approx y_{51}$. To maintain $|\mathbf{y}^{\mathrm{T}}\mathbf{1}| \leqslant 2\beta/\sqrt{N}$, y_{50} and y_{51}

are placed near the origin and the remaining y_i values split into two groups of opposite sign. Because $\mathbf{v}^{[2]}$ has captured the tri-partition, the third eigenvector, $\mathbf{v}^{[3]}$, adds no further information.

The normalized Fiedler vector, $D^{-1/2}\mathbf{w}^{[2]}$, behaves differently according to the size of $w_{50,51}$. For $w_{50,51} = 10^2$, the algorithm is similar to the unnormalized version. For $w_{50,51} = 10^3$, $D^{-1/2}\mathbf{w}^{[2]}$, creates a bi-partition where the {50, 51} pair are split from the rest, but no further information is provided about the existence of two large clusters. Here, the constraint $|\mathbf{y}^{\mathrm{T}} D\mathbf{1}| \leqslant \beta/\sqrt{\theta N}$, which balances the centre of mass, is strongly influenced by the extra large $w_{50,51}$ value. For $w_{50,51} = 10^4$, we see the effect mentioned in Section 3 where the nodes 50 and 51 have y_i values close to zero. For both $w_{50,51} = 10^3$ and $w_{50,51} = 10^4$, the third eigenvector, $D^{-1/2}\mathbf{w}^{[3]}$, completes the picture by revealing an appropriate bi-partition of the data points that were lumped together by $D^{-1/2}\mathbf{w}^{[2]}$.

As a final comment, we note that the examples support the general implication from Section 3 that the spectral algorithms are searching over all bi-partitions into two non-empty sets, and hence any solution between the extremes of evenly sized and disparately sized clusters may emerge.

6. Tests with microarray data

To illustrate the performance of the spectral algorithms on real-world data we have used results from three Affymetrix oligonucleotide microarray experiments involving leukaemia [4,9], brain tumours [20] and lymphoma [17]. In each

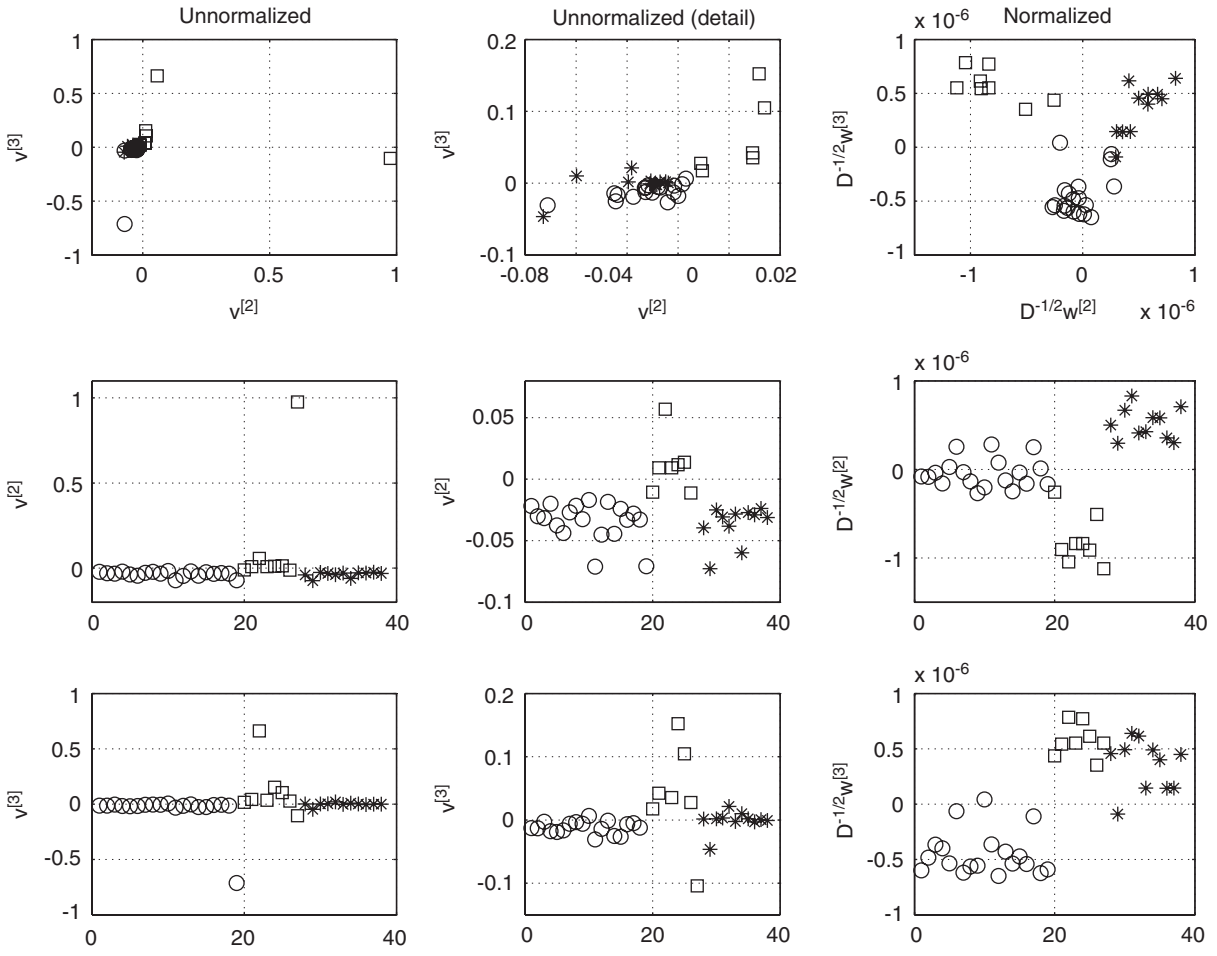

Fig. 6. Leukaemia: ALL-B (circles), ALL-T (squares), AML (stars). Upper line: scatter plots of the second versus third eigenvectors. Middle line: components of the second singular vectors. Lower line: components of the third singular vectors.

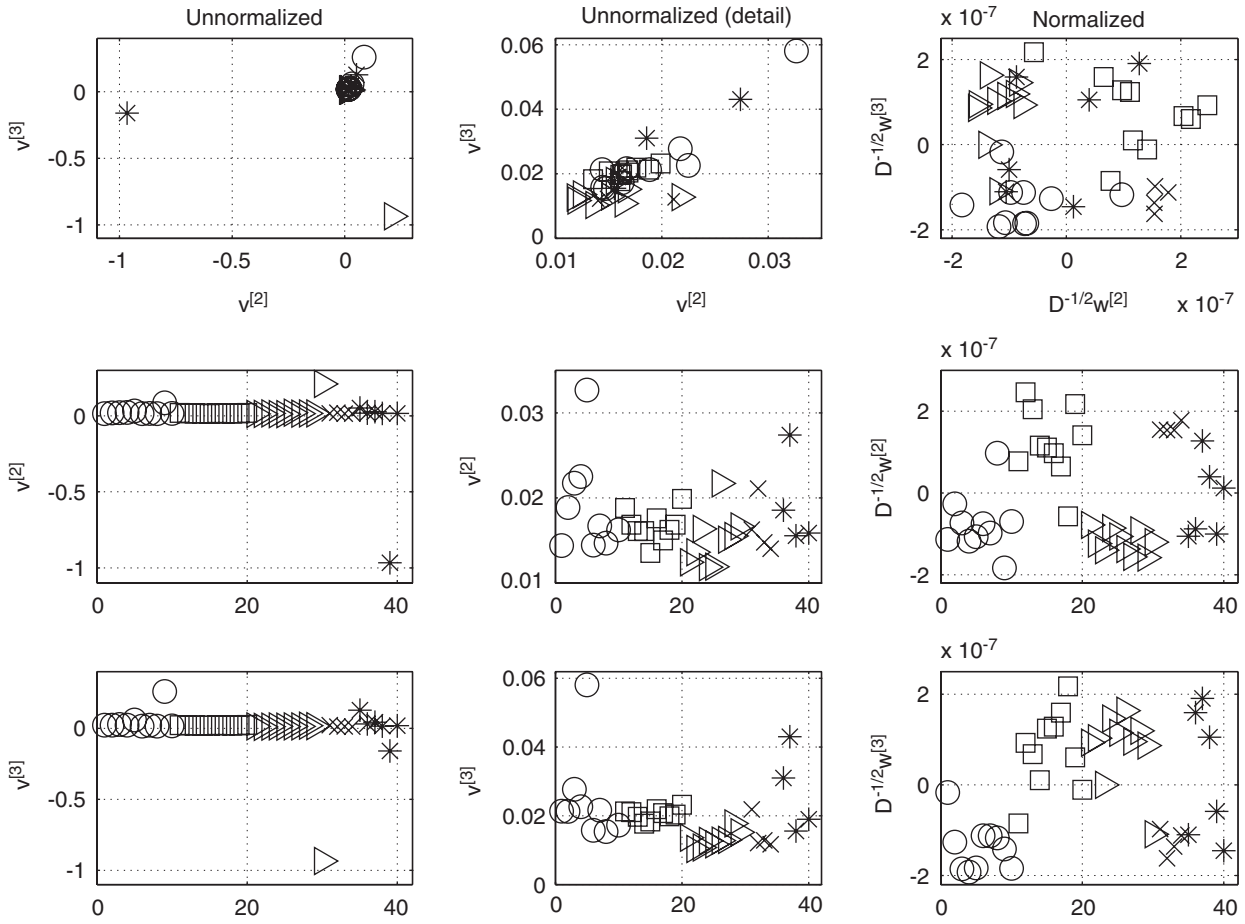

Fig. 7. Central nervous system: medulloblastoma (circles), malignant glioma (squares), normal cerebella (crosses), primitive neuro-ectodermal tumour (stars), rhabdoid (triangles). Upper line: scatter plots of the second versus third eigenvectors. Middle line: components of the second singular vectors. Lower line: components of the third singular vectors.

case the data of interest can be regarded as an array $A \in \mathbb{R}^{M\times N}$, where a_{ij} records the activity of the ith gene in the jth sample. We focus here on clustering the samples, and hence we form the weight matrix $W = A^{\mathrm{T}}A$ and regard w_{ij} as a measure of similarity between samples i and j. (Comparable results were obtained when we formed W as a correlation matrix.) For these datasets a breakdown of the samples into subcategories is available, and hence the clustering results can be judged. However, an ultimate goal in this area is to use clustering methods to discover new information, for example to classify a new patient's tumour. We remark that the related problem of simultaneously clustering genes and samples is considered in, for example, [12,18].

A general conclusion from our tests on microarray data is that the normalized spectral algorithm is far superior to the unnormalized version at revealing biologically relevant information. This phenomenon can be attributed to the wide range of entries in the similarity weight matrix and the relative insensitivity of the normalized algorithm. Hence, in presenting results we display output from both algorithms, but focus on describing the normalized case.

The initial leukaemia dataset [9] has bone marrow samples from $N = 38$ patients. Here, 27 samples are categorized as ALL and 11 are categorized as AML, with ALL being further subdivided into B and T cell types. Expression intensities are given for $M = 7129$ genes. We have used a post-processed version of the data as published in [4], which involves $M = 5000$ genes. All expression intensities smaller than 20 were changed to 20. We remark that several methods have been tested on this data [4,8,18,25].

Fig. 6 shows the performance of the two spectral algorithms: the first two columns correspond to the unnormalized version (the second column zooms in on the first one) and the third column corresponds to the normalized version of

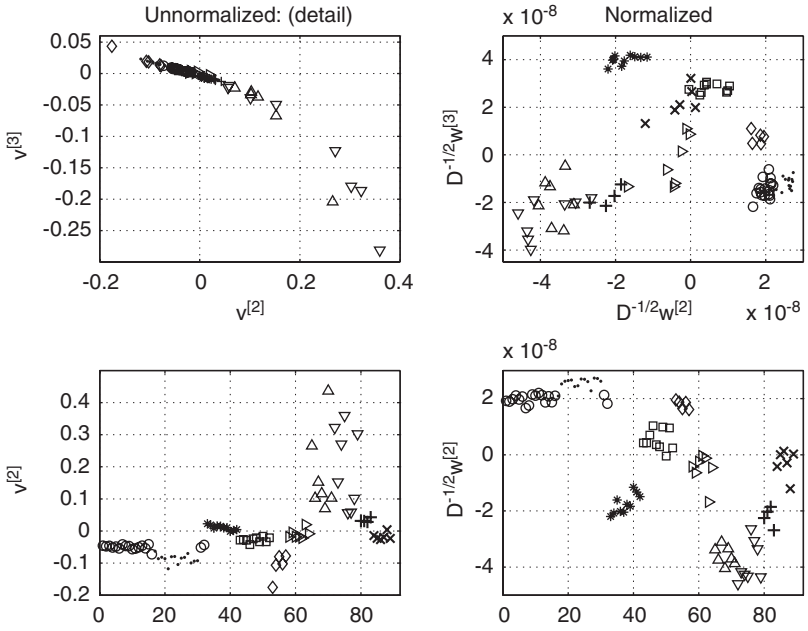

Fig. 8. Lymphoma: B-CLL (biopsy/purified) (circles), B-CLL (biopsy) (dots), GC B-cell (stars), non-GC B-cell (squares), cord blood B-cell (diamonds), DLCL (biopsy/purified) (triangles right), DLCL (cell line) (triangles up), BL-Burkitt (cell line) (triangles down), BL (biopsy/purified) (pluses), FL (crosses). Upper line: scatter plots of the second versus third eigenvectors. Lower line: components of the second singular vectors.

the algorithm. The top row shows scatter plots of the second versus third eigenvectors. The second and the third rows show components of the second and the third eigenvectors, respectively.

The normalized scatter plot (top right) exhibits a good separation of the samples into the three known clusters. The second normalized eigenvector alone did a good job of distinguishing all three clusters and the third normalized eigenvector separated ALL-B from the rest of the samples. The unnormalized version only separated the three outliers with smallest overall expression levels.

We next analysed the dataset A1 of [20], consisting of expression values of $M = 7129$ genes measured in 40 brain tumour samples: 10 medulloblastomas (MD), 10 malignant gliomas (Mglio), 4 normal cerebellas (Ncer), 6 primitive neuro-ectodermal tumours (PNET) and 10 rhabdoids (Rhab). We changed allexpression intensities smaller than 20 to 20 and then removed the genes with constant expression 20 across all samples.

In Fig. 7 we see that overlapping of the brain tumour subgroups is slightly larger than in case of leukaemia. This indicates that tumour subclasses may share some of the active genes. MD (circles) and Rhab (triangles) were separated from the Ncer (crosses) and Mglio (squares) by the second normalized eigenvector. Strong correlation of Ncer and Mglio was also mentioned in [25]. The third normalized eigenvector separated the MD (circles) and Ncer (crosses) from the group of the Rhab (triangles) and Mglio (squares). In general, two normalized eigenvectors were able to separate four out of the five brain tumour subgroups.

Data from [17] included five cell types: B cell-derived chronic lymphocytic leukaemia (B-CLL), normal B cell subpopulations (B-cell), diffuse large cell lymphoma (DLCL), Burkitt lymphoma (BL), and follicular lymphoma (FL).

The authors in [17] draw a few conclusions from comparison of gene expression profiles of CLL to those of B cell subsets: GC (centroblasts and centrocytes), non-GC (naive and memory) and GC-independent cord blood B cells. First, they recognized purified and non-purified cases of B-CLL as different. In Fig. 8 these are clearly distinguished by the second normalized singular vector. Note that the dots, the only non-purified biopsy in the dataset, are situated far right in the scatter plot (top of the $D^{-1/2}\mathbf{w}^{[2]}$ plot). Moreover, two groups of cell lines DLCL and BL are situated together on the left-hand side of the scatter plot (bottom of the $D^{-1/2}\mathbf{w}^{[2]}$ plot). There is clear separation of purified biopsy DLCL from cell line DLCL but only marginal separation of these subgroups in case of BL. Second, in [17] CLL is identified as significantly more related to non-GC than to GC. This is in agreement with our result: the second (and the third) normalized singular vector placed non-GC closer to B-CLL than GC. Finally, [17] found that CLL is more

related to non-GC than to GC-independent cord blood B cells. In Fig. 8 cord blood B cells are closer to B-CLL than GC or non-GC.

References

[1] F. Abascal, A. Valencia, Clustering of proximal sequence space for the identification of protein families, Bioinformatics 18 (2002) 908–921.
[2] C.J. Alpert, S.-Z. Yao, Spectral partitioning: the more eigenvectors, the better, in: Proceedings of the 32nd Conference on Design Automation, 1995, pp. 195–200.
[3] S.T. Barnard, A. Pothen, H.D. Simon, A spectral algorithm for envelope reduction of sparse matrices, Numer. Linear Algebra Appl. 2 (1995) 317–334.
[4] J.P. Brunet, P. Tamayo, T.R. Golub, J.P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization, Proc. Nat. Acad. Sci. 101 (2004) 4164–4169.
[5] I.S. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in: Proceedings of the Seventh ACM SIGKDD Conference, 2001.
[6] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory, Czechoslovak Math. J. 25 (1975) 619–633.
[7] C. Fowlkes, Q. Shan, S. Belongie, J. Malik, Extracting global structure from gene expression profiles, in: S.M. Lin, K.F. Johnson (Eds.), Methods of Microarray Data Analysis II, 2002.
[8] G. Getz, E. Levine, E. Domany, Coupled two-way clustering analysis of gene microarray data, Proc. Nat. Acad. Sci. 97 (2000) 12079–12084.
[9] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caliguri, C.D. Bloomfield, E.S. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[10] B. Hendrickson, R. Leland, The Chaco user's guide: version 2.0, Technical Report, SAND94–2692, Sandia National Laboratories, Albuquerque, 1994.
[11] B. Hendrickson, R. Leland, An improved spectral graph partitioning algorithm for mapping parallel computations, SIAM J. Sci. Statist. Comput. 16 (1995) 452–469.
[12] D.J. Higham, G. Kalna, J.K. Vass, Analysis of the singular value decomposition as a tool for processing microarray expression data, in: Proceedings of ALGORITMY 2005, Slovak University of Technology, 2005, pp. 250–259.
[13] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
[14] Y. Hu, J.A. Scott, `HSL_MC735`: a fast multilevel Fiedler and profile reduction code, RAL-TR-2003-36, Numerical Analysis Group, Computational Science and Engineering Department, Rutherford Appleton Laboratory, 2003.
[15] E. Jennings, L. Motyckova, D. Carr, Evaluating graph theoretic clustering algorithms for reliable multicasting, in: Proceedings of IEEE GLOBECOM, 2001.
[16] R. Kannan, S. Vempala, A. Vetta, On clusterings: good, bad and spectral, in: Proceedings of the 41st Foundations of Computer Science (FOCS '00), 2000.
[17] U. Klein, Y. Tu, G.A. Stolovitzky, M. Mattioli, G. Cattoretti, H. Husson, A. Freedman, G. Inghirami, L. Cro, L. Baldini, A. Neri, A. Califano, R. Dalla-Favera, Gene expression profiling of B cell chronic lymphocytic leukemia reveals a homogeneous phenotype related to memory B cells, J. Experimental Medicine 194 (2001) 1625–1638.
[18] Y. Kluger, R. Basri, J.T. Chang, M. Gerstein, Spectral biclustering of microarray data: coclustering genes and conditions, Genome Res. 13 (2003) 703–716.
[19] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, vol. 14, NIPS, 2001.
[20] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J.Y. Kim, L.C. Goumnerova, P.M. Black, C. Lau, J.C. Allen, D. Zagzag, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D.N. Louis, J.P. Mesirov, E.S. Lander, T.R. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression, Nature 415 (2002) 436–442.
[21] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Machine Intelligence 22 (2000) 888–905.
[22] B. Snel, P. Bork, M. Huymen, The identification of functional modules from the genomic association of genes, Proc. Nat. Acad. Sci 99 (2002) 5890–5895.
[23] D.A. Spielman, S.-H. Teng, Spectral partitioning works: planar graphs and finite element meshes, in: Proceedings of the 37th Annual IEEE Conference on Foundations of Computer Science, 1996.
[24] R. Van Driessche, D. Roose, An improved spectral bisection algorithm and its application to dynamic load balancing, Parallel Comput. 21 (1995) 29–48.
[25] J. Wang, T.H. Bo, I. Jonassen, O. Myklebost, E. Hovig, Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data, BMC Bioinformatics 4 (2003) 60.
[26] Y. Weiss, Segmentation using eigenvectors: a unifying view, in: Proceedings IEEE International Conference on Computer Vision, 1999, pp. 975–982.
[27] E.P. Xing, R.P. Karp, CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts, Bioinformatics (Discovery Note) 1 (2001) 1–9.