Noname manuscript No.
(will be inserted by the editor)

Minimum Spectral Connectivity Projection Pursuit

Divisive Clustering using Optimal Projections for Spectral Clustering

David P. Hofmeyr ·
Nicos G. Pavlidis ·
Idris A. Eckley

Received: date / Accepted: date

Abstract We study the problem of determining the optimal low dimensional projection for maximising the separability of a binary partition of an unlabelled dataset, as measured by spectral graph theory. This is achieved by finding projections which minimise the second eigenvalue of the graph Laplacian of the projected data, which corresponds to a non-convex, non-smooth optimisation problem. We show that the optimal univariate projection based on spectral connectivity converges to the vector normal to the maximum margin hyperplane through the data, as the scaling parameter is reduced to zero. This establishes a connection between connectivity as measured by spectral graph theory and maximal Euclidean separation. The computational cost associated with each eigen-problem is quadratic in the number of data. To mitigate this issue, we propose an approximation method using microclusters with provable approximation error bounds. Combining multiple binary partitions within a divisive hierarchical model allows us to construct clustering solutions admitting clusters with varying scales and lying within different subspaces. We evaluate the performance of the proposed method on a large collection of benchmark datasets and find that it compares favourably with existing methods for projection pursuit and dimension reduction for data clustering. Applying the proposed approach for a decreasing sequence of scaling parameters allows us to obtain large margin clustering solutions, which are found to be competitive with those from dedicated maximum margin clustering algorithms.

Keywords Spectral clustering · dimension reduction · projection pursuit · maximum margin

David Hofmeyr acknowledges support from the EPSRC funded EP/H023151/1 STOR-i centre for doctoral training as well as the Oppenheimer Memorial Trust. Idris Eckley was supported by EPSRC grant EP/N031938/1 (StatScale)

D.P. Hofmeyr
Dept. of Statistics and Actuarial Science
Stellenbosch University, South Africa
E-mail: dhofmeyr@sun.ac.za

N.G. Pavlidis
Dept. of Management Science
Lancaster University, United Kingdom

I.A. Eckley
Dept. of Mathematics and Statistics
Lancaster University, United Kingdom

1 Introduction

Identifying distinct groups, or *clusters*, in unlabelled data is a fundamental task in exploratory data analysis, with applications in diverse disciplines ranging from computer science and biology to sociology and marketing. Spectral clustering methods have gained considerable attention because of their simplicity, versatility and strong performance in numerous applications (Shi and Malik, 2000; Weiss, 1999; Ning et al., 2010; Chi et al., 2009). One of the appealing properties of spectral clustering is its ability to identify highly non-convex clusters, which may lie on or close to highly non-linear manifolds. It is, however, sensitive to choices of scaling and to irrelevant or noisy features which may be present in the data (Bach and Jordan, 2006; Niu et al., 2011).

In spectral clustering, clusters are defined as strongly connected components of a graph whose vertices correspond to data points, and edge weights represent pairwise similarities between them (von Luxburg, 2007). The minimum-cut problem seeks the partition of the graph that minimises the sum of edge weights connecting different components of the partition. In other words, the partition which minimises the total similarity between data assigned to different clusters. Although intuitive this formulation frequently produces partitions in which some components contain very few vertices (data), which may not constitute complete clusters. To avoid this, normalisations of the minimum-cut problem that favour balanced partitions are used. Normalisation, however, renders the problem NP-hard (Wagner and Wagner, 1993), and so a continuous relaxation is solved instead. The solution of the relaxed problem is given by the eigenvectors of the *graph Laplacian* matrix. This spectral decomposition of the graph Laplacian gives rise to the term spectral clustering.

The successful application of any clustering method critically depends on the extent to which the true group

structure in the data is captured by spatial similarities between points. However, the presence of irrelevant and noisy features, which abound in modern applications, can distort this spatial structure. This has been shown to have particularly adverse effects on the performance of spectral clustering, even in problems of moderate dimensionality (Bach and Jordan, 2006; Niu et al., 2011). Dimension reduction techniques attempt to mitigate the effects of noisy and irrelevant features by identifying low dimensional representations of a dataset that preserve the maximum amount of relevant information. Commonly these low dimensional representations are defined by the projection of the data into a linear subspace. Classical techniques, like principal component analysis (PCA), although widely used in clustering, are not guaranteed to identify subspaces that preserve cluster structure. More recently a number of dimension reduction methods that explicitly aim to reveal cluster structure have been developed (Krause and Liebscher, 2005; Pavlidis et al., 2016; Hofmeyr and Pavlidis, 2015; Peña and Prieto, 2001; Niu et al., 2011).

Peña and Prieto (2001) show that under certain conditions the one-dimensional projection of the data with minimum kurtosis maximises bimodality. Such a projection can thus be used to separate *high-density clusters*, defined as contiguous regions of high probability density around modes of the (assumed) underlying probability density function. For the same purpose, Krause and Liebscher (2005) propose maximising the *dip statistic* (Hartigan and Hartigan, 1985), a measure of departure from unimodality of a univariate dataset. More recently Pavlidis et al. (2016) proposed an approach that aims to identify regions of low probability density that separate high-density clusters. This is achieved by identifying the univariate subspace normal to the hyperplane that has the minimum integrated density along it, called the *minimum density hyperplane*. Hofmeyr and Pavlidis (2015) proposed a method to identify projections that maximise the variance-ratio clusterability measure (Zhang, 2001). This measure is a normalisation of the K-means objective, which is invariant to changes in scale and is thus less susceptible to projections which exhibit high variance but little cluster structure. The problem of dimensionality reduction for spectral clustering was first considered by Niu et al. (2011). A detailed description of this method and its relation to our work is provided in Section 2 after the presentation of necessary background material.

The main problem we consider in this paper is the identification of the optimal projection to bi-partition a dataset through spectral clustering. This is achieved by minimising the second smallest eigenvalue of the graph Laplacian, which measures the spectral connectivity between the two clusters. We consider the graph Laplacians arising from the two most widely used normalisations of the minimum-cut objective, namely Ratio Cut (Hagen and Kahng, 1992) and Normalised Cut (Shi and Malik, 2000). Although both formulations can lead to high quality clustering models, our experience suggests that for our purposes the Normalised Cut formulation yields overall superior performance. Applying this bi-partitioning approach recursively produces a divisive spectral clustering algorithm capable of identifying clusters with varying scales and defined in different subspaces. The minimisation of the sum of the K smallest eigenvalues of the normalised graph Laplacian with respect to a projection of the data was first proposed by Niu et al. (2011) to perform dimension reduction for spectral clustering.

In this paper we develop an improved methodology for finding optimal projections based on the spectral clustering objective, and provide new theoretical perspectives on the problem. We perform a rigorous investigation into the continuity and differentiability properties of eigenvalues of graph Laplacians as functions of the projection, and find that they are Lipschitz continuous (and hence differentiable almost everywhere), and everywhere directionally differentiable. We derive expressions for the derivative of an eigenvalue with respect to the projection when the eigenvalue is simple, thereby allowing us to minimise the objective directly using generalised gradient descent methods. This approach is guaranteed to converge to a local minimum, whereas existing methodology for this problem does not directly minimise the overall objective and may fail to find an optimal projection. In addition, we provide a formulation of the directional derivative which allows us to easily derive optimality conditions for the proposed method. Although our focus is on minimising the second smallest eigenvalue our analysis applies to an arbitrary eigenvalue of the Laplacian, and so the proposed methodology can easily be extended to minimising sums of eigenvalues of graph Laplacians.

Each eigenvalue computation requires $\mathcal{O}(N^2)$ operations, where N is the size of the dataset. This can be prohibitive for large datasets. We show how preprocessing the dataset using microclusters provides an approximation of the optimisation surface which enables a speed-up of up to two orders of magnitude without an appreciable degradation in empirical clustering accuracy. We also derive theoretical worst case error bounds for this approximation.

We establish an asymptotic connection between optimal univariate projections for spectral bi-partitioning and maximum margin hyperplanes. Formally, we show

that as the scaling parameter defining pairwise similarities is reduced to zero, the optimal one-dimensional projection for spectral bi-partitioning converges to the vector normal to the largest margin hyperplane through the data. This establishes a theoretical connection between connectivity as measured by spectral graph theory and Euclidean separation, which underlies maximum margin clustering (Xu et al., 2004; Zhang et al., 2009), an increasingly popular and effective approach to clustering.

The remainder of the paper is organised as follows. In Section 2 we provide a brief introduction to spectral clustering, and existing dimension reduction based on the spectral clustering objective. Section 3 presents our methodology for finding optimal projections based on spectral connectivity. Section 4 describes the theoretical connection between the optimal one-dimensional projection for spectral bi-partitioning and maximum margin hyperplanes. In Section 5 we discuss an approximation technique which allows for a substantial improvement in computation time of the method, and derive theoretical worst case error bounds. Experimental results and sensitivity analyses are presented in Section 6.

2 Background

In this section we provide a brief introduction to spectral clustering, with particular attention to binary partitioning, and discuss existing methodology for dimension reduction based on the spectral clustering objective. Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ denote a dataset in $\mathbb{R}^d$. Then define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where vertices correspond to elements in $\mathcal{X}$, and the *undirected* edges assume weights equal to the pairwise *similarities* between data. The information in $\mathcal{G}$ can be represented by the *adjacency*, or *affinity* matrix, $A \in \mathbb{R}^{N\times N}$, with $A_{ij} = \mathcal{E}_{ij} := \text{similarity}(x_i, x_j)$. The *degree* of the i-th vertex is defined as $d_i = \sum_{j=1}^{N} A_{ij}$, and the degree matrix is defined as $D = \text{diag}(d_1, \ldots, d_N)$. For a subset $\mathcal{C} \subset \mathcal{X}$, the size of $\mathcal{C}$ can be measured either by its cardinality, $|\mathcal{C}|$, or by its *volume*, $\text{vol}(\mathcal{C}) := \sum_{i:x_i\in\mathcal{C}} d_i$.

Definition 1 *The* normalised minimum-cut of a graph *is the solution to the optimisation problem*

$$\min_{\mathcal{C}\subset\mathcal{X}} \sum_{i,j:x_i\in\mathcal{C},x_j\in\mathcal{X}\setminus\mathcal{C}} A_{ij}\left(\frac{1}{size(\mathcal{C})} + \frac{1}{size(\mathcal{X}\setminus\mathcal{C})}\right). \quad (1)$$

When $\text{size}(\mathcal{C}) = |\mathcal{C}|$ the above objective is referred to as *Ratio Cut* (Hagen and Kahng, 1992), whereas when $\text{size}(\mathcal{C}) = \text{vol}(\mathcal{C})$ it is known as *Normalised Cut* (Shi and Malik, 2000). Hagen and Kahng (1992) and Shi and Malik (2000) have shown that the normalised minimum-cut problems arising from these two definitions of size can be formulated in terms of the *graph Laplacian* matrices,

$$(standard)\ L = D - A, \quad (2)$$

$$(normalised)\ L_{\text{N}} = D^{-1/2}LD^{-1/2}, \quad (3)$$

as follows. For $\mathcal{C} \subset \mathcal{X}$ define $u^{\mathcal{C}} \in \mathbb{R}^N$ to be the vector with i-th entry,

$$u_i^{\mathcal{C}} = \begin{cases} \sqrt{\text{size}\left(\mathcal{X}\setminus\mathcal{C}\right)/\text{size}(\mathcal{C})}, \text{ if } x_i \in \mathcal{C} \\ -\sqrt{\text{size}\left(\mathcal{C}\right)/\text{size}(\mathcal{X}\setminus\mathcal{C})}, \text{ if } x_i \in \mathcal{X}\setminus\mathcal{C}. \end{cases} \quad (4)$$

For $\text{size}(\mathcal{C}) = |\mathcal{C}|$, the optimisation problem in (1) can be written as,

$$\min_{\mathcal{C}\subset X} (u^{\mathcal{C}})^\top L u^{\mathcal{C}} \ \text{ s.t. } \ u^{\mathcal{C}} \perp \mathbf{1},\ \|u^{\mathcal{C}}\| = \sqrt{N}. \quad (5)$$

If instead $\text{size}(\mathcal{C}) = \text{vol}(\mathcal{C})$ then (1) is equivalent to,

$$\min_{\mathcal{C}\subset X} (u^{\mathcal{C}})^\top L u^{\mathcal{C}} \ \text{ s.t. } Du^{\mathcal{C}} \perp \mathbf{1},\ (u^{\mathcal{C}})^\top D u^{\mathcal{C}} = \text{vol}(\mathcal{X}). \quad (6)$$

Both problems in (5) and (6) are NP-hard (Wagner and Wagner, 1993). However continuous relaxations, in which the discreteness condition on $u^{\mathcal{C}}$, Eq. (4), is removed, can be solved in quadratic time (Hagen and Kahng, 1992; Shi and Malik, 2000). The solutions to the relaxed problems are given by the second eigenvector of L, and the second eigenvector of the generalised eigen-equation $Lu = \lambda Du$ respectively. The latter is thus equivalently solved by $D^{-1/2}u$, where u is the second eigenvector of L_{N}. The above approach readily extends to the problem of obtaining a K-partition of the dataset. In this case the solution is obtained from the eigenvectors corresponding to the K smallest eigenvalues of L or L_{N} (von Luxburg, 2007), respectively.

Dimension reduction based on the spectral clustering objective using the normalised graph Laplacian was first considered by Niu et al. (2011). The objective considered by the authors is equivalent to the objective we consider, and can be formulated as follows,

$$\max_{U,V} \text{trace}(U^\top D^{-1/2} A D^{-1/2} U) \quad (7a)$$

$$s.t. \quad U^\top U = I \quad (7b)$$

$$A_{ij} = k(\|V^\top x_i - V^\top x_j\|) \quad (7c)$$

$$V^\top V = I. \quad (7d)$$

Note that since $L_{\text{N}} = I - D^{-1/2}AD^{-1/2}$, the trace maximisation in (7a) is equivalent to $\min_{U,V} \text{trace}(U^\top L_{\text{N}} U)$. The elements of the affinity matrix, A, are determined by a function, $k(\cdot)$, of the pairwise distances of the points projected into the subspace defined by the projection matrix V; and D is the corresponding degree

matrix. It is clear that for a given V the matrix U that maximises the trace in (7a) has columns given by the K eigenvectors associated with the K largest eigenvalues of $D^{-1/2}AD^{-1/2}$ (or equivalently the K smallest eigenvalues of L_{N}). To solve the problem in (7), Niu et al. (2011) propose an algorithm that alternates between two stages: (i) for a fixed V a spectral decomposition of L_{N} determines the optimal U; and (ii) fixing U and D a gradient ascent method is used to maximise $\mathrm{trace}(U^\top D^{-1/2}AD^{-1/2}U)$ with respect to V, where the dependence of this objective on the projection matrix V is through Eq. (7c). This process is then iterated. However, this approach does not account for the fact that the degree matrix D is a function of A and therefore it is itself a function of V. An ascent direction for the objective assuming a fixed D is thus not necessarily an ascent direction for the overall objective. We have further observed that in practice this algorithm is not guaranteed to lead to an increase in the overall objective across iterations and may thus fail to converge. In the following section we derive expressions for the gradient of the overall objective as a function of the projection, allowing us to optimise it directly.

3 Projection Pursuit for Spectral Connectivity

In this section we study the problem of minimising the second eigenvalue of the graph Laplacian of the projected data. If the projected data are bi-partitioned through spectral clustering, then the projection that minimises the second eigenvalue of the graph Laplacian minimises the connectivity between the two clusters, as measured by spectral graph theory.

Let $\mathcal{X} = \{x_1, \dots x_N\}$ be a dataset in $\mathbb{R}^d$. We define the *projection matrix* V as a $d \times l$ matrix, with $l < d$, whose columns $\{v_1, \dots, v_l\}$, have unit norm. With this formulation it is convenient to express V in polar coordinates. Let $\Theta = [0, \pi)^{(d-1)\times l}$, then for $\boldsymbol{\theta} \in \Theta$, the projection matrix $V(\boldsymbol{\theta})$ is given by,

$$V(\boldsymbol{\theta})_{ij} = \begin{cases} \cos(\boldsymbol{\theta}_{ij}) \prod_{k=1}^{i-1} \sin(\boldsymbol{\theta}_{kj}), & i = 1, ..., d-1 \\ \prod_{k=1}^{d-1} \sin(\boldsymbol{\theta}_{kj}), & i = d. \end{cases} \tag{8}$$

The l-dimensional *projected data set* is denoted by $\mathcal{P}(\boldsymbol{\theta}) = \{p(\boldsymbol{\theta})_1, \dots, p(\boldsymbol{\theta})_N\} = \{V(\boldsymbol{\theta})^\top x_1, \dots, V(\boldsymbol{\theta})^\top x_N\}$. We also define the data matrix, $X \in \mathbb{R}^{d\times N}$, and the projected data matrix $P \in \mathbb{R}^{l\times N}$, as matrices whose columns contain the original and projected data, respectively.

We define $L(\boldsymbol{\theta})$ (resp. $L_{\mathrm{N}}(\boldsymbol{\theta})$) as the Laplacian (resp. normalised Laplacian) of the graph constructed from the projected data set $\mathcal{P}(\boldsymbol{\theta})$. Throughout we use $\lambda_i(\cdot)$ to denote the i-th smallest eigenvalue of its real symmetric matrix argument, and we assume that all eigenvectors are normalised. Edge weights in the graph of $\mathcal{P}(\boldsymbol{\theta})$ are determined by a Lipschitz continuous and continuously differentiable *similarity function* $s : \mathbb{R}^{l\times N} \times \{1 \dots N\}^2 \to \mathbb{R}^+$, in that the affinity matrix is given by,

$$A(\boldsymbol{\theta})_{ij} := s(P(\boldsymbol{\theta}), i, j) = k\left(d(p(\boldsymbol{\theta})_i, p(\boldsymbol{\theta})_j)/\sigma\right), \tag{9}$$

where $k : \mathbb{R}^+ \to \mathbb{R}^+$ is a smooth decreasing function, $d(\cdot,\cdot)$ is a metric and $\sigma > 0$ is the *scaling parameter*. It is common to use the Euclidean metric, however our experience has shown that projection pursuit for spectral clustering can be sensitive to outliers when this metric is used. This is especially the case when using the standard Laplacian. To mitigate against this we define a metric which encourages cluster boundaries to intersect a chosen convex set, $\boldsymbol{\Delta}(\boldsymbol{\theta})$, which depends on the projection $\boldsymbol{\theta}$. This is achieved by defining $d(\cdot,\cdot)$ so that the resulting similarities between points outside $\boldsymbol{\Delta}(\boldsymbol{\theta})$, which may be outliers, and other points, are increased. A detailed discussion is provided in Appendix A.

A common requirement in linear dimension reduction methods is that the projection matrix V is orthonormal, that is $V^\top V = I$. Niu et al. (2011) directly enforce this constraint by generating the columns of V sequentially and optimising each column over the null space of previously determined columns. By restricting the domain of the optimisation problem to the manifold of $d\times l$ orthonormal matrices, known as the Stiefel manifold, it is possible to optimise over the entire matrix V (Edelman et al., 1998; Boumal et al., 2014). However, optimisation algorithms operating over the Stiefel manifold have only been shown to have guaranteed convergence when the objective function is everywhere continuously differentiable. As we discuss in the next section this requirement is not necessarily met by the eigenvalues of graph Laplacians. We instead introduce a penalty term into the objective function which leads to approximately orthogonal projection matrices. Specifically, we consider the objective,

$$\min_{\boldsymbol{\theta}\in\Theta} \lambda_2(L(\boldsymbol{\theta})) + \omega \sum_{i\neq j} \left(V(\boldsymbol{\theta})_i^\top V(\boldsymbol{\theta})_j\right)^2, \tag{10}$$

or replacing $\lambda_2(L(\boldsymbol{\theta}))$ with $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ in the normalised case. As in the case of optimising over the Stiefel manifold, this formulation enables us to update the entire matrix V at each iteration. This is an important advantage because the expensive computation of the eigenvalue of the graph Laplacian is performed once rather than l times for each complete update of V.

3.1 Continuity and Differentiability

In this subsection we investigate the continuity and differentiability properties of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$, which are required to establish global convergence of the optimisation algorithm discussed in Section 3.2.

To begin with, simple applications of the inequalities of Weyl (1912) and Schur (1911) give us,

$$|\lambda_i(L(\boldsymbol{\theta})) - \lambda_i(L(\boldsymbol{\theta}'))| \leq N\sqrt{\max_{ij}|L(\boldsymbol{\theta}) - L(\boldsymbol{\theta}')|_{ij}}.$$

By assumption the similarity function, s, is Lipschitz continuous in $P \in \mathbb{R}^{l\times N}$ for fixed i, j. The elements of $L(\boldsymbol{\theta})$ are therefore Lipschitz continuous as compositions of Lipschitz functions ($V(\boldsymbol{\theta})$ is Lipschitz in $\boldsymbol{\theta}$ as a collection of finite products of Lipschitz functions). Thus the objective $\lambda_2(L(\boldsymbol{\theta}))$ is Lipschitz continuous in $\boldsymbol{\theta}$. An analogous argument can be used to show that $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ is Lipschitz continuous. Rademacher's theorem therefore tells us that $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ are almost everywhere differentiable (Polak, 1987). Generalised gradient descent methods therefore provide a natural framework for finding locally optimal projections for spectral bipartitioning (Polak, 1987).

Eigenvalue optimisation is made challenging by the fact that eigenvalues are only guaranteed to be differentiable when they are *simple*, i.e., are not repeated. However, minimising the smallest eigenvalue tends to separate it from other eigenvalues, and therefore the issue of non-differentiability becomes less of a concern (Lewis and Overton, 1996). A basic property of graph Laplacian matrices is that both $\lambda_1(L)$ and $\lambda_1(L_{\mathrm{N}})$ are always equal to zero (von Luxburg, 2007). If the similarity function, s, is strictly positive, then $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ are bounded away from zero. Therefore minimising $\lambda_2(\cdot)$ tends to separate it from other eigenvalues, guiding the search to regions of the domain where the objective function is differentiable. Nonetheless, we cannot guarantee that $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ are simple throughout the optimisation procedure. We next provide expressions for the derivatives of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ as functions of $\boldsymbol{\theta}$, when they are simple. Using these we then establish that these eigenvalue objectives are in fact *continuously* differentiable when they are simple.

A useful formulation of eigenvalue derivatives is found in (Magnus, 1985, Th. 1); if λ is a simple eigenvalue of a real symmetric matrix M, then λ is infinitely differentiable on a neighbourhood of M, and the differential at M is given by,

$$d\lambda = u^\top d(M) u, \tag{11}$$

where u is the corresponding eigenvector. As previously mentioned $s(P, i, j)$ is assumed to be continuously differentiable in $P \in \mathbb{R}^{l\times N}$ for fixed $i, j \in \{1 \dots N\}$. The derivative $D_{\boldsymbol{\theta}}\lambda_2(\cdot)$ is given by the $(d-1)\times l$ matrix with i-th column $D_{\boldsymbol{\theta}_i}\lambda_2(\cdot)$, which can be obtained through the chain rule decomposition,

$$D_{\boldsymbol{\theta_i}}\lambda_2(\cdot) = D_P\lambda_2\, D_V P\, D_{\boldsymbol{\theta_i}} V,$$

where $D_{\cdot}$ is the differential operator. Since only the i-th column of V depends on $\boldsymbol{\theta}_i$, and only the i-th row of P depends on V_i, this product can be simplified as

$$D_{\boldsymbol{\theta_i}}\lambda_2(\cdot) = D_{P_i}\lambda_2\, D_{V_i} P_i\, D_{\boldsymbol{\theta_i}} V_i,$$

where P_i is used to denote the i-th row of P, while V_i and $\boldsymbol{\theta}_i$ are, as usual, the i-th columns of V and $\boldsymbol{\theta}$ respectively. By definition $D_{V_i}P_i = X^\top$, while $D_{\boldsymbol{\theta}_i}V_i \in \mathbb{R}^{d\times(d-1)}$ is obtained by differentiating Eq. (8), and is given by,

$$\frac{\partial V(\boldsymbol{\theta})_{ji}}{\partial \boldsymbol{\theta}_{ki}} = \begin{cases} 0, & j<k \\ -\sin(\boldsymbol{\theta}_{ki})\prod_{m=1}^{k-1}\sin(\boldsymbol{\theta}_{mi}), & j=k<d \\ \cos(\boldsymbol{\theta}_{ki})\cos(\boldsymbol{\theta}_{ji})\prod_{m<j,m\neq k}\sin(\boldsymbol{\theta}_{mi}), & k<j<d \\ \cos(\boldsymbol{\theta}_{ki})\prod_{m\neq k}\sin(\boldsymbol{\theta}_{mi}), & j=d. \end{cases} \tag{12}$$

Finally, in the case of the standard Laplacian, we find,

$$\frac{\partial\lambda_2(L)}{\partial P_{ij}} = \frac{1}{2}\sum_{m,n}(u_m - u_n)^2\frac{\partial s(P,m,n)}{\partial P_{ij}}, \tag{13}$$

and for the normalised Laplacian we instead have,

$$\frac{\partial\lambda_2(L_{\mathrm{N}})}{\partial P_{ij}} = \frac{1}{2}\sum_{m,n}\left(\frac{u_m}{\sqrt{d_m}} - \frac{u_n}{\sqrt{d_n}}\right)^2\frac{\partial s(P,m,n)}{\partial P_{ij}} - \lambda\sum_{m,n}\frac{u_m^2}{d_m}\frac{\partial s(P,m,n)}{\partial P_{ij}}. \tag{14}$$

Complete derivations of Eqs. (13) and (14) can be found in Appendix B. The elements of the eigenvector, u, are continuous since we have assumed the corresponding eigenvalue $\lambda_2(\cdot)$ to be simple (Magnus, 1985). In addition we have assumed that s is continuously differentiable. Therefore, the product $D_P\lambda_2\, D_V P\, D_{\boldsymbol{\theta_i}} V$ is continuous in $\boldsymbol{\theta}$, as desired.

If $\lambda_2(\cdot)$ is not simple at $\boldsymbol{\theta}$ the derivative $D_{\boldsymbol{\theta}}\lambda_2(\cdot)$ may not be defined. Gradient sampling (Burke et al., 2006) can be applied to minimising objectives which are not differentiable everywhere. The method works by sampling points within a shrinking radius, ϵ, of the current iterate. The convex hull of the gradients at these sampled points acts as an approximation for the Clarke ϵ-subdifferential, and the minimum norm element of this

convex hull provides an approximate steepest descent direction. This approach is appealing for its broad applicability and almost sure convergence to a local minimum on objectives which are locally Lipschitz and almost everywhere continuously differentiable. However to obtain a search direction at each iteration a quadratic program has to be solved, the formulation of which requires $\mathcal{O}(d)$ gradient computations. This makes the method computationally expensive for large problems. We consider a simple modification which exploits the properties of eigenvalues of graph Laplacians, and uses directional derivatives to derive optimality conditions.

The eigenvalues of a real symmetric matrix can be expressed as the difference between two convex matrix functions (Fan, 1949). Therefore $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ are directionally differentiable everywhere. Overton and Womersley (1993) provide an expression for the directional derivative of the sum of the K largest eigenvalues of a matrix whose elements are continuous functions of a parameter, at a point of non-simplicity of the K-th largest eigenvalue. We discuss the case of $\lambda_2(L(\boldsymbol{\theta}))$, where $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ is analogous. Consider the function $F^K:\mathbb{R}^{N\times N}\to\mathbb{R}$ which takes as input a square matrix and returns the sum of its K largest eigenvalues. Then,

$$\lambda_2(L(\boldsymbol{\theta}))=F^{N-1}(L(\boldsymbol{\theta}))-F^{N-2}(L(\boldsymbol{\theta})).$$

Now consider a $\boldsymbol{\theta}$ such that,

$$\begin{aligned}&\lambda_N(L(\boldsymbol{\theta}))\geqslant\cdots\geqslant\lambda_{N-r+1}(L(\boldsymbol{\theta}))>\\&\lambda_{N-r}(L(\boldsymbol{\theta}))=\cdots=\lambda_{N-K+1}(L(\boldsymbol{\theta}))=\\&\cdots=\lambda_{N-r-t+1}(L(\boldsymbol{\theta}))\\&>\lambda_{N-r-t}(L(\boldsymbol{\theta}))\geqslant\cdots>\lambda_1(L(\boldsymbol{\theta}))=0.\end{aligned}$$

That is, the K-th largest eigenvalue has multiplicity t and $K-r$ of the repeated eigenvalues are included in the sum $F^K(L(\boldsymbol{\theta}))$. Overton and Womersley (1993) have shown that the directional derivative of $F^K(L(\boldsymbol{\theta}))$ in the direction $\boldsymbol{\psi}$, $dF^K(L(\boldsymbol{\theta});\boldsymbol{\psi})$, is equal to,

$$F^r\left(\sum_{i=1}^{d-1}\sum_{j=1}^{l}\boldsymbol{\psi}_{ij}R^\top L_{ij}R\right)+F^{K-r}\left(\sum_{i=1}^{d-1}\sum_{j=1}^{l}\boldsymbol{\psi}_{ij}Q^\top L_{ij}Q\right),$$

where $L_{ij}=\partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta}_{ij}$, the j-th column of the matrix $R\in\mathbb{R}^{N\times r}$ is equal to the eigenvector associated with the j-th largest eigenvalue of $L(\boldsymbol{\theta})$, and the j-th column of the matrix $Q\in\mathbb{R}^{N\times t}$ is equal to the eigenvector associated with the $(r+j)$-th largest eigenvalue of $L(\boldsymbol{\theta})$. The directional derivative of $\lambda_2(L(\boldsymbol{\theta}))$ in the direction $\boldsymbol{\psi}$ is thus,

$$\begin{aligned}d\lambda_2(L(\boldsymbol{\theta});\boldsymbol{\psi})=&dF^{N-1}(L(\boldsymbol{\theta});\boldsymbol{\psi})-dF^{N-2}(L(\boldsymbol{\theta});\boldsymbol{\psi})\\=&\lambda_1\left(\sum_{i=1}^{d-1}\sum_{j=1}^{l}\boldsymbol{\psi}_{ij}Q^\top L_{ij}Q\right),\end{aligned}\tag{15}$$

where the columns of Q are given by the complete set of eigenvectors for the eigenvalue $\lambda=\lambda_2(L(\boldsymbol{\theta}))$.

3.2 Minimising $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$.

Applying standard gradient descent methods to functions which are almost everywhere differentiable can result in convergence to sub-optimal points (Wolfe, 1972). This occurs when the method for determining the gradient is applied at a point of non-differentiability and produces a non-descent direction. In this case the algorithm cannot reduce the objective function value and terminates at a point that is not necessarily a local minimum. The second eigenvalues of the graph Laplacian matrices, while not necessarily differentiable everywhere, benefit from the fact that their minimisation tends to separate them from other eigenvalues. Thus a standard gradient descent algorithm performs well on these objectives, very often converging to locally optimal solutions. Our approach for minimising $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$, therefore assumes them to be continuously differentiable until there is evidence that this assumption fails. Only then is it necessary to use the computationally more expensive gradient sampling algorithm to identify a descent direction.

Our approach is summarised in Algorithm 1. Once again we discuss only $\lambda_2(L(\boldsymbol{\theta}))$ explicitly, noting that the methodology for minimising $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ is equivalent, with the only difference being in the computation of the gradients and directional derivatives.

At each iteration a standard gradient-based algorithm with inexact line-search is used to minimise the objective function using the formulation for the gradient presented in Section 3.1. When this algorithm terminates, say with solution $\boldsymbol{\theta}^\star$, either the magnitude of the computed gradient is below a threshold, or a sufficient decrease in the objective function value was not feasible. We then need to verify whether $\boldsymbol{\theta}^\star$ is a local minimum. If $\lambda_2(L(\boldsymbol{\theta}^\star))$ is simple then $\lambda_2(\cdot)$ is continuously differentiable at $\boldsymbol{\theta}^\star$, and therefore $\boldsymbol{\theta}^\star$ is close to a local minimiser. In this case the algorithm terminates. On the other hand, if $\lambda_2(L(\boldsymbol{\theta}^\star))$ is not simple, then $\boldsymbol{\theta}^\star$ may or may not be a local minimiser. The directional derivative formulation in Eq. (15) provides a computationally efficient way to determine if a descent direction from $\boldsymbol{\theta}^\star$ exists. In particular, if at $\boldsymbol{\theta}^\star$, $Q^\top L_{ij}Q\approx\mathbf{0}$ for all pairs, i,j, then the directional derivative $d\lambda_2(L(\boldsymbol{\theta}^\star);\boldsymbol{\psi})$ is approximately zero for all directions $\boldsymbol{\psi}$. In this case the algorithm terminates as $\boldsymbol{\theta}^\star$ is sufficiently close to a local minimiser. If this condition is not met a descent directions exists, that is $\exists\boldsymbol{\psi}\in\Theta$ s.t. $\lambda_1\left(\sum_{i=1}^{d-1}\sum_{j=1}^{l}\boldsymbol{\psi}_{ij}Q^\top L_{ij}Q\right)<0$. At this point a single

step of the gradient sampling algorithm is performed. As in the standard gradient sampling algorithm (Burke et al., 2006) the magnitude of the sampling radius ϵ is progressively reduced until a valid descent direction is identified, or the radius is reduced beyond a user-specified threshold ϵ_f. In the latter case the current solution is considered sufficiently close to a local minimiser and the algorithm terminates. In the former case, once a valid descent direction is identified $\boldsymbol{\theta}^\star$ is updated using an inexact line-search algorithm.

Termination under any of the above criteria indicates the identification of a local minimiser. Moreover, the convergence of the method is guaranteed under the same analyses as for gradient descent on smooth functions (Nocedal and Wright, 2006) and gradient sampling (Burke et al., 2006).

Algorithm 1: Minimising $\lambda_2(L(\boldsymbol{\theta}))$

Input: Initial projection $\boldsymbol{\theta}_0$, optimality tolerance τ, initial sampling radius for gradient sampling ϵ_0, minimum sampling radius ϵ_f, radius reduction factor η, number of sampled gradients n_g
Output: Optimal projection $\boldsymbol{\theta}^\star$

$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}_0$
$\epsilon \leftarrow \epsilon_0$
while $\epsilon > \epsilon_f$ **do**
  # apply standard gradient descent to convergence
  $\boldsymbol{\theta}^\star \leftarrow \text{GradientDescentSolution}(\boldsymbol{\theta}^\star)$
  # check for optimality of the solution
  if $\lambda_2(L(\boldsymbol{\theta}^\star))$ is simple **or** $\max_{i,j} |Q^\top L_{ij} Q| < \tau$ **then**
    return $\boldsymbol{\theta}^\star$
  else
    # obtain gradients at points sampled uniformly in a
    # ball of radius ϵ around the current solution
    for $i = 1 \dots n_g$ **do**
      $\boldsymbol{\theta}_i \sim U(\mathcal{B}_\epsilon(\boldsymbol{\theta}^\star))$
      $\boldsymbol{\Gamma}_i \leftarrow D_{\boldsymbol{\theta}} \lambda_2(L(\boldsymbol{\theta}))|_{\boldsymbol{\theta}=\boldsymbol{\theta}_i}$
    end for
    # obtain the search direction
    $\boldsymbol{\Gamma}_s \leftarrow \text{argmin}_{\boldsymbol{\Gamma} \in \mathbf{conv}(\{\boldsymbol{\Gamma}_1, \dots, \boldsymbol{\Gamma}_{n_g}\})} \|\boldsymbol{\Gamma}\|_F$
    # if the magnitude of the search direction is below
    # the optimality threshold decrease sampling radius
    if $\|\boldsymbol{\Gamma}_s\|_F < \tau$ **then**
      $\epsilon \leftarrow \eta\epsilon$
    else
      # update solution using inexact line-search
      $\nu^\star \leftarrow \approx \text{argmin}_{\nu > 0} \lambda_2(L(\boldsymbol{\theta}^\star - \nu \boldsymbol{\Gamma}_s))$
      $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \nu^\star \boldsymbol{\Gamma}_s$
    end if
  end if
end while
return $\boldsymbol{\theta}^\star$

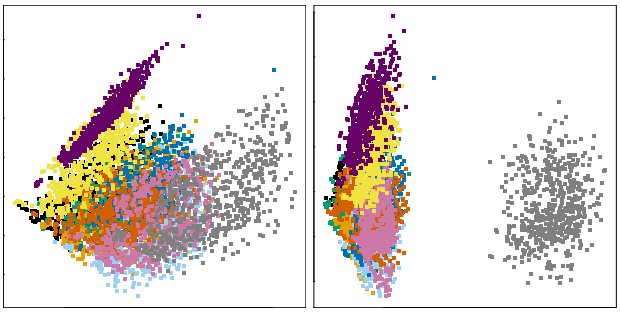
A brief derivation of the computational complexity of each iteration of the method is provided in Appendix C. Each step in the standard gradient descent algorithm requires $\mathcal{O}(lN(N + d(d-1)))$ operations. The gradient sampling step requires $\mathcal{O}(d)$ gradient computations, therefore having complexity $\mathcal{O}(dlN(N + d(d-1)))$. The complexity of computing the optimality conditions using directional derivatives is similar, requiring $\mathcal{O}(t^2 lN(n + d(d-1)))$ operations, where t is the multiplicity of the eigenvalue $\lambda = \lambda_2(L(\boldsymbol{\theta}))$. Our experience with this method indicates that the algorithm almost always terminates with $\lambda_2(\cdot)$ being simple, without the need for any gradient sampling or directional derivative computations.

Figure 1 shows two dimensional plots of three of the datasets used in our experiments in Section 6. The left plots show projections of the data onto the first two principal components. The right plots show the optimal projections obtained by minimising the objective in (10), using the normalised Laplacian. Figure 1(a) shows an example where the principal components do not allow a clear identification of any of the clusters, whereas the optimal projection for spectral clustering clearly admits a strong separation of one of the clusters from the remainder. Figures 1(b) and 1(c) show that when there is moderate separability of clusters within the PCA projection used for initialisation, optimisation of the spectral connectivity increases the separabilty and makes the individual clusters more compact within the projected space.

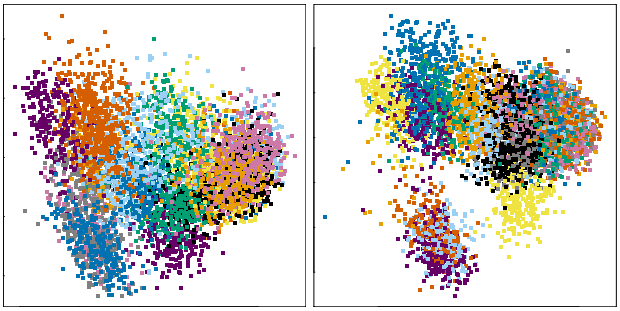
4 Connection to Maximum Margin Hyperplanes

Maximum margin hyperplanes have become a unifying principle in data classification tasks. Starting with the fully supervised problem using support vector machines (Vapnik and Kotz, 1982), the methodology has been extended to semi-supervised classification (Joachims, 1999), and more recently to the problem of maximum margin clustering (Xu et al., 2004; Zhang et al., 2009).

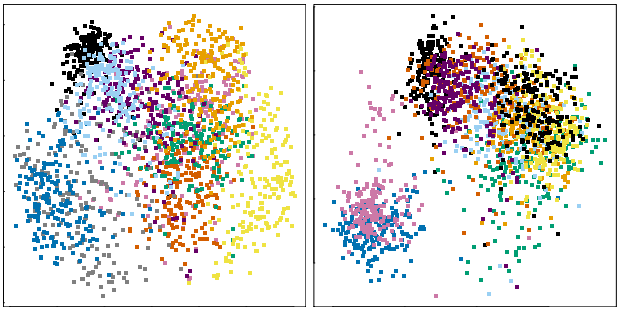
In this section, we establish a connection between the optimal univariate projection for spectral clustering and maximum margin hyperplanes for clustering. In particular, we show that under suitable conditions, as the scaling parameter, σ, tends to zero, the optimal univariate projection for spectral bi-partitioning converges to the vector normal to the largest margin hyperplane through the data. This establishes a theoretical connection between separability measured by spectral graph theory, and standard notions of separation in terms of the Euclidean metric. Connections between maximum margin hyperplanes and Bayes optimal hyperplanes (Tong and Koller, 2000) as well as minimum density hyperplanes (Pavlidis et al., 2016) have previously been established. The result we discuss herein



(a) Yale Faces B



(b) Isolet



(c) Multiple Feature Digits

Fig. 1 Two dimensional projections of publicly available datasets. PCA (left) and optimal projection for spectral clustering (right).

therefore connects spectral connectivity to these objectives as well.

In this section we use the notation $v(\boldsymbol{\theta})$ instead of $V(\boldsymbol{\theta})$ to stress that the we are concerned with univariate projections. A hyperplane is a translated subspace of co-dimension 1, and can be parameterised by a vector $v \in \mathbb{R}^d \setminus \{0\}$ and a scalar b as the set $H(v,b) = \{x \in \mathbb{R}^d \big| v^\top x = b\}$. No generality is lost if v is assumed to have unit norm, thus the same parameterisation by $\boldsymbol{\theta}$ can be used. For a finite set of points $\mathcal{X}$ in $\mathbb{R}^d$, the *margin* of hyperplane $H(v(\boldsymbol{\theta}),b)$ w.r.t. $\mathcal{X}$ is the minimal Euclidean distance between $H(v(\boldsymbol{\theta}),b)$ and $\mathcal{X}$,

$$\text{margin}(v(\boldsymbol{\theta}),b) = \min_{x \in \mathcal{X}} |v(\boldsymbol{\theta})^\top x - b|. \tag{16}$$

The set $\boldsymbol{\Delta}(\boldsymbol{\theta})$ again plays an important role as in many cases the largest margin hyperplane through a set of data separates only a few points from the rest, making it meaningless for the purpose of clustering. For the theory presented herein we consider an arbitrary convex and compact set $\boldsymbol{\Delta} \subset \mathbb{R}^d$ and define $\boldsymbol{\Delta}(\boldsymbol{\theta})$ to be the projection of $\boldsymbol{\Delta}$ onto $v(\boldsymbol{\theta})$. What we in fact show in this section is that there exists a set $\boldsymbol{\Delta}' \subset \boldsymbol{\Delta}$ satisfying $\boldsymbol{\Delta}' \cap \mathcal{X} = \boldsymbol{\Delta} \cap \mathcal{X}$, such that, as the scaling parameter tends to zero, the optimal projections for $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{N}}(\boldsymbol{\theta}))$ converge to the vector admitting the largest margin hyperplane that intersects $\boldsymbol{\Delta}'$. The distinction between the largest margin hyperplane intersecting $\boldsymbol{\Delta}'$ and that intersecting $\boldsymbol{\Delta}$ is scarcely of practical relevance, but plays an important role theoretically. It accounts for situations when the largest margin hyperplane intersecting $\boldsymbol{\Delta}$ lies close to its boundary and the distance between the hyperplane and the nearest point outside $\boldsymbol{\Delta}$ is larger than to the nearest point inside $\boldsymbol{\Delta}$. Aside from this very specific case, the two solutions in fact coincide.

The following theorem is the main result of this section. The proof and supporting results are provided in Appendix D. The result holds for all similarities in which the function k, in Eq. (9), satisfies the tail condition $\lim_{x \to \infty} k((1+\epsilon)x)/k(x) = 0$ for all $\epsilon > 0$. This condition is satisfied by functions with exponentially decaying tails, including the popular Gaussian and Laplace kernels, but not those with polynomially decaying tails.

The proof of the result relies on obtaining upper and lower bounds on the magnitude of $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\text{N}}(\boldsymbol{\theta}))$ which depend essentially on $k(M/\sigma)$, where M is the largest gap between consecutive points in $\mathcal{P}(\boldsymbol{\theta})$. Notice that M is equal to twice the maximum margin of all hyperplanes orthogonal to $v(\boldsymbol{\theta})$. These bounds show immediately that as σ approaches zero, if $\lambda_2(L(\boldsymbol{\theta}_1)) < \lambda_2(L(\boldsymbol{\theta}_2))$ (or $\lambda_2(L_{\text{N}}(\boldsymbol{\theta}_1)) < \lambda_2(L_{\text{N}}(\boldsymbol{\theta}_2))$) then the maximum margin of all hyperplanes orthogonal to $v(\boldsymbol{\theta}_1)$ is greater than the maximum margin of all hyperplanes orthogonal to $v(\boldsymbol{\theta}_2)$. The convergence of the optimal projection itself to the vector normal to the maximum margin hyperplane uses a property of the maximum margin hyperplane established by Pavlidis et al. (2016).

Theorem 2 *Let $\mathcal{X} = \{x_1, ..., x_N\}$ be a finite set of points in $\mathbb{R}^d$ and suppose that there is a unique hyperplane, which can be parameterised by $(v(\boldsymbol{\theta}^\star), b^\star)$, intersecting $\boldsymbol{\Delta}'$ and attaining maximal margin on $\mathcal{X}$. Let $k : \mathbb{R}_+ \to$*

$\mathbb{R}_+$ *be decreasing, positive and satisfy* $\lim_{x\to\infty} k((1+\epsilon)x)/k(x)=0$ *for all* $\epsilon>0$. *For* $\sigma>0$ *define*

$$\boldsymbol{\theta}_\sigma := argmin_{\boldsymbol{\theta}\in\Theta}\lambda_2(L(\boldsymbol{\theta},\sigma)),$$

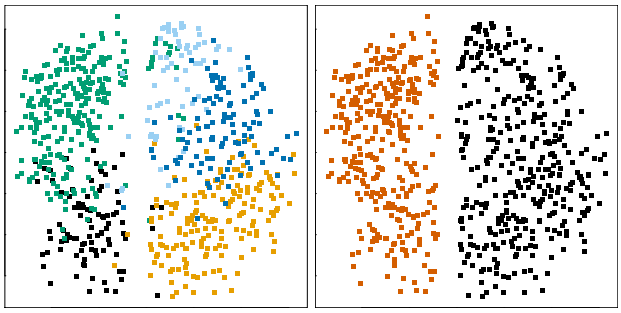
$$\boldsymbol{\theta}_\sigma^N := argmin_{\boldsymbol{\theta}\in\Theta}\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta},\sigma)),$$

where there is now an explicit dependence on the scaling parameter, σ. *Then,*

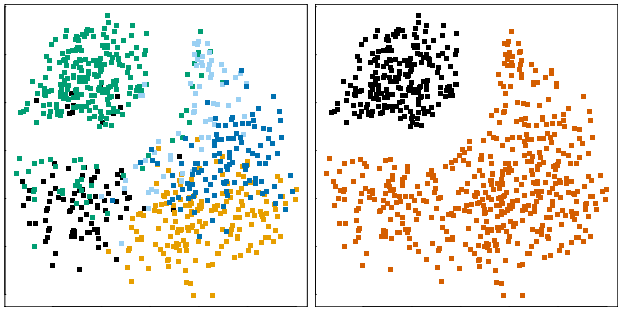
$$\lim_{\sigma\to 0^+} v(\boldsymbol{\theta}_\sigma) = \lim_{\sigma\to 0^+} v(\boldsymbol{\theta}_\sigma^N) = v(\boldsymbol{\theta}^\star).$$

We note that the same result holds when using the Euclidean metric. In this case the optimal projection based on spectral connectivity converges to the vector normal to the maximum margin hyperplane through the data. The importance of constraining the maximum margin hyperplane to avoid separating only outliers was also observed by Xu et al. (2004) and Zhang et al. (2009).

While the above result is only established for univariate projections, we have observed empirically that if a decreasing sequence of scaling parameters is employed for a multivariate projection, then the projected data, $\mathcal{P}(\boldsymbol{\theta})$, tend to exhibit large Euclidean separation. This is illustrated in Figure 2 which shows two dimensional plots of the 72 dimensional yeast cell cycle analysis dataset (Bache and Lichman, 2013). The left plots show the true clusters, while the right plots show the cluster assignments made by the algorithm. In Figure 2(a) the horizontal axis corresponds to the optimal projection obtained by minimising $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ for a decreasing sequence of scaling parameters, while the vertical axis is the direction of maximum variance orthogonal to this vector. Figure 2(b) instead shows the result of two dimensional projection pursuit for a decreasing sequence of scaling parameters.



(a) One dimensional projection pursuit



(b) Two dimensional projection pursuit

Fig. 2 Large Euclidean separation of yeast cell cycle dataset by decreasing the scaling parameter during one and two dimensional projection pursuit.

5 Speeding up Computation

Each step in the projection pursuit algorithm involves the solution of an eigen problem which requires $\mathcal{O}(N^2)$ operations. In this section we discuss how preprocessing a dataset using *microclusters* (Zhang et al., 1996) can reduce this cost significantly, and derive theoretical bounds on the approximation error. Microclusters are small clusters of data which can in turn be clustered to obtain a complete clustering of a data set. A microcluster based approach to reduce the computational cost of the standard spectral clustering algorithm has been previously proposed by Yan et al. (2009). In this work we use microclusters to obtain an approximation of the optimisation surface for projection pursuit which is significantly less expensive to explore.

In the microcluster approach, the data set is replaced by m points, $\{c_1,\ldots,c_m\}$, which represent the centres of a m-way clustering of $\mathcal{X}$. By projecting these microcluster centres during projection pursuit rather than the data the computational cost associated with each eigen problem is reduced to $\mathcal{O}(m^2)$. If we define the radius, ρ, of a cluster C to be the largest distance between any one of its members and its centre,

$$\rho(C)=\max_{x\in C}\left\|x-\frac{1}{|C|}\sum_{x\in C}x\right\|, \tag{17}$$

then we expect the approximation error to be small whenever the microcluster radii are small. This relationship is shown in the following lemma. The proof of the lemma, which is given in Appendix D, relies on a result from matrix perturbation theory for diagonally dominant matrices (Ye, 2009, Th. 3.3)

Lemma 3 *Let* $\mathcal{C}=C_1,\ldots,C_m$ *be a* m*-way clustering of* $\mathcal{X}$ *with centres* $c_1,\ldots,c_m$, *radii* $\rho_1,\ldots,\rho_m$ *and counts* $n_1,\ldots,n_m$. *For* $\boldsymbol{\theta}\in\Theta$ *define* $N(\boldsymbol{\theta}),B(\boldsymbol{\theta})\in\mathbb{R}^{m\times m}$ *where*

$N(\boldsymbol{\theta})$ is the diagonal matrix with,

$$N(\boldsymbol{\theta})_{i,i}=\sum_{j=1}^{m} n_j s(P^c(\boldsymbol{\theta}),i,j),$$

and

$$B(\boldsymbol{\theta})_{i,j}=\sqrt{n_i n_j}\, s(P^c(\boldsymbol{\theta}),i,j),$$

where $P^c(\boldsymbol{\theta})=\{V(\boldsymbol{\theta})^\top c_1,...,V(\boldsymbol{\theta})^\top c_m\}$ are the projected microcluster centres and the similarities are given by $s(P^c(\boldsymbol{\theta}),i,j)=k(d(V(\boldsymbol{\theta})^\top c_i,V(\boldsymbol{\theta})^\top c_j)/\sigma)$, and $k(x)$ is positive and non-increasing for $x\geq 0$. Then,

$$\frac{|\lambda_2(L(\boldsymbol{\theta}))-\lambda_2(N(\boldsymbol{\theta})-B(\boldsymbol{\theta}))|}{\lambda_2(L(\boldsymbol{\theta}))} \leqslant \max_{i\neq j}\max\left\{1-\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)}, \frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}-1\right\},$$

where $D_{ij}=d(V(\boldsymbol{\theta})^\top c_i,V(\boldsymbol{\theta})^\top c_j)$ and $(x)^+=\max\{0,x\}$.

The bound in the above lemma depends on $\boldsymbol{\theta}$ via the quantity D_{ij}. Uniform bounds can be derived for specific functions, k. For example, if using the Gaussian kernel, $k=\exp(-x^2/2)$, then we can show that

$$\frac{|\lambda_2(L(\boldsymbol{\theta}))-\lambda_2(N(\boldsymbol{\theta})-B(\boldsymbol{\theta}))|}{\lambda_2(L(\boldsymbol{\theta}))} \leqslant \max_{i\neq j}\exp\left(\frac{(\rho_i+\rho_j)^2+2(\rho_i+\rho_j)\mathrm{Diam}(\mathcal{X})}{2\sigma^2}\right)-1.$$

If k is the Laplace kernel, $k(x)=\exp(-|x|)$, then we instead have

$$\frac{|\lambda_2(L(\boldsymbol{\theta}))-\lambda_2(N(\boldsymbol{\theta})-B(\boldsymbol{\theta}))|}{\lambda_2(L(\boldsymbol{\theta}))} \leqslant \max_{i\neq j}\exp\left(\frac{\rho_i+\rho_j}{\sigma}\right)-1.$$

Clearly if the radii of the microclusters are small relative to the scale parameter, σ, then these bounds are close to zero. However the uniform bounds are pessimistic, and to obtain a reasonable bound on the approximation surface, as many as $m\approx 0.6N$ might be needed, leading to only a threefold speed up. We have observed empirically, however, that even for $m=0.1N$ (and sometimes lower) one still obtains a close approximation of the optimisation surface. This renders the projection pursuit of the order of 100 times faster. While bounds of the above type are not verifiable for $L_{\mathrm{N}}(\boldsymbol{\theta})$ since this matrix is not diagonally dominant, a similar degree of agreement between the true and approximate eigenvalues has been observed.

Once an optimal projection has been determined, the corresponding bi-partition needs to be established. We again use the microclusters to determine this partition. Let $\mathcal{P}(\boldsymbol{\theta})'=\{V(\boldsymbol{\theta})^\top c_1,\ V(\boldsymbol{\theta})^\top c_1,\ \ldots, V(\boldsymbol{\theta})^\top c_m,\ V(\boldsymbol{\theta})^\top c_m\}$, where each $V(\boldsymbol{\theta})^\top c_i$ is repeated n_i times. $\mathcal{P}(\boldsymbol{\theta})'$ therefore represents an approximation of the projected data set, where each datum is replaced by the center of its assigned microcluster. It is straightforward to verify that if $u^C\in\mathbb{R}^m$ is the second eigenvector of $N(\boldsymbol{\theta})-B(\boldsymbol{\theta})$, then the vector $u\in\mathbb{R}^N$, with $u_i=u^C_j/\sqrt{n_j}$ for all i s.t. x_i is assigned to microcluster j, is the second eigenvector of the Laplacian of $\mathcal{P}(\boldsymbol{\theta})'$. The vector u therefore represents an approximation of the second eigenvector of $L(\boldsymbol{\theta})$. In case of the normalised Laplacian the $m\times m$ matrix is given by the normalised Laplacian of the graph of $\mathcal{P}^c(\boldsymbol{\theta})$ with similarities given by $n_i n_j s(P^c(\boldsymbol{\theta}),i,j)$. This matrix has the same structure as the original normalised Laplacian, the only difference being the introduction of the factors n_i,n_j. The approximation of the second eigenvector of $L_{\mathrm{N}}(\boldsymbol{\theta})$ is again given by $u\in\mathbb{R}^N$ satisfying $u_i=u^C_j/\sqrt{n_j}$ whenever x_i is in microcluster j, where $u^C\in\mathbb{R}^m$ is the second eigenvector of the normalised Laplacian of the graph of $\mathcal{P}^c(\boldsymbol{\theta})$. This approximate eigenvector is then used to determine the partition of the data.

6 Practical Implementation and Experimental Results

We have found that projection pursuit based on both $\lambda_2(L(\boldsymbol{\theta}))$ and $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ leads to high quality clustering results. However, we have observed empirically that the minimisation of $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ is more robust to varying parameter settings, and we recommend using this objective. Our complete clustering algorithm, which we will refer to as Spectral Clustering Projection Pursuit (SCPP), is summarised in Algorithm 2[1]. Starting with all the data in a single cluster, we recursively bi-partition the data until we have the desired number of clusters. At each iteration we simply split the largest cluster in the current partition. To split a cluster, we first obtain m microclusters from it, for which we use the K-means algorithm. We then apply Algorithm 1 to obtain the optimal projection, $\boldsymbol{\theta}^\star$, based on Eq. (10). Recall that the normalised Laplacian based on (weighted) projected microcluster centers $\mathcal{P}^c(\boldsymbol{\theta})=\{V(\boldsymbol{\theta})^\top c_1,...,V(\boldsymbol{\theta})^\top c_m\}$ is given by

$$L_{\mathrm{N}}(\boldsymbol{\theta})=D(\boldsymbol{\theta})^{-1/2}L(\boldsymbol{\theta})D(\boldsymbol{\theta})^{-1/2} =I-D(\boldsymbol{\theta})^{-1/2}A(\boldsymbol{\theta})D(\boldsymbol{\theta})^{-1/2},$$

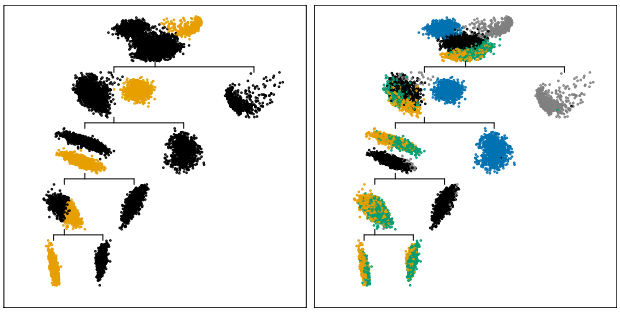
[1] An `R` implementation of the SCPP algorithm is available at `https://github.com/DavidHofmeyr/SCPP`

Algorithm 2: SCPP

Input: Dataset $\mathcal{X}$, number of clusters K
Output: Partition Π of $\mathcal{X}$ into K clusters
 # Initialise Π as the set containing $\mathcal{X}$
$\Pi \leftarrow \{\mathcal{X}\}$
while $|\Pi| < K$ **do**
  # Select the next cluster to split, $\mathcal{C}'$
  $\mathcal{C}' \leftarrow \text{argmax}_{\mathcal{C}\in\Pi} |\mathcal{C}|$
  # Obtain centers and counts from microclustering of $\mathcal{C}'$
  $[\{c_1 \ldots c_m\}, \{n_1 \ldots n_m\}] \leftarrow Microcluster(\mathcal{C}')$
  # Optimise projection for spectral clustering of $\mathcal{P}^c(\boldsymbol{\theta})$
  $\boldsymbol{\theta}^\star \leftarrow \text{argmin}_{\boldsymbol{\theta}} \lambda_2(L_N(\boldsymbol{\theta})) + \omega \sum_{i\neq j} (V(\boldsymbol{\theta})_i^\top V(\boldsymbol{\theta})_j)^2$
  # Find the first two eigenvectors of $L_N(\boldsymbol{\theta}^\star)$
  $U^c \leftarrow \text{argmin}_U \text{trace}(U^\top L_N(\boldsymbol{\theta}^\star) U)$ s.t. $U^\top U = I$
  # Get approximate eigenvectors of Laplacian of $V(\boldsymbol{\theta}^\star)^\top \mathcal{C}'$
  $U \leftarrow U_i = U^c_j / \sqrt{n_j} \iff x_i \in \text{microcluster } j$
  # Normalise the rows of U
  $U_i \leftarrow U_i / \|U_i\|, \ \forall\ i = 1, \ldots, N$
  # Bi-partition rows of U using k-means
  $[\mathcal{U}_1, \mathcal{U}_2] \leftarrow K\text{-means}(U, 2)$
  # Obtain corresponding split of $\mathcal{C}'$
  $\mathcal{C}_1 \leftarrow \cup_{i:U_i\in\mathcal{U}_1}\{x_i\}, \mathcal{C}_2 \leftarrow \cup_{i:U_i\in\mathcal{U}_2}\{x_i\}$
  # Update overall partition Π
  $\Pi \leftarrow (\Pi \setminus \{\mathcal{C}'\}) \cup \{\mathcal{C}_1, \mathcal{C}_2\}$
end while
return Π

where $A(\boldsymbol{\theta})_{ij} = n_i n_j s(P^C(\boldsymbol{\theta}), i, j)$ and $D_{ii} = \sum_{j=1}^m A(\boldsymbol{\theta})_{ij}$. To obtain a bi-partition of the cluster we use the method recommended by Ng et al. (2002). For this we obtain the first two eigenvectors of $L_N(\boldsymbol{\theta}^\star)$ as the matrix $U^c \in \mathbb{R}^{m\times 2}$. From these we obtain the approximate eigenvectors of the Laplacian of the complete set of projected points as the matrix $U \in \mathbb{R}^{N\times 2}$, with i-th row equal to the j-th row of U^c divided by $\sqrt{n_j}$ for each x_i in microcluster j. We then normalise the rows of U and apply K-means for $K = 2$. For the sake of easier interpretability we make our algorithm completely deterministic by initialising all implementations of K-means as follows. We select the first center to be the point furthest from the mean of the data. We then iteratively add to the set of initial centers the furthest point from the current set.

The clustering model obtained by the SCPP algorithm has a binary tree structure, as illustrated in Figure 3. The figure shows a divisive hierarchical clustering of the 256 dimensional phoneme dataset (Hastie et al., 2009). Each scatter plot shows the data assigned to the corresponding node in the model projected into the optimal subspace based on the minimisation of the second eigenvalue of the Laplacian matrix. In Figure 3(a) the colours indicate the binary partitions made by the SCPP algorithm, while in Figure 3(b) the colours show the true cluster labels of the data. The model has accurately partitioned the clusters; indicated by the fact that the leaf nodes each contain primarily data of a single cluster, and aside from the two clusters arising in the bottom most level of the hierarchy no cluster is split among multiple leaves.



(a) Without cluster labels (b) With true cluster labels

Fig. 3 Hierarchical clustering model obtained by SCPP on phoneme dataset

6.1 Parameter Settings for SCPP

For the experiments herein, we use the following settings. In all cases the data dependent settings are determined for each partition using the subset of the data being split. We set l, the dimension of the projection, to 2 as this is the lowest number of dimensions which admits non-linear separation of clusters. We initialise the projection pursuit using the first two principal components. We have found that this often leads to higher quality solutions compared to random initialisations. Experiments with higher dimensional projections ($l > 2$) have not shown substantially improved performance. Similarities between projected points are determined using the Gaussian kernel, i.e.,

$$A(\boldsymbol{\theta})_{i,j} = n_i n_j \exp\left(\frac{d(V(\boldsymbol{\theta})^\top c_i, V(\boldsymbol{\theta})^\top c_j)^2}{2\sigma^2}\right),$$

where c_i and c_j are the centers of the i-th and j-th microclusters respectively, and n_i and n_j are the sizes of these microclusters. The scale parameter is set to

$$\sigma = \sqrt{\bar{\lambda}} \left(\frac{4}{3N}\right)^{\frac{1}{4+d^\star}},$$

where $\bar{\lambda}$ is the average of the largest $d^\star$ eigenvalues of the covariance matrix, Σ, and $d^\star = \min\{20, |\lambda(\Sigma) \cap [1, \infty)|\}$. Here $\lambda(\Sigma)$ is the set of eigenvalues of the covariance matrix, and thus $|\lambda(\Sigma) \cap [1, \infty)|$ is the number of eigenvalues of Σ greater than or equal to one. This latter term has been used to estimate the intrinsic

dimensionality of a dataset whose columns have been standardised to have unit variance (Kaiser, 1960). We choose to place an upper bound on this value as for some very high dimensional datasets the resulting value of σ was extremely small relative to the actual scale of the data. The precise value of this upper bound does not affect performance of the method substantially. Setting σ in this way captures the scale of the data through the factor $\sqrt{\overline{\lambda}}$, while $\left(\frac{4}{3N}\right)^{\frac{1}{4+d^*}}$ is borrowed from kernel density bandwidth estimation, where connections between spectral clustering and kernel density estimation have been established (Trillos et al., 2016; Hofmeyr, 2017).

Now, recall that we use $\boldsymbol{\Delta}(\boldsymbol{\theta})$ to mitigate the influence of outliers. We define $\boldsymbol{\Delta}(\boldsymbol{\theta}) = \Delta_1 \times \cdots \times \Delta_l$, where $\Delta_i = [\mu_i - \beta\sigma_i, \mu_i + \beta\sigma_i]$; μ_i and σ_i are the mean and standard deviation of the i-th component of the projected data respectively; and $\beta \geqslant 0$ controls the size of $\boldsymbol{\Delta}(\boldsymbol{\theta})$. Rather than attempting to define a single value of β which is appropriate for all datasets, we initialise β to a large value, $\beta = 3$, and decrease β until the induced bi-partition is sufficiently balanced. For this we define a minimum cluster size, the average cluster size in the complete clustering solution divided by 5. That is, we decrease β until the smaller of the two clusters contains at least $\frac{N}{5K}$ points, where N is the number of data in the complete dataset being clustered. Again the precise value is not important for performance. We select a value which is small enough to allow the detected clusters to vary greatly in size, but large enough that the performance of the method is not compromised when clustering datasets containing a substantial number of outliers. Note that in general we do not have to execute the optimisation of $\boldsymbol{\theta}$ to convergence for each value of β, since a few iterations generally suffice to determine if the optimisation is focusing on outliers. We therefore terminate the optimisation as soon as the induced partition does not meet the desired balance, reduce β, and reinitialise.

The setting of the parameter ω, which controls the penalisation of non-orthogonal projections, does not affect the result substantially provided it is relatively larger than the eigenvalues being optimised. We simply set $\omega = 1$ since $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ is bounded above by 1.

Finally, for our experiments we use a small number of microclusters, $m = 200$. A sensitivity study presented in Section 6.4.2 using simulated data shows that even for data sets of up to 10 000 points and in 50 dimensions, 200 microclusters are sufficient to obtain high quality clustering results.

6.2 Competing Approaches

We compare our approach against existing dimension reduction methods for clustering, where the final clustering result is determined using spectral clustering. We use SC to refer to spectral clustering applied to the original data, and $\mathrm{SC_{PC}}$ and $\mathrm{SC_{IC}}$ to refer to spectral clustering applied to Principal and Independent Component projections of the data respectively. DRSC refers to dimensionality reduction for spectral clustering, proposed by Niu et al. (2011). For $\mathrm{SC_{PC}}$, $\mathrm{SC_{IC}}$ and DRSC we consider $K-1$ dimensional projections, as suggested by Niu et al. (2011). These approaches all directly seek a K way partition of the data, which was obtained using the method of Ng et al. (2002). In addition we compare with the approach of recursively dividing the data in the same manner as SCPP but using the first two principal component projections. Since SCPP is initialised using principal components, this allows us to more directly measure the benefit of optimising the projection before applying spectral clustering. We will refer to this method as $\mathrm{SC_{PC}^{rec}}$.

For all competing approaches except $\mathrm{SC_{PC}^{rec}}$ we compute clustering results for all values of σ in the set $\{0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, \ 100, 200\}$, and select the solution which gives the lowest cluster distortion measure. This selection criterion is recommended by Ng et al. (2002) and Niu et al. (2011). We also compute the clustering result for the local scaling approach of Zelnik-Manor and Perona (2004). We report the highest performance of these two in each case. We also provide DRSC with a warm start via PCA as this improved performance over a random initialisation, and provides a fair comparison. For $\mathrm{SC_{PC}^{rec}}$, in order to assess the effect of the proposed projection pursuit as directly as possible, we use the same parameters as in SCPP.

The connection between optimal projections for spectral clustering and maximum margin clustering, established in Section 4, also leads us to investigate the effectiveness of the proposed approach for finding large margin clustering solutions. To obtain large margin solutions we apply Algorithm 1 repeatedly for a decreasing sequence of scaling parameters until convergence of the projection matrix. We compare this approach with two dedicated maximum margin clustering algorithms. The iterative Support Vector Regression algorithm (Zhang et al., 2009, iSVR) alternates between applying Support Vector Regression (SVR) using the assigned cluster labels, and updating the labels based on the predictions of the resulting SVR model. The Cutting Plane Maximum Margin Clustering algorithm (Wang et al., 2010, CPMMC) uses a series of convex relaxations of the non-convex maximum margin clustering objective.

The relaxed problems use cutting planes to progressively improve the approximation of the original problem. For the proposed approach we consider both one- and two-dimensional projections, to correspond with a linear and non-linear kernel respectively in the context of support vector methods. For iSVR we use both the linear and Gaussian kernels, while for CPMMC we use only the linear kernel, as in Wang et al. (2010).

It is important to note that these dedicated maximum margin clustering algorithms use a soft-margin, which is a relaxation of the hard margin solution to which SCPP converges. The soft-margin formulation accommodates noise near otherwise large margin separators by penalising points which lie within their margins rather than constraining the problem to exclude such solutions. Wang et al. (2010) do not provide practical recommendations on how to tune the parameter which controls the penalisation of such points. We used the default values provided by the authors[2]. For iSVR we use the parameter settings suggested by Zhang et al. (2009).

6.3 Results

6.3.1 Spectral Clustering

We compare the different methods based on two popular evaluation metrics for clustering, namely Purity (Zhao and Karypis, 2004), and Normalised Mutual Information (NMI) (Strehl and Ghosh, 2002). These metrics compare the cluster assignments with the true labels of the data. Both take values in $[0, 1]$, with larger values indicating better performance. The following benchmark datasets were used for comparison. Optical recognition of handwritten digits (Opt. Digits)[3], Pen based recognition of handwritten digits (Pen Digits)[2], Multiple feature digits (M.F. Digits)[2], Satellite[2], Statlog image segmentation (Image Seg.)[2], Breast cancer Wisconsin (Br. Cancer)[2], Synthetic control chart (Chart)[2], Isolet[2], Dermatology[2], Yeast cell cycle analysis (Yeast)[4], Smartphone based activity recognition (Smartphone)[2], Yale faces dataset B 30×40 (Faces)[5], Phoneme[6]. Before applying the clustering algorithms, data were scaled so that every column had unit variance.

Clustering results for all methods considered are given in Table 1. SCPP achieves the highest performance in more than half the cases considered. Furthermore, in every case SCPP is competitive with the method which obtained the highest performance on the corresponding dataset. All other methods achieve substantially lower performance than SCPP in multiple examples.

The vastly different natures of the datasets considered means that the associated clustering tasks differ in difficulty. This is evidenced by the range of performance values achieved by the clustering algorithms on different datasets. To combine the results from the different datasets we standardise them as follows. For each dataset $\mathcal{X}$ we compute for each method the relative deviation from the average performance of all methods when applied to $\mathcal{X}$. That is, for each method, M_i, we compute the relative purity,

$$\frac{\mathrm{Purity}(M_i, \mathcal{X}) - \frac{1}{\#\mathrm{Methods}} \sum_{j=1}^{\#\mathrm{Methods}} \mathrm{Purity}(M_j, \mathcal{X})}{\frac{1}{\#\mathrm{Methods}} \sum_{j=1}^{\#\mathrm{Methods}} \mathrm{Purity}(M_j, \mathcal{X})}, \tag{18}$$

and similarly for NMI. We can then compare the distributions of the relative performance measures from all datasets and for all methods. It is clear from Table 1 that the DRSC method is not competitive with other methods in the examples considered, due to its substantially inferior performance on multiple datasets. Moreover, the performance of DRSC is sufficiently low to obscure the comparisons between other methods. We therefore remove DRSC from this comparison and in computing the relative performance measures. Figure 4 shows boxplots of the relative performance measures. These plots show clearly that SCPP achieves substantially higher performance overall than all other methods considered.

Among competing spectral clustering variants, we see that while both principal and independent component projections are capable of improving the performance of spectral clustering, across multiple datasets the overall performance is not appreciably higher. In fact, over multiple datasets and considering both evaluation metrics none of the four methods besides SCPP sets itself apart from the others.

The proposed SCPP method is capable of achieving a substantial improvement over alternative spectral clustering and combined spectral clustering and dimension reduction methods. The comparison between SCPP and $\mathrm{SC}_{\mathrm{PC}}^{\mathrm{rec}}$ also suggests that a substantial component of the improved performance of SCPP is contributed by the proposed projection pursuit method, as opposed to incidental differences in implementation.

[2] we used the implementation provided by the authors, taken from `https://sites.google.com/site/binzhao02/`

[3] `https://archive.ics.uci.edu/ml/datasets.html`

[4] `http://genome-www.stanford.edu/cellcycle/`

[5] `https://cervisia.org/machine_learning_data.php/`

[6] `https://web.stanford.edu/~hastie/ElemStatLearn/`

		SCPP	DRSC	SC_{PC}	SC_{IC}	SC	SC_{PC}^{rec}
Opt. Digits	Purity	**0.82**	0.10	0.66	0.69	0.66	0.74
(N = 5620, d = 64, K = 10)	NMI	**0.80**	0.03	0.63	0.67	0.63	0.68
Pen Digits	Purity	0.84	0.44	0.77	0.77	**0.87**	0.72
(N = 10992, d = 16, K = 10)	NMI	0.82	0.41	0.76	0.75	**0.82**	0.68
M.F. Digits	Purity	**0.80**	0.66	0.75	0.72	0.77	0.75
(N = 2000, d = 216, K = 10)	NMI	**0.77**	0.67	0.70	0.68	0.72	0.66
Satellite	Purity	**0.80**	0.53	0.73	0.74	0.76	**0.80**
(N = 6435, d = 36, K = 6)	NMI	**0.66**	0.22	0.61	0.62	0.62	**0.66**
Image Seg.	Purity	0.62	0.38	0.56	**0.76**	0.50	0.58
(N = 2310, d = 19, K = 7)	NMI	0.64	0.40	0.55	**0.69**	0.48	0.56
Br. Cancer	Purity	**0.97**	0.89	**0.97**	**0.97**	0.96	**0.97**
(N = 699, d = 9, K = 2)	NMI	0.78	0.51	0.81	**0.82**	0.76	0.79
Chart	Purity	**0.88**	0.24	0.67	0.73	0.67	0.81
(N = 600, d = 60, K = 6)	NMI	**0.87**	0.01	0.81	0.76	0.74	0.86
Isolet	Purity	**0.60**	-	0.59	**0.60**	**0.60**	**0.60**
(N = 6238, d = 617, K = 26)	NMI	**0.74**	-	0.69	0.67	0.69	0.70
Dermatology	Purity	0.87	0.59	0.92	0.91	**0.95**	0.86
(N = 366, d = 34, K = 6)	NMI	0.88	0.40	0.87	0.83	**0.91**	0.87
Yeast	Purity	0.74	0.42	0.68	0.60	**0.78**	0.74
(N = 698, d = 72, K = 5)	NMI	0.55	0.05	0.51	0.34	**0.57**	0.56
Smartphone	Purity	**0.71**	-	0.61	0.70	0.67	0.66
(N = 10929, d = 561, K = 12)	NMI	**0.61**	-	0.52	0.58	0.55	0.56
Faces	Purity	0.66	-	0.68	0.69	**0.73**	0.60
(N = 5850, d = 1200, K = 10)	NMI	0.70	-	0.77	**0.82**	0.76	0.62
Phoneme	Purity	**0.86**	0.56	0.83	0.84	0.80	0.82
(N = 4509, d = 256, K = 5)	NMI	0.82	0.45	**0.84**	0.76	0.71	0.70

'-' indicates that a clustering solution could not be obtained in a reasonable amount of time.

Table 1 Clustering performance. Highest performance in each case is highlighted in bold. Details of datasets in terms of number of data (N), number of dimensions (d), and number of clusters (K) are provided.

6.3.2 Large Margin Clustering

Here we present results from applying the proposed approach for a decreasing sequence of scaling parameters to obtain large margin cluster separators. We compare with two popular dedicated maximum margin clustering algorithms, namely iSVR Zhang et al. (2009) and CPMMC Wang et al. (2010)[2].

Following both Zhang et al. (2009) and Wang et al. (2010) we consider the two-cluster problems from all 45 pairs of the digits 0–9, where we use all three digits datasets considered above, namely Opt. Digits; Pen Digits and M.F. Digits. We use only the test set from the Opt. Digits data, again following Zhang et al. (2009) and Wang et al. (2010), and only the last 2500 data from the Pen Digits dataset. The results of these experiments are summarised in Table 2, which shows the average Purity and NMI from all 45 pairs for each of the three datasets. The two-dimensional SCPP obtained the highest performance overall, however the performance of iSVR and SCPP are similar in most cases. The CPMMC algorithm did not converge on 17 digit pairs from the Opt. Digits dataset, nor on any of the 45 pairs from the Pen Digits dataset. The performance of CPMMC on the M.F Digits dataset was substantially below that of SCPP and iSVR.

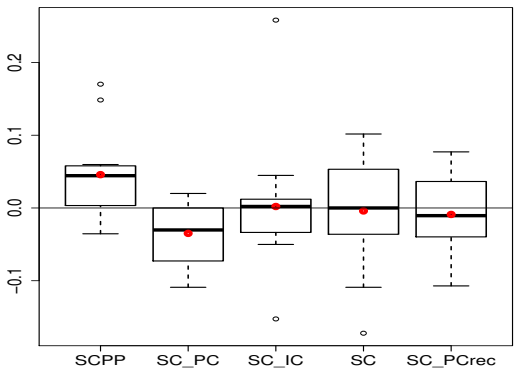
6.4 Sensitivity of SCPP

6.4.1 The Effect of σ on Performance

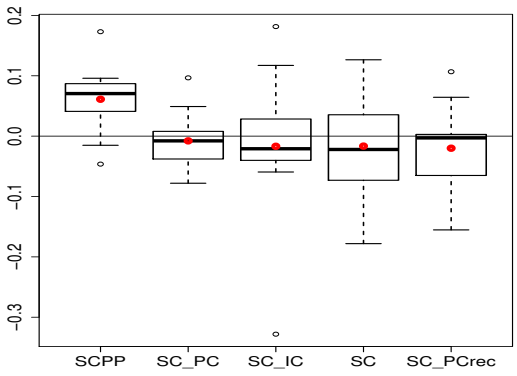
Appropriate selection of the scaling parameter is crucial for the success of spectral clustering (von Luxburg, 2007). In many cases the performance of spectral clustering can be severely affected by even slight changes in the value of this parameter. Here we present a brief sensitivity analysis of the performance of SCPP to changes in σ. We consider varying σ between $\frac{1}{2}\sigma_0$ and $2\sigma_0$, where σ_0 is the value used for the corresponding dataset in the above experiments. It is evident that no straightforward modification of the approach used to select σ would improve the general performance substantially. Indeed, in certain examples the performance is improved by increasing σ, while in others decreasing σ would have positively influenced the performance. We present the results from a subset of the datasets used previously. These are illustrated graphically in Figure 5. These examples were selected to illustrate the variety of effects that varying σ can have on the performance of SCPP.

		SCPP		iSVR		CPMMC
		dim=1	dim=2	lin. kernel	rbf kernel	lin. kernel
Opt. Digits	Avg. Purity	0.965	**0.968**	0.945	0.940	-
	Avg. NMI	0.858	**0.870**	0.846	0.789	-
Pen Digits	Avg. Purity	0.863	0.876	**0.879**	0.876	-
	Avg. NMI	0.594	**0.646**	0.641	0.622	-
M.F. Digits	Avg. Purity	0.971	**0.986**	0.979	0.971	0.857
	Avg. NMI	0.870	**0.913**	0.878	0.839	0.612

Table 2 Average clustering accuracy of large margin methods from two-way clustering solutions of all 45 pairs of digits 0–9. Highest average performance in each case is highlighted in bold, while '-' indicates that the algorithm did not converge in all cases.



(a) Relative Purity



(b) Relative NMI

Fig. 4 Box plots of relative performance measures with additional red dots to indicate means.

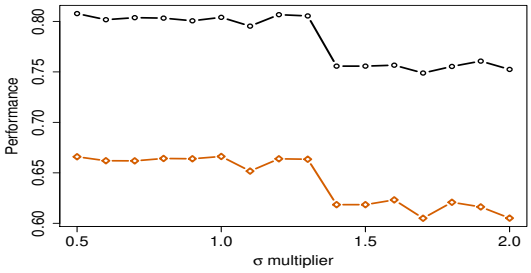
All other examples showed a relationship similar to one of these.

Figure 5(a) shows the effect of varying σ on the performance of SCPP for clustering the Satellite dataset. Here the performance is almost constant for values up to a threshold at $\sigma \approx 1.3\sigma_0$. On closer inspection it is evident that the clustering solution itself scarcely varies for $\sigma < 1.3\sigma_0$, but for higher values the increased scaling parameter has the effect of smoothing over the separation of one of the clusters which is less distinguishable than the others. As σ increases still further this solution of lower quality then remains almost unchanged up to $\sigma = 2\sigma_0$. On the other hand the clustering solution of the Image Segmentation dataset, Figure 5(b), becomes unstable for very large and very small scaling parameters. It is however extremely stable over a fairly wide range surrounding the value σ_0. Finally, in the case of the Multiple Feature Digits dataset there is a general increasing trend in the clustering accuracy, as σ increases (Figure 5(c)). The clustering solution is, however, not as consistent as in the case of the Satellite dataset.

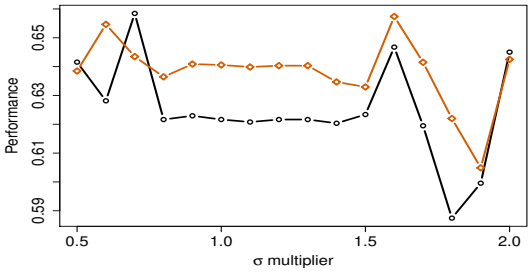
Given that the method used to compute σ_0 is a simple data driven rule, it is encouraging to see that SCPP is, in general, fairly robust to changes in this important parameter. We believe this is as a result of the projection pursuit being capable of obtaining a projection of the data for which the chosen value of the scaling parameter appropriately captures the clusters present within that projection. This option is not available to methods which do not utilise the spectral clustering objective in their dimension reduction objective. Furthermore, SCPP is protected from very small scaling parameters leading to a focus on outliers by the modified metric used to compute similarities.

6.4.2 The Effect of Microclusters on Performance

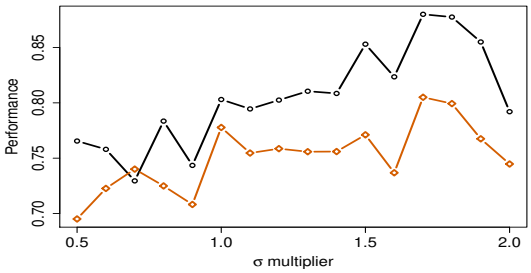
To investigate the effect of microclusters on clustering accuracy we simulated datasets from Gaussian mixtures containing 5 components (clusters) in 50 dimensions. This allows us to generate datasets of any desired size. For these experiments 30 sets of parameters for the Gaussian mixtures were generated randomly. In the first case a single dataset of size 1000 was simulated from each set of parameters, and clustering solutions obtained for a number of microclusters, m, ranging from 100 to 1000, the final value therefore applying



(a) Satellite



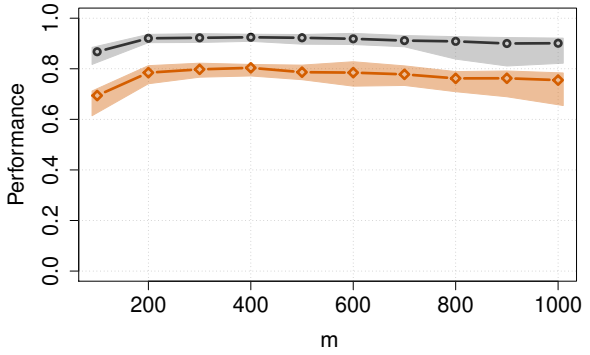
(b) Image Segmentation



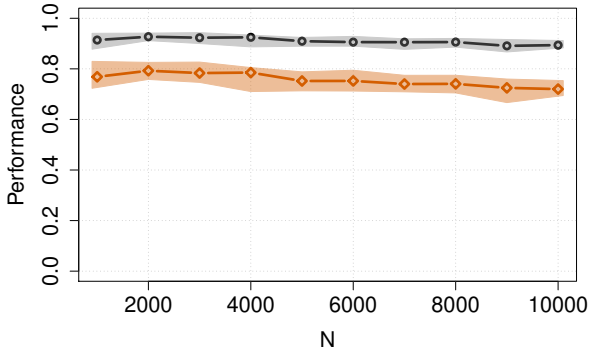
(c) M.F. Digits

Purity (–○–), NMI (–◇–)

Fig. 5 Effect of σ on performance. Plots show performance measures for values of σ between 0.5 and 2 times the value used for experiments.



(a) Fixed number of data (1000) and varying number of microclusters, m



(b) Fixed number of microclusters (200) and varying number of data, N

Purity (–○–), NMI (–◇–)

Fig. 6 Effect of microclusters on performance. Plots show median and interquartile ranges of performance measures from 30 datasets simulated from 50 dimensional Gaussian mixtures with 5 clusters.

no approximation. Figure 6(a) shows the median and interquartile range of both performance measures for 10 values of m. It is evident that aside from $m=100$, performance is similar for all other values, and so using a small value, say $m=200$, should be sufficient to obtain a good approximation of the underlying optimisation surface.

In the second case, we fix the number of microclusters, $m=200$, and for each set of parameters simulate datasets with between 1000 and 10 000 observations. In the most extreme case, therefore, the number of microclusters is only 2% of the total number of data. Figure 6(b) shows the corresponding performance plots, again containing the medians and interquartile ranges. Even for datasets of size 10 000, the coarse approximation of the dataset through 200 microclusters is sufficient to obtain a high quality projection using the proposed approach.

7 Conclusions

We proposed an approach to identify optimal projections to bi-partition a dataset through spectral clustering, based on the minimisation of the second smallest

eigenvalue of the graph Laplacian (which measures the connectivity of the two clusters) with respect to the projection. We provided a rigorous analysis of this optimisation problem and proposed a globally convergent algorithm, which directly minimises the overall objective. Using this approach to perform binary partitioning recursively gives rise to a divisive clustering algorithm capable of identifying clusters defined in different subspaces.

The computational cost of the proposed projection pursuit method per iteration is $\mathcal{O}(N^2)$, where N is the number of observations, which can become prohibitive for large datasets. To mitigate this an approximation method using microclusters, with provable error bounds is proposed. This reduces the complexity to $\mathcal{O}(m^2)$, where m is the number of microclusters. We found that in practice using even a small number of microclusters, $m=200$, our method is capable of generating high quality clustering models. This results in a speed up of up to two orders of magnitude for the examples considered in this paper.

Finally, we established an asymptotic connection between optimal univariate projections for spectral bi-partitioning and maximum margin hyperplanes. In particular we showed that as the scaling parameter of the similarity function is reduced towards zero, the optimal vector to bi-partition the data using spectral clustering also achieves the maximum Euclidean distance between the two clusters. In other words, the optimal projection vector for spectral bi-partitioning converges to the normal vector to the maximum margin separating hyperplane.

Experimental results on a large collection of datasets indicate that the proposed approach is highly competitive with spectral clustering applied on the full dimensional data, and with existing dimension reduction methods for spectral clustering.

It is interesting to note that while we discuss only the linear projection of Euclidean embedded data, the methodology we present can be generalised to apply to any differentiable transformation of a collection of data objects admitting a similarity measure. Extensions to structured data such as time series, graphical and image data represent interesting future directions for this work.

Acknowledgements: The authors would like to thank the anonymous reviewers for their insightful recommendations, which helped improve the quality of the paper. They would also like to thank Dr. Teemu Roos for his valuable comments on this work. Finally, they are very grateful to Dr. Kai Zhang for providing code to implement the iSVR algorithm.

A Avoiding Outliers

It has been documented that spectral clustering can be sensitive to outliers (Rahimi and Recht, 2004). Our experience has shown that this problem becomes more pronounced when performing dimension reduction based on the spectral clustering objective, especially in high dimensional applications. Consider the extreme case where $d>N$: since the linear system $V^\top X=P$ is underdetermined, for any P there exists $\boldsymbol{\theta}\in\Theta, c\in\mathbb{R}\backslash\{0\}$ s.t. $V(\boldsymbol{\theta})^\top X=cP$. The projected data can therefore be made to have *any* distribution (up to a scaling constant). In other words there will always be projections that contain outliers. We have found that even in problems of moderate dimensionality, there often exist projections which induce large separation of a small group of points from the remainder of the data. These projections frequently achieve the minimum spectral connectivity for both Ratio Cut and Normalised Cut.

We have found that by defining a metric which encourages the induced cluster boundaries to intersect a compact set, $\boldsymbol{\Delta}(\boldsymbol{\theta})$, around the mean of the projected data, the problem of outliers can be mitigated. This is achieved by reducing the distance, relative to the usual Euclidean metric, to points lying outside $\boldsymbol{\Delta}(\boldsymbol{\theta})$. Points lying outside $\boldsymbol{\Delta}(\boldsymbol{\theta})$, which may be outliers, therefore have increased similarity to all others. We define $\boldsymbol{\Delta}(\boldsymbol{\theta})=\Delta_1\times\ldots\times\Delta_l$, where $\Delta_i=[\mu_i-\beta\sigma_i,\mu_i+\beta\sigma_i]$; μ_i and σ_i are the mean and standard deviation of the i-th component of the projected data; and $\beta\geqslant 0$ controls the size of $\boldsymbol{\Delta}(\boldsymbol{\theta})$. The modified distance metric, $d(\cdot,\cdot)$, is defined with respect to a continuously differentiable transformation, T_Δ, of the projected data,

$$d(p_i,p_j)=\|T_\Delta(p_i)-T_\Delta(p_j)\|_2, \tag{19}$$

$$T_\Delta(y)=(t_{\Delta_1}(y_1),\ldots,t_{\Delta_l}(y_l)), \tag{20}$$

$$t_{\Delta_i}(z):=\begin{cases} c_2-\beta\sigma_i-\delta(c_1-\beta\sigma_i-z)^{1-\delta}, & z<-\beta\sigma_i \\ z, & z\in\Delta_i \\ \beta\sigma_i+\delta(z-\beta\sigma_i+c_1)^{1-\delta}-c_2, & z>\beta\sigma_i, \end{cases} \tag{21}$$

where $\delta\in(0,0.5]$ is the distance reducing parameter, and c_1 and c_2 are equal to $(\delta(1-\delta))^{1/\delta}$ and $\delta c_1^{1-\delta}$ respectively. By construction $\|T_\Delta(p_i)-T_\Delta(p_j)\|_2\leq\|p_i-p_j\|_2$ for any $p_i,p_j\in\mathbb{R}^l$, with strict inequality when either or both $p_i,p_j\notin\boldsymbol{\Delta}(\boldsymbol{\theta})$.

Figure 7 illustrates the impact of T_Δ on pairwise distances in the univariate case. As shown, distance increases linearly in the interval Δ, but outside Δ it increases much more slowly, with the rate being determined by δ. In the limit as δ approaches zero, all points outside Δ are mapped to the boundary of Δ. As a result distances between points outside Δ and all other points are much smaller after being transformed through T_Δ, and points which can be characterised as outliers in terms of the original projections, $\mathcal{P}$, do not appear as such in terms of $T_\Delta(\mathcal{P})$.

An illustration of the usefulness of this modified metric is provided in Figure 8. The figure shows two dimensional projections of the 64 dimensional optical recognition of handwritten digits dataset (Bache and Lichman, 2013). The left plots show the true clusters while the right plots show the clustering assignments based on spectral clustering using the normalised Laplacian (Shi and Malik, 2000). Figure 8(a) shows the projection onto the first two principal components, which are also used as initialisation for our method. There are clearly a few points outlying from the remainder of the data, which are separated by the spectral clustering algorithm. Figure 8(b) shows the optimal projection from minimising $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ using the Euclidean metric. The result is that the outlying points have been further separated from the remainder of the

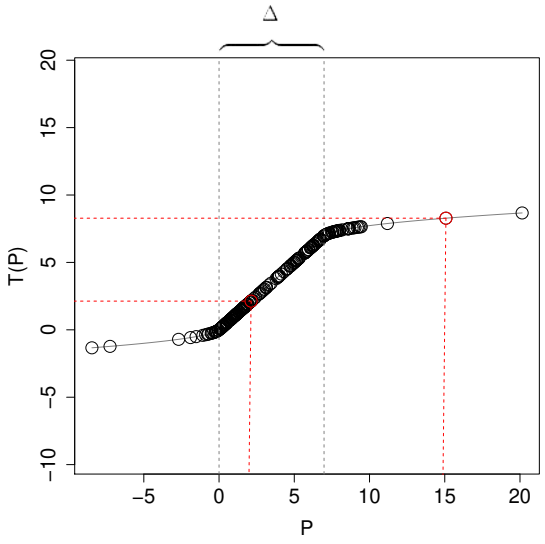


Fig. 7 Pairwise distances of points outside Δ are decreased through the transformation T_Δ

data, thereby exacerbating the outlier problem. Finally, Figure 8(c) shows the same result but using the modified metric discussed above, and with β=3. In this case the projection pursuit is able to find a projection which separates two of the true clusters clearly from the remainder.

B Derivatives

B.1 Evaluating $D_{P_i}\lambda_2(\cdot)$

We first consider the standard Laplacian L, and use λ and u to denote the second eigenvalue and corresponding eigenvector. By Eq. (11) we have $d\lambda{=}u^\top d(L)u{=}u^\top d(D)u{-}u^\top d(A)u$. Now,

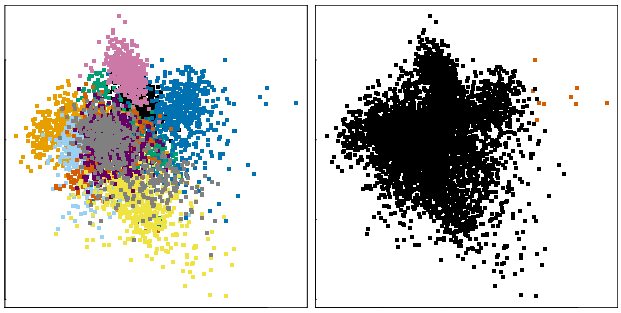
$$\frac{\partial D_{ii}}{\partial P_{mn}}{=}\sum_{j=1}^{N}\frac{\partial A_{ij}}{\partial P_{mn}}{=}\sum_{j=1}^{N}\frac{\partial s(P,i,j)}{\partial P_{mn}},$$
$$\frac{\partial A_{ij}}{\partial P_{mn}}{=}\frac{\partial s(P,i,j)}{\partial P_{mn}},$$

and so,

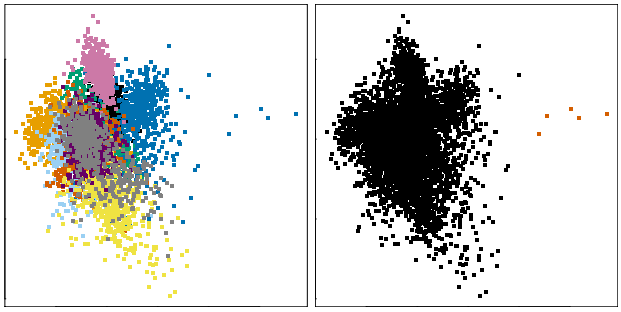
$$\frac{\partial\lambda}{\partial P_{mn}}{=}u^\top\frac{\partial L}{\partial P_{mn}}u{=}\frac{1}{2}\sum_{i,j}(u_i{-}u_j)^2\frac{\partial s(P,i,j)}{\partial P_{mn}}.$$

For the normalised Laplacian, L_{N}, consider first

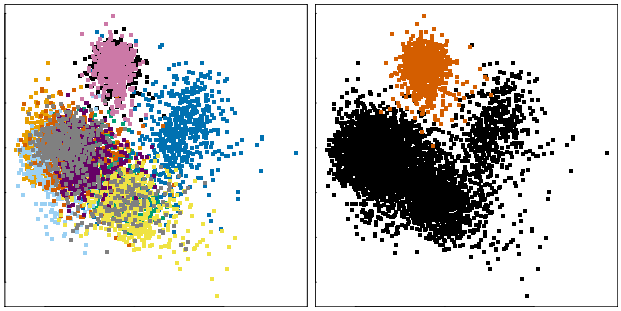
$$\begin{aligned}d(L_{\mathrm{N}}){=}&d(D^{-1/2}LD^{-1/2})\\ {=}&d(D^{-1/2})LD^{-1/2}{+}D^{-1/2}d(D)D^{-1/2}\\ &{-}D^{-1/2}d(A)D^{-1/2}{+}D^{-1/2}Ld(D^{-1/2}).\end{aligned}$$



(a) PCA projection used for initialisation



(b) Optimal projection from minimising $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ with the Euclidean metric



(c) Optimal projection from minimising $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ with the modified metric (β=3)

Fig. 8 Two dimensional projections of optical recognition of handwritten digits dataset. The left plots show the true clusters while the right plots show the partitions made by spectral clustering.

We again use λ and u to denote the second eigenvalue and corresponding eigenvector. Using $LD^{-1/2}u{=}\lambda D^{1/2}u$,

$$\begin{aligned}d\lambda{=}&u^\top d(D^{-1/2})LD^{-1/2}u{+}u^\top D^{-1/2}d(D)D^{-1/2}u\\ &{-}u^\top D^{-1/2}d(A)D^{-1/2}u{+}u^\top D^{-1/2}Ld(D^{-1/2})u\\ {=}&\lambda u^\top d(D^{-1/2})D^{1/2}u{+}u^\top D^{-1/2}d(D)D^{-1/2}u\\ &{-}u^\top D^{-1/2}d(A)D^{-1/2}u{+}\lambda u^\top D^{1/2}d(D^{-1/2})u\\ {=}&(1{-}\lambda)u^\top D^{-1/2}d(D)D^{-1/2}u{-}u^\top D^{-1/2}d(A)D^{-1/2}u.\\ {=}&u^\top D^{-1/2}d(L)D^{-1/2}u{-}\lambda u^\top D^{-1/2}d(D)D^{-1/2}u.\end{aligned}$$

Where in the third step we made use of the fact that $d(D^{-1/2})DD^{-1/2}+D^{-1/2}d(D)D^{-1/2}+D^{-1/2}Dd(D^{-1/2})=d(D^{-1/2}DD^{-1/2})=d(I)=\mathbf{0}$. Therefore,

$$\frac{\partial\lambda}{\partial P_{mn}}=\frac{1}{2}\sum_{i,j}\left(\frac{u_i}{\sqrt{d_i}}-\frac{u_j}{\sqrt{d_j}}\right)^2\frac{\partial s(P,i,j)}{\partial P_{mn}}-\lambda\sum_{i,j}\frac{u_i^2}{d_i}\frac{\partial s(P,i,j)}{\partial P_{mn}}.$$

B.2 Derivatives of the Approximate Eigenvalue Functions based on Microclusters

In the general case we may consider a set of m microclusters with centers $c_1,...,c_m$ and counts $n_1,...,n_m$. The derivations we provide are valid for $n_i=1\ \forall i\in\{1,...,m\}$, and so apply to the exact formulation of the problem as well. Let $\boldsymbol{\theta}\in\Theta$. We find it practically convenient to associate the transformation in Eq. (20), which incorporates the set $\boldsymbol{\Delta}(\boldsymbol{\theta})$, with the projection of the microclusters rather than with the computation of similarities. Specifically, we now let $\mathcal{T}$ be the transformed projected microcluster centers, i.e.,

$$\begin{aligned}\mathcal{T}&=\{t_1,t_1,...,t_m,t_m\}\\&=\{T_{\boldsymbol{\Delta}(\boldsymbol{\theta})}(V(\boldsymbol{\theta})^\top c_1),T_{\boldsymbol{\Delta}(\boldsymbol{\theta})}(V(\boldsymbol{\theta})^\top c_1),\\&\quad ...,T_{\boldsymbol{\Delta}(\boldsymbol{\theta})}(V(\boldsymbol{\theta})^\top c_m),T_{\boldsymbol{\Delta}(\boldsymbol{\theta})}(V(\boldsymbol{\theta})^\top c_m)\},\end{aligned}$$

where each t_i is repeated n_i times. The reason for this is that with this formulation the majority of terms in the above sums corresponding to $\partial\lambda$ (which are now partial derivatives w.r.t. the elements of $\mathcal{T}$, and not $\mathcal{P}$ as before) are zero. Specifically, with this expression for $\mathcal{T}$, and letting T be the matrix with columns corresponding to elements in $\mathcal{T}$, we have

$$\begin{aligned}\frac{\partial\lambda}{\partial T_{mn}}&=\frac{1}{2}\sum_{i,j}(u_i-u_j)^2\frac{\partial k(\|t_i-t_j\|/\sigma)}{\partial T_{mn}}\\&=\sum_{i\neq n}(u_i-u_n)^2\frac{\partial k(\|t_i-t_n\|/\sigma)}{\partial T_{mn}},\end{aligned}\tag{22}$$

and similarly for the normalised Laplacian.

In Section 3 we expressed $D_{\boldsymbol{\theta}}\lambda$ via the chain rule decomposition $D_P\lambda D_v P D_{\boldsymbol{\theta}}v$, which we can now simply restructure as $D_T\lambda D_v T D_{\boldsymbol{\theta}}v$. The compression of $\mathcal{T}$ to the size m non-repeated set, $\mathcal{T}^C=\{t_1,...,t_m\}$, requires a slight restructuring, as described in Section 5. We begin with the standard Laplacian, letting T^C be the matrix corresponding to $\mathcal{T}^C$, and define $N(\boldsymbol{\theta})$ and $B(\boldsymbol{\theta})$ as in Lemma 3. That is, $N(\boldsymbol{\theta})$ is the diagonal matrix with i-th diagonal element equal to $\sum_{j=1}^m n_j k(\|t_i-t_j\|/\sigma)$ and $B(\boldsymbol{\theta})_{i,j}=\sqrt{n_i n_j}k(\|t_i-t_j\|/\sigma)$. The derivative of the second eigenvalue of the Laplacian relies on the corresponding eigenvector, u. However, this vector is not explicitly available as we only solve the $m\times m$ eigenproblem of $N(\boldsymbol{\theta})-B(\boldsymbol{\theta})$. Let u^C be the second eigenvector of $N(\boldsymbol{\theta})-B(\boldsymbol{\theta})$. As in the proof of Lemma 3 if i,j are such that the i-th element of $\mathcal{T}$ corresponds to the j-th microcluster, then $u_j^C=\sqrt{n_j}u_i$. The derivative of $\lambda_2(N(\boldsymbol{\theta})-B(\boldsymbol{\theta}))$ with respect to the i-th column of $\boldsymbol{\theta}$, and thus equivalently of the second eigenvalue of the Laplacian, is therefore the vector with j-th entry given by

$$\sum_{k\neq j}\left(\frac{u_k^C}{\sqrt{n_k}}-\frac{u_j^C}{\sqrt{n_j}}\right)^2 n_k n_j\frac{\partial k\left(\frac{\|t_k-t_j\|}{\sigma}\right)}{\partial T_{kj}^C}D_{V_i}T_i^C D_{\boldsymbol{\theta}_i}V_i,$$

where $D_{\boldsymbol{\theta}_i}V_i$ is given in Eq. (12) and $D_{V_i}T_i^C$ is expressed below. We provide expressions for the case where

$$\Delta(\boldsymbol{\theta})=\prod_{i=1}^{l}[-\beta\sigma_{\boldsymbol{\theta}_i},\beta\sigma_{\boldsymbol{\theta}_i}],$$

as in our implementation, where we have again assumed that the data have been centered, i.e., have zero mean. Then $D_{V_i}T_i^C$ is the $m\times d$ matrix with j-th row equal to,

$$\frac{\delta(1-\delta)}{(-\beta\sigma_{\boldsymbol{\theta}_i}-V_i^\top c_j+(\delta(1-\delta))^{1/\delta})^\delta}\left(\frac{\beta}{\sigma_{\boldsymbol{\theta}_i}}\Sigma V_i+c_j\right),$$

if $V_i^\top c_j<-\beta\sigma_{\boldsymbol{\theta}_i}$,

$$c_j,$$

if $-\beta\sigma_{\boldsymbol{\theta}_i}\leq V_i^\top c_j\leq\beta\sigma_{\boldsymbol{\theta}_i}$, and

$$\frac{\delta(1-\delta)}{(V_i^\top c_j-\beta\sigma_{\boldsymbol{\theta}_i}+(\delta(1-\delta))^{1/\delta})^\delta}\left(c_j-\frac{\beta}{\sigma_{\boldsymbol{\theta}_i}}\Sigma V_i\right)+2\frac{\beta}{\sigma_{\boldsymbol{\theta}_i}}\Sigma V_i,$$

if $V_i^\top c_j>\beta\sigma_{\boldsymbol{\theta}_i}$. Here Σ is the covariance matrix of the data.

For the normalised Laplacian, the reduced $m\times m$ eigenproblem has precisely the same form as the original $N\times N$ problem, with the only difference being the introduction of the factors $n_j n_k$. Specifically, with the derivation in Section 3 we can see that the corresponding derivative is as for the standard Laplacian above, except that the coefficients $(u_j^C/\sqrt{n_j}-u_k^C/\sqrt{n_k})^2 n_j n_k$ in Eq. (23) are replaced with $(u_j^C/\sqrt{d_j}-u_k^C/\sqrt{d_k})^2-\lambda((u_j^C)^2/d_j+(u_k^C)^2/d_k)$, where λ is the second eigenvalue of the normalised Laplacian, u^C is the corresponding eigenvector and d_j is the degree of the j-th element of $\mathcal{T}^C$.

C Computational Complexity

Here we give a very brief discussion of the computational complexity of the proposed method. At each iteration in the gradient descent, computing the projected data matrix, $P(\boldsymbol{\theta})$, requires $\mathcal{O}(Nld)$ operations. Computing all pairwise similarities from elements of the l-dimensional $\mathcal{P}(\boldsymbol{\theta})$ has computational complexity $\mathcal{O}(lN^2)$, and determining both Laplacian matrices, and their associated eigenvalue/vector pairs adds a further computational cost $\mathcal{O}(N^2)$. Each evaluation of the objectives $\lambda_2(L(\boldsymbol{\theta}))$ or $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}))$ therefore requires $\mathcal{O}(lN(N+d))$ operations. In order to compute the gradients of these objectives, the partial derivatives with respect to each element of the projected data matrix need to be calculated. As we discussed in relation to the derivatives above, the majority of the terms in the sums in Eqs. (13) and (14) are zero, and in fact each partial derivative can be computed in $\mathcal{O}(N)$ time, and so all such partial derivatives can be computed in $\mathcal{O}(lN^2)$ time. The matrix derivatives $D_{\boldsymbol{\theta}_i}V_i, i=1,...,l$, in (12) can each be computed with $\mathcal{O}(d(d-1))$ operations. Finally, determining the gradients with respect to each column of $\boldsymbol{\theta}$ involves computing the matrix product $D_{\boldsymbol{\theta}_i}\lambda=D_{P_i}\lambda D_{V_i}P_i D_{\boldsymbol{\theta}_i}V_i$, where $D_{P_i}\lambda\in\mathbb{R}^{1\times N}, D_{V_i}P_i\in\mathbb{R}^{N\times d}$ and $D_{\boldsymbol{\theta}_i}V_i\in\mathbb{R}^{d\times(d-1)}$. This has complexity $\mathcal{O}(Nd(d-1))$. The complete gradient calculation therefore requires $\mathcal{O}(lN(N+d(d-1)))$ operations. We have found that the optimality conditions based on directional derivatives and gradient sampling steps are seldom, if ever required, and moreover that these do not constitute the bottleneck in the running time of the method in practice. The complexity of the optimality condition check may be computed along similar lines, and be found to be $\mathcal{O}(t^2 lN(N+d(d-1)))$, where t is the multiplicity of the eigenvalue $\lambda=\lambda_2(L(\boldsymbol{\theta}))$. The

	SCPP	DRSC
Opt. Digits	450	173120⋆
Pen Digits	787	83386⋆
Breast Cancer	19	51
Chart	17	32178⋆
Dermatology	19	4843⋆
Yeast	35	48588⋆

Table 3 Running time (in seconds) of SCPP and DRSC on six datasets. ⋆ indicates that the algorithm had not converged when terminated after the amount of time given above.

gradient sampling is simply $\mathcal{O}(d)$ times the cost of computing a single gradient. The total complexity of the projection pursuit optimisation depends on the number of iterations in the gradient descent method, where in general this number is bounded for a given accuracy level. For our experiments we use the BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm as this has been found to perform well on non-smooth functions (Lewis and Overton, 2013).

Table 3 shows the observed running times for SCPP and DRSC when applied to six datasets which were used in the experiments. To render the comparison relevant, we did not use the microcluster approach to speed up the SCPP algorithm. We considered only subsets of the Opt. Digits and Pen Digits datasets so that run times for DRSC could be obtained in a reasonable amount of time. We used the same subsets as in the experiments for maximum margin clustering. The SCPP algorithm converged in a reasonable amount of time in all cases, despite the absence of the microcluster speed up. DRSC on the other hand took as many as three orders of magnitude longer to run on some datasets. Moreover it failed to converge in half of the cases considered.

D Proofs

D.1 Proof of Theorem 2

Before proving Theorem 2, we require some supporting theory which we present below. We will use the notation $v^\top\mathcal{X}=\{v^\top x_1,...,v^\top x_N\}$, and for a set $\mathcal{P}\subset\mathbb{R}$ and $y\in\mathbb{R}$ we write, for example, $\mathcal{P}_{>y}$ for $\mathcal{P}\cap(y,\infty)$. Recall that for scaling parameter $\sigma>0$ we define $\boldsymbol{\theta}_\sigma:=\text{argmin}_{\boldsymbol{\theta}\in\Theta}\lambda_2(L(\boldsymbol{\theta},\sigma))$, where $L(\boldsymbol{\theta},\sigma)$ is as $L(\boldsymbol{\theta})$ from before, but with an explicit dependence on the scaling parameter. That is, $\boldsymbol{\theta}_\sigma$ defines the projection generating the minimal spectral connectivity of $\mathcal{X}$ for a given value of σ. We define $\boldsymbol{\theta}_\sigma^N$ similarly for the normalised Laplacian.

Recall that we are interested in those hyperplanes which intersect an arbitrary convex set $\boldsymbol{\Delta}$. This is because very often the maximum marging hyperplane will separate only a few points from the remainder, as data tend to be more sparse in the tails of the underlying distribution. To account for the potential for hyperplanes with very large margins lying in the tails of the distribution, we make the additional assumption that the distance reducing parameter, δ, tends to zero along with σ.

Lemmas 4 and 5 provide lower bounds on the second eigenvalue of the graph Laplacians of a one dimensional data set in terms of the largest Euclidean separation of adjacent points which lie within the interval Δ, used to represent $\boldsymbol{\Delta}(\boldsymbol{\theta})$ in the context of a projection of $\mathcal{X}$. These lemmas also show how we construct the set $\boldsymbol{\Delta}'$. Lemmas 6 and 7 use these results to show that a projection angle $\boldsymbol{\theta}\in\Theta$ leads to lower spectral connectivity than all projections admitting smaller maximal margin hyperplanes intersecting $\boldsymbol{\Delta}'$ for all pairs σ,δ sufficiently close to zero.

Lemma 4 *Let $k:\mathbb{R}_+\to\mathbb{R}_+$ be a non-increasing, positive function and let $\sigma>0,\delta\in(0,0.5]$. Let $\mathcal{P}=\{p_1,...,p_N\}$ be a univariate data set and let $\Delta=[a,b]$ for $a<b\in\mathbb{R}$. Suppose that $|\mathcal{P}\cap\Delta|\geq 2$ and $a\geq\min\{\mathcal{P}\},b\leq\max\{\mathcal{P}\}$. Define $\Delta'=[a',b']$, where $a'=(a+\min\{\mathcal{P}\cap\Delta\})/2$, and $b'=(b+\max\{\mathcal{P}\cap\Delta\})/2$. Let $M=\max_{x\in\Delta'}\{\min_{i=1...N}|x-p_i|\}$. Define $L(\mathcal{P})$ to be the Laplacian of the graph with vertices $\mathcal{P}$ and similarities according to $s(P,i,j)=k(|T_\Delta(p_i)-T_\Delta(p_j)|/\sigma)$, where $P\in\mathbb{R}^{1\times N}$ is the matrix with i-th column equal to p_i. Then $\lambda_2(L(\mathcal{P}))\geq\frac{1}{|\mathcal{P}|^3}k((2M+\delta C)/\sigma)$, where $C=\max\{D,D^{1-\delta}\}$, $D=\max\{a-\min\{\mathcal{P}\},\max\{\mathcal{P}\}-b\}$.*

Proof We can assume that $\mathcal{P}$ is sorted in increasing order, i.e. $p_i\leq p_{i+1}$, since this does not affect the eigenvalues of $L(\mathcal{P})$. We first show that $s(P,i,i+1)\geq k((2M+\delta C)/\sigma)$ for all $i=1,...,N-1$. To this end observe that $\delta\Big(x+\Big(\delta(1-\delta)^{\frac{1}{\delta}}\Big)\Big)^{1-\delta}-\delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}}\leq\delta\max\{x,x^{1-\delta}\}$ for $x\geq 0$.

- If $p_i,p_{i+1}\leq a$ then $s(P,i,i+1)=k((T_\Delta(p_{i+1})-T_\Delta(p_i))/\sigma)\geq k((T_\Delta(a)-T_\Delta(p_i))/\sigma)\geq k((2M+\delta C)/\sigma)$ by the definition of C and using the above inequality, since k is non-increasing. The case $p_i,p_{i+1}\geq b$ is similar.
- If $p_i,p_{i+1}\in\Delta$ then $p_i,p_{i+1}\in\Delta'\Rightarrow|p_i-p_{i+1}|\leq 2M\Rightarrow s(P,i,i+1)\geq k(2M/\sigma)\geq k((2M+\delta C)/\sigma)$ since M is the largest margin in Δ'.
- If none of the above hold, then we lose no generality in assuming $p_i<a$, $a<p_{i+1}<b$ since the case $a<p_i<b$, $p_{i+1}>b$ is analogous. We must have $p_{i+1}=\min\{\mathcal{P}\cap\Delta\}$ and so $a'=(a+p_{i+1})/2$. If $p_{i+1}-a>2M$ then $\min_{j=1...N}|a'-p_j|>M$, a contradiction since $a'\in\Delta'$ and M is the largest margin in Δ'. Therefore $p_{i+1}-a\leq 2M$. In all

$$\begin{aligned}T_\Delta(p_{i+1})-T_\Delta(p_i)&=(p_{i+1}-a)+\delta(a-p_i+(\delta(1-\delta))^{\frac{1}{\delta}})^{1-\delta}\\&\quad-\delta(\delta(1-\delta))^{\frac{1-\delta}{\delta}}\\&\leq 2M+\delta C\\&\Rightarrow s(P,i,i+1)\geq k((2M+\delta C)/\sigma).\end{aligned}$$

Now, let u be the second eigenvector of $L(\mathcal{P})$. Then $\|u\|=1$ and $u\perp\mathbf{1}$ and therefore $\exists i,j$ s.t. $u_i-u_j\geq\frac{1}{\sqrt{N}}$. We thus know that there exists m s.t. $|u_m-u_{m+1}|\geq\frac{1}{N^{3/2}}$. By von Luxburg (2007, Proposition 1), we know that

$$\begin{aligned}u^\top L(\mathcal{P})u&=\frac{1}{2}\sum_{i,j}s(P,i,j)(u_i-u_j)^2\\&\geq s(P,m,m+1)(u_m-u_{m+1})^2\\&\geq\frac{1}{N^3}k((2M+\delta C)/\sigma),\end{aligned}$$

since all consecutive pairs p_m, p_{m+1} have similarity at least $k((2M+\delta C)/\sigma)$, by above. Therefore $\lambda_2(L(\mathcal{P}))\geq\frac{1}{N^3}k((2M+\delta C)/\sigma)$ as required.

Lemma 5 *Let the conditions of Lemma 4 hold and let $L_N(\mathcal{P})$ be the normalised Laplacian of the graph with vertices $\mathcal{P}$ and similarities $s(P,i,j)=k(|T_\Delta(p_i)-T_\Delta(p_j)|/\sigma)$. Then*

$$\lambda_2(L_N(\mathcal{P}))\geq\frac{1}{|\mathcal{P}|^4}k((2M+\delta C)/\sigma).$$

Proof The proof is similar to that of Lemma refthm:maxdistbound, but requires a few simple modifications. Let u be the second eigenvector of $L_{\mathrm{N}}(\mathcal{P})$. Since $\|u\|=1, \exists i\in\{1,...,N\}$ s.t. $|u_i|\geq\frac{1}{\sqrt{N}}$. Suppose w/o loss of generality that $u_i\leq-\frac{1}{\sqrt{N}}$. Now consider that for all $j,k\in\{1,...,N\}$ we have $0<s(P,j,k)\leq 1$ and $s(P,j,j)=1$ and so $1<\sqrt{d_j}\leq\sqrt{N}$ for all $j\in\{1,...,N\}$. Therefore we have $u_i/\sqrt{d_i}\leq-\frac{1}{N}$. Furthermore, since $uD^{1/2}\perp\mathbf{1}$ we have $u_j>0$ for some $j\in\{1,...,N\}\Rightarrow u_j/\sqrt{d_j}>0$. Therefore, $u_j/\sqrt{d_j}-u_i/\sqrt{d_i}>\frac{1}{N}$. We thus know that $\exists m\in\{1,...,N\}$ s.t. $\left|u_m/\sqrt{d_m}-u_{m+1}/\sqrt{d_{m+1}}\right|>\frac{1}{N^2}$. By von Luxburg (2007, Proposition 3), we know that

$$\begin{aligned}u^\top L_{\mathrm{N}}(\mathcal{P})u&=\frac{1}{2}\sum_{i\neq j}s(P,i,j)(u_i/\sqrt{d_i}-u_j/\sqrt{d_j})^2\\&\geq S(P,m,m+1)(u_m/\sqrt{d_m}-u_{m+1}/\sqrt{d_{m+1}})^2\\&>\frac{1}{N^4}k((2M+\delta C)/\sigma),\end{aligned}$$

where the bound on $s(P,m,m+1)$ is taken from the proof of Lemma 5. Therefore $\lambda_2(L_{\mathrm{N}}(\mathcal{P}))\geq\frac{1}{N^4}k((2M+\delta C)/\sigma)$ as required.

In the above we have assumed that Δ is contained within the convex hull of the points $\mathcal{P}$, however the results of this section can easily be modified to allow for cases where this does not hold. In particular, if an unconstrained large margin hyperplane is sought, then setting $\boldsymbol{\Delta}$ to be arbitrarily large allows for this. We have merely stated the results in the most convenient context for our practical implementation.

The set Δ' in the above is defined in terms of the one dimensional interval $[a,b]$. We define the full dimensional set $\boldsymbol{\Delta}'$ along the same lines by,

$$\boldsymbol{\Delta}'=\{x\in\mathbb{R}^d|v(\boldsymbol{\theta})^\top x\in\Delta(\boldsymbol{\theta})'\ \forall\boldsymbol{\theta}\in\Theta\},$$

$$\Delta(\boldsymbol{\theta})':=\left[\frac{\min\Delta(\boldsymbol{\theta})+\min\{v(\boldsymbol{\theta})^\top\mathcal{X}\cap\Delta(\boldsymbol{\theta})\}}{2},\right. \tag{23}$$

$$\left.\frac{\max\Delta(\boldsymbol{\theta})+\max\{v(\boldsymbol{\theta})^\top\mathcal{X}\cap\Delta(\boldsymbol{\theta})\}}{2}\right]. \tag{24}$$

Here we assume that $\boldsymbol{\Delta}$ is contained within the convex hull of the d-dimensional data set X. Notice that since $\boldsymbol{\Delta}$ is convex, we have $v(\boldsymbol{\theta})^\top\boldsymbol{\Delta}'=\Delta(\boldsymbol{\theta})'$. In what follows we show that as σ is reduced to zero the optimal projection for spectral partitioning converges to the projection admitting the largest margin hyperplane intersecting $\boldsymbol{\Delta}'$. If it is the case that the largest margin hyperplane intersecting $\boldsymbol{\Delta}$ also intersects $\boldsymbol{\Delta}'$, as is often the case, although this fact will not be known, then it is actually not necessary that δ tend towards zero. In such cases it only needs to satisfy $\delta\leq 2M/C$ for the corresponding values of M and C over all possible projections. In particular, choosing $\max\{\mathrm{Diam}(\mathcal{X}),\mathrm{Diam}(\mathcal{X})^{1-\delta}\}$ instead of C is appropriate for all projections.

Lemma 6 *Let $\boldsymbol{\theta}\in\Theta$ and let $k:\mathbb{R}_+\rightarrow\mathbb{R}_+$ be non-increasing, positive, and satisfy*

$$\lim_{x\rightarrow\infty}k(x(1+\epsilon))/k(x)=0$$

for all $\epsilon>0$. Then for any $0<m<\max_{b\in\Delta(\boldsymbol{\theta})'}margin(v(\boldsymbol{\theta}),b)$ there exists $\sigma'>0$ s.t. if $0<\sigma<\sigma'$ and

$$\max_{c\in\Delta(\boldsymbol{\theta}')'}margin(v(\boldsymbol{\theta}'),c)<\max_{b\in\Delta(\boldsymbol{\theta})'}margin(v(\boldsymbol{\theta}),b)-m$$

then $\lambda_2(L(\boldsymbol{\theta},\sigma))<\lambda_2(L(\boldsymbol{\theta}',\sigma))$.

Proof Let $B=\mathrm{argmax}_{b\in\Delta(\boldsymbol{\theta})'}\mathrm{margin}(v(\boldsymbol{\theta}),b)$ and let M be the corresponding margin, i.e., $M=\mathrm{margin}(v(\boldsymbol{\theta}),B)$. We assume that $M\neq 0$, since otherwise there is nothing to show. Now, since spectral clustering solves a relaxation of the minimum normalised cut problem we have,

$$\begin{aligned}\lambda_2(L(\boldsymbol{\theta},\sigma))&\leq\frac{1}{|\mathcal{X}|}\min_{\mathcal{C}\subset\mathcal{X}}\sum_{\substack{i,j:x_i\in\mathcal{C}\\x_j\notin\mathcal{C}}}s(P(\boldsymbol{\theta}),i,j)\left(\frac{1}{|\mathcal{C}|}+\frac{1}{|\mathcal{X}\backslash\mathcal{C}|}\right)\\&\leq\frac{1}{|\mathcal{X}|}\sum_{\substack{i,j:v(\boldsymbol{\theta})^\top x_i<B\\v(\boldsymbol{\theta})^\top x_j>B}}s(P(\boldsymbol{\theta}),i,j)\left(\frac{1}{|(v(\boldsymbol{\theta})^\top\mathcal{X})_{<B}|}\right.\\&\qquad\left.+\frac{1}{|(v(\boldsymbol{\theta})^\top\mathcal{X})_{>B}|}\right)\\&=\frac{1}{|\mathcal{X}|}\sum_{\substack{i,j:v(\boldsymbol{\theta})^\top x_i<B\\v(\boldsymbol{\theta})^\top x_j>B}}k\left(\frac{T_{\Delta(\boldsymbol{\theta})}(v(\boldsymbol{\theta})^\top x_j)-T_{\Delta(\boldsymbol{\theta})}(v(\boldsymbol{\theta})^\top x_i)}{\sigma}\right)\\&\qquad\times\left(\frac{|\mathcal{X}|}{|(v(\boldsymbol{\theta})^\top\mathcal{X})_{<B}||(v(\boldsymbol{\theta})^\top\mathcal{X})_{>B}|}\right)\\&\leq\left|(v(\boldsymbol{\theta})^\top\mathcal{X})_{<B}\right|\left|(v(\boldsymbol{\theta})^\top\mathcal{X})_{>B}\right|k\left(\frac{2M}{\sigma}\right)\\&\qquad\times\left(\frac{1}{|(v(\boldsymbol{\theta})^\top\mathcal{X})_{<B}||(v(\boldsymbol{\theta})^\top\mathcal{X})_{>B}|}\right)\\&=k(2M/\sigma).\end{aligned}$$

The final inequality holds since for any i,j s.t. $v(\boldsymbol{\theta})^\top x_i<B$ and $v(\boldsymbol{\theta})^\top x_j>B$ we must have $T_{\Delta(\boldsymbol{\theta})}(v(\boldsymbol{\theta})^\top x_j)-T_{\Delta(\boldsymbol{\theta})}(v(\boldsymbol{\theta})^\top x_i)\geq 2M$. Now, for any $\boldsymbol{\theta}'\in\Theta$, let $M_{\boldsymbol{\theta}'}=\max_{c\in\Delta(\boldsymbol{\theta}')'}\mathrm{margin}(v(\boldsymbol{\theta}'),c)$. By Lemma 4 we know that $\lambda_2(L(\boldsymbol{\theta}',\sigma))\geq\frac{1}{|\mathcal{X}|^3}k((2M_{\boldsymbol{\theta}'}+\delta C/\sigma)$, where $C=\max\{\mathrm{Diam}(X),\ \mathrm{Diam}(X)^{1-\delta}\}$. Therefore,

$$\begin{aligned}&\lim_{\sigma\rightarrow 0^+}\frac{\lambda_2(L(\boldsymbol{\theta},\sigma))}{\inf_{\boldsymbol{\theta}'\in\Theta}\{\lambda_2(L(\boldsymbol{\theta}',\sigma))\big|M_{\boldsymbol{\theta}'}<M-m\}}\\&\quad\leq\lim_{\sigma\rightarrow 0^+}\frac{|\mathcal{X}|^3k(2M/\sigma)}{k((2(M-m)+\delta C)/\sigma)}\\&\quad=0.\end{aligned}$$

Since $\delta\rightarrow 0$ as $\sigma\rightarrow 0$, this gives the result.

Lemma 7 *Let the conditions of Lemma 6 hold. For any $0<m<\max_{b\in\Delta(\boldsymbol{\theta})'}margin(v(\boldsymbol{\theta}),b)$ there exists $\sigma'>0$ s.t. if $0<\sigma<\sigma'$ and*

$$\max_{c\in\Delta(\boldsymbol{\theta}')'}margin(v(\boldsymbol{\theta}'),c)<\max_{b\in\Delta(\boldsymbol{\theta})'}margin(v(\boldsymbol{\theta}),b)-m$$

then $\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta},\sigma))<\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta}',\sigma))$.

Proof Using a similar approach to that in the proof of Lemma 6, we can arrive at the following.

$$\begin{aligned}\lambda_2(L_{\mathrm{N}}(\boldsymbol{\theta},\sigma))&\leq\frac{\sum_{\substack{i,j:v(\boldsymbol{\theta})^\top x_i<B\\v(\boldsymbol{\theta})^\top x_j>B}}k\left(\frac{T_{\Delta(\boldsymbol{\theta})}(v(\boldsymbol{\theta})^\top x_j)-T_{\Delta(\boldsymbol{\theta})}(v(\boldsymbol{\theta})^\top x_i)}{\sigma}\right)}{\mathrm{vol}((v(\boldsymbol{\theta})^\top\mathcal{X})_{<B})\mathrm{vol}((v(\boldsymbol{\theta})^\top\mathcal{X})_{>B})}\\&\leq k\left(\frac{2M}{\sigma}\right)\frac{\left|(v(\boldsymbol{\theta})^\top\mathcal{X})_{<B}\right|\left|(v(\boldsymbol{\theta})^\top\mathcal{X})_{>B}\right|}{\mathrm{vol}((v(\boldsymbol{\theta})^\top\mathcal{X})_{<B})\mathrm{vol}((v(\boldsymbol{\theta})^\top\mathcal{X})_{>B})}\\&\leq k(2M/\sigma)\end{aligned}$$

where the final inequality comes from the fact that $1<d_i$ for all $i\in\{1,...,N\}$, and hence $\mathrm{vol}((v(\boldsymbol{\theta})^\top\mathcal{X})_{>B})\geq|(v(\boldsymbol{\theta})^\top\mathcal{X})_{>B}|$,

and similarly for $(v(\boldsymbol{\theta})^\top\mathcal{X})_{<B}$. The final step in the proof is equivalent to that of Lemma 6, except that $|\mathcal{X}|^3$ is replaced with $|\mathcal{X}|^4$.

Lemmas 6 and 7 show almost immediately that the margin admitted by the optimal projection for spectral bi-partitioning converges to the largest margin through $\boldsymbol{\Delta}'$ as σ goes to zero. Theorem 2, which we are now in a position to prove, shows the stronger result that the optimal projection itself converges to the projection admitting the largest margin.

Proof of Theorem 2: Take any $\epsilon>0$. Pavlidis et al. (2016) have shown that $\exists m_\epsilon>0$ s.t. for $w\in\mathbb{R}^d, c\in\mathbb{R}$, $\|(w,c)/\|w\|-(v(\boldsymbol{\theta}^\star),b^\star)\|>\epsilon\Rightarrow\text{margin}(w/\|w\|,c/\|w\|)<\text{margin}(v(\boldsymbol{\theta}^\star),b^\star)-m_\epsilon$. By Lemma 6 we know $\exists\sigma'>0$, $\delta'>0$ s.t. if $0<\sigma<\sigma'$ then $\exists c\in\Delta(\boldsymbol{\theta})$ s.t. $\text{margin}(v(\boldsymbol{\theta}_\sigma),c)\geq\text{margin}(v(\boldsymbol{\theta}^\star),b^\star)-m_\epsilon$, since $\boldsymbol{\theta}_\sigma$ is optimal for σ. Thus, by above, $\|(v(\boldsymbol{\theta}_\sigma),c)-(v(\boldsymbol{\theta}^\star),b^\star)\|\leq\epsilon$. But $\|(v(\boldsymbol{\theta}_\sigma),c)-(v(\boldsymbol{\theta}^\star),b^\star)\|\geq\|v(\boldsymbol{\theta}_\sigma)-v(\boldsymbol{\theta}^\star)\|$ for any $c\in\mathbb{R}$. Since $\epsilon>0$ was arbitrary, we therefore have $v(\boldsymbol{\theta}_\sigma)\to v(\boldsymbol{\theta}^\star)$ as $\sigma\to0^+$. The proof for $\boldsymbol{\theta}^N_\sigma$ is analogous. □

D.2 Proof of Lemma 3

The proof of Lemma 3 uses the following result from matrix perturbation theory.

Theorem 8 (Ye (2009)) *Let $A=[a_{ij}]$ and $\tilde{A}=[\tilde{a}_{ij}]$ be two symmetric positive semidefinite diagonally dominant matrices, and let $\lambda_1\leq\lambda_2\leq\ldots\leq\lambda_n$ and $\tilde{\lambda}_1\leq\tilde{\lambda}_2\leq\ldots\leq\tilde{\lambda}_n$ be their respective eigenvalues. If, for some $0\leq\epsilon<1$, $|a_{ij}-\tilde{a}_{ij}|\leq\epsilon|a_{ij}|$ $\forall i\neq j$, and $|v_i-\tilde{v}_i|\leq\epsilon v_i$ $\forall i$, where $v_i=a_{ii}-\sum_{j\neq i}|a_{ij}|$, and similarly for $\tilde{v}_i$, then*

$$|\lambda_i-\tilde{\lambda}_i|\leq\epsilon\lambda_i\ \forall i.$$

An inspection of the proof of Theorem 8 reveals that $\epsilon<1$ is necessary only to ensure that the signs of a_{ij} are the same as those of $\tilde{a}_{ij}$. In the case of Laplacian matrices this equivalence of signs holds by design, and so in this context the requirement that $\epsilon<1$ can be relaxed.

Now, for brevity we drop the notational dependence on $\boldsymbol{\theta}$. Let $\mathcal{P}^{c\prime}=\{V^\top c_1,V^\top c_1,\ldots,V^\top c_m,V^\top c_m\}$, where each $V^\top c_i$ is repeated n_i times, and let $P^{c\prime}$ be the corresponding matrix of repeated projected centroids. Let $L^{c\prime}$ be the Laplacian of the graph with vertices $\mathcal{P}^{c\prime}$ and edges given by $s(P^{c\prime},i,j)$. We begin by showing that $\lambda_2(L^{c\prime})=\lambda_2(N-B)$. Take $v\in\mathbb{R}^m$, then,

$$\begin{aligned}v^\top(N-B)v&=\sum_{i,j}s(P^c,i,j)(v_i^2n_j-v_iv_j\sqrt{n_in_j})\\&=\frac{1}{2}\sum_{i,j}s(P^c,i,j)(v_i^2n_j+v_j^2n_i-2v_iv_j\sqrt{n_in_j})\\&\geq0,\end{aligned}$$

and so $N-B$ is positive semi-definite. In addition, it is straightforward to verify that $(N-B)(\sqrt{n_1}\ \ldots\ \sqrt{n_K})=\mathbf{0}$, and hence 0 is the smallest eigenvalue of $N-B$ with corresponding eigenvector $(\sqrt{n_1}\ \ldots\ \sqrt{n_m})$. Now, let u be the second eigenvector of $L^{c\prime}$. Then $u_j=u_k$ for pairs of indices j,k aligned with the same $V^\top c_i$ in $P^{c\prime}$. Define $u^c\in\mathbb{R}^m$ s.t. $u^c_i=\sqrt{n_i}u_j$ where index j is aligned with $V^\top c_i$ in $P^{c\prime}_j$. Then $(u^c)^\top(\sqrt{n_1}\ \ldots\ \sqrt{n_m})=\sum_{i=1}^m u^c_i\sqrt{n_i}=\sum_{i=1}^m n_iu_{j_i}$ where index j_i is aligned with $V^\top c_i$ in $P^{c\prime}_{j_i}$ for each i. Therefore $n_iu_{j_i}=\sum_{j:P^{c\prime}=V^\top c_i}u_j$ and hence $(u^c)^\top(\sqrt{n_1}\ \ldots\ \sqrt{n_m})=\sum_{i=1}^m\sum_{j:P^{c\prime}_j=V^\top c_i}u_j=\sum_{i=1}^N u_i=0$ since $\mathbf{1}$ is the smallest eigenvector of $L^{c\prime}$ and so $u\perp\mathbf{1}$. Similarly $\|u^c\|^2=\sum_{i=1}^m n_iu_{j_i}^2=\sum_{i=1}^N u_i^2=1$. Thus $u^c\perp(\sqrt{n_1}\ \ldots\ \sqrt{n_m})$ and $\|u^c\|=1$ and so is a candidate for the second eigenvector of $N-B$. In addition it is straightforward to show that $(u^c)^\top(N-B)u^c=u\cdot L^{c\prime}u$. Now, suppose by way of contradiction that $\exists w\perp(\sqrt{n_1}\ \ldots\ \sqrt{n_m})$ with $\|w\|=1$ s.t. $w^\top(N-B)w<(u^c)^\top(N-B)u^c$. Then let $w'=(w_1/\sqrt{n_1}\ w_1/\sqrt{n_1}\ \ldots\ w_m/\sqrt{n_m})$ where each $w_i/\sqrt{n_i}$ is repeated n_i times. Then $\|w'\|=1$, $(w')^\top\mathbf{1}=w^\top(\sqrt{n_1}\ \ldots\ \sqrt{n_m})=0$ and $w^\top L^{c\prime}w<u^\top L^{c\prime}u$, a contradiction since u is the second eigenvector of $L^{c\prime}$.

Now, let i,j,q,r be such that $x_q\in C_i$ and $x_r\in C_j$. We temporarily drop the notational dependence on Δ. Then,

$$\begin{aligned}\|T(V^\top x_q)-T(V^\top x_r)\|&=\|T(V^\top x_q)-T(V^\top c_i)+T(V^\top c_i)\\&\qquad-T(V^\top c_j)+T(V^\top c_j)-T(V^\top x_r)\|\\&\leq\|T(V^\top x_q)-T(V^\top c_i)\|\\&\quad+\|T(V^\top c_i)-T(V^\top c_j)\|\\&\quad+\|T(V^\top c_j)-T(V^\top x_r)\|\\&\leq\rho_i+\rho_j+D_{ij},\end{aligned}$$

since T contracts distances and ρ_i and ρ_j are the radii of C_i and C_j. Since k is non-increasing we therefore have,

$$\begin{aligned}\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)}&\leq\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)}\\&\leq\frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}\\\Rightarrow1-\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)}&\leq1-\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)}\end{aligned}$$

and

$$\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)}-1\leq\frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}-1.$$

Therefore

$$\left|\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)}-1\right|\leq\max\left\{1-\frac{k(D_{ij}/\sigma)}{k((D_{ij}-\rho_i-\rho_j)^+/\sigma)},\frac{k(D_{ij}/\sigma)}{k((D_{ij}+\rho_i+\rho_j)/\sigma)}-1\right\}.$$

Now, we lose no generality by assume that $\mathcal{X}$ is ordered such that for each i the elements of cluster C_i are aligned with $V^\top c_i$ in $P^{c\prime}$, since this does not affect the eigenvalues of the Laplacian of $V^\top\mathcal{X}$, L. By the design of the Laplacian matrix the "v_i" of Theorem 8 are exactly zero. For off diagonal terms q,r with corresponding i,j as above, consider

$$\begin{aligned}\frac{|L_{qr}-L^{c\prime}_{qr}|}{|L_{qr}|}&=\frac{|k(D_{ij}/\sigma)-k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)|}{k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)}\\&=\left|\frac{k(D_{ij}/\sigma)}{k(\|T(V^\top x_q)-T(V^\top x_r)\|/\sigma)}-1\right|.\end{aligned}$$

Theorem 8 thus gives the result. □

References

Bach, F.R., Jordan, M.I.: Learning spectral clustering, with application to speech separation. Journal of Machine Learning Research **7**, 1963–2001 (2006)

Bache, K., Lichman, M.: UCI machine learning repository (2013). `http://archive.ics.uci.edu/ml`

Boumal, N., Mishra, B., Absil, P.A., Sepulchre, R.: Manopt, a matlab toolbox for optimization on manifolds. Journal of Machine Learning Research **15**, 1455–1459 (2014).

Burke, J.V., Lewis, A.S., Overton, M.L.: A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM Journal on Optimization **15**(3), 751–779 (2006)

Chi, Y., Song, X., Zhou, D., Hino, K., Tseng, B.L.: On evolutionary spectral clustering. ACM Transactions on Knowledge Discovery from Data **3**(4), 17:1–17:30 (2009).

Edelman, A., Arias, T., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications **20**(2), 303–353 (1998)

Fan, K.: On a theorem of weyl concerning eigenvalues of linear transformations i. Proceedings of the National Academy of Sciences of the United States of America **35**(11), 652 (1949)

Hagen, L., Kahng, A.B.: New spectral methods for ratio cut partitioning and clustering. IEEE transactions on computer-aided design of integrated circuits and systems **11**(9), 1074–1085 (1992)

Hartigan, J.A., Hartigan, P.M.: The dip test of unimodality. The Annals of Statistics **13**(1), 70–84 (1985)

Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Texts in Statistics. Springer, 2 ed. (2009)

Hofmeyr, D.: Improving spectral clustering using the asymptotic value of the normalised cut. arXiv preprint arXiv:1703.09975 (2017)

Hofmeyr, D., Pavlidis, N.: Maximum clusterability divisive clustering. In: Computational Intelligence, 2015 IEEE Symposium Series on, pp. 780–786. IEEE (2015)

Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of International Conference on Machine Learning (ICML), vol. 99, pp. 200–209. Bled, Slowenien (1999)

Kaiser, H.F.: The application of electronic computers to factor analysis. Educational and psychological measurement **20**(1), 141–151 (1960)

Krause, A., Liebscher, V.: Multimodal projection pursuit using the dip statistic. Preprint-Reihe Mathematik **13** (2005)

Lewis, A., Overton, M.: Nonsmooth optimization via quasi-Newton methods. Mathematical Programming **141**, 135–163 (2013)

Lewis, A.S., Overton, M.L.: Eigenvalue optimization. Acta numerica **5**, 149–190 (1996)

Magnus, J.R.: On differentiating eigenvalues and eigenvectors. Econometric Theory **1**(02), 179–191 (1985)

Ng, A., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Dietterich, T., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems 14, pp. 849 –856. MIT Press, Cambridge (2002)

Ning, H., Xu, W., Chi, Y., Gong, Y., Huang, T.S.: Incremental spectral clustering by efficiently updating the eigensystem. Pattern Recognition **43**(1), 113–127 (2010).

Niu, D., Dy, J.G., Jordan, M.I.: Dimensionality reduction for spectral clustering. In: International Conference on Artificial Intelligence and Statistics, pp. 552–560 (2011)

Nocedal, J., Wright, S.: Numerical optimization. Springer Science & Business Media (2006)

Overton, M.L., Womersley, R.S.: Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. Mathematical Programming **62**(1-3), 321–357 (1993)

Pavlidis, N.G., Hofmeyr, D.P., Tasoulis, S.K.: Minimum density hyperplanes. Journal of Machine Learning Research **17**(156), 1–33 (2016)

Peña, D., Prieto, F.J.: Cluster identification using projections. Journal of the American Statistical Association (2001)

Polak, E.: On the mathematical foundations of nondifferentiable optimization in engineering design. SIAM Review **29**(1), 21–89 (1987).

Rahimi, A., Recht, B.: Clustering with normalized cuts is clustering with a hyperplane. Statistical Learning in Computer Vision **56** (2004)

Schur, J.: Bemerkungen zur theorie der beschränkten bilinearformen mit unendlich vielen veränderlichen. Journal für die reine und Angewandte Mathematik **140**, 1–28 (1911)

Shi, J., Malik, J.: Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**(8), 888–905 (2000)

Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of machine learning research **3**(Dec), 583–617 (2002)

Tong, S., Koller, D.: Restricted bayes optimal classifiers. In: AAAI/IAAI, pp. 658–664 (2000)

Trillos, N.G., Slepčev, D., Von Brecht, J., Laurent, T., Bresson, X.: Consistency of cheeger and ratio graph cuts. The Journal of Machine Learning Research **17**(1), 6268–6313 (2016)

Vapnik, V.N., Kotz, S.: Estimation of dependences based on empirical data, vol. 40. Springer-verlag New York (1982)

von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4), 395–416 (2007).

Wagner, D., Wagner, F.: Between min cut and graph bisection. Springer (1993)

Wang, F., Zhao, B., Zhang, C.: Linear time maximum margin clustering. IEEE Transactions on Neural Networks **21**(2), 319–332 (2010)

Weiss, Y.: Segmentation using eigenvectors: a unifying view. In: Proceedings of the 7th IEEE International Conference on Computer Vision, vol. 2, pp. 975–982 (1999)

Weyl, H.: Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung). Mathematische Annalen **71**(4), 441–479 (1912)

Wolfe, P.: On the convergence of gradient methods under constraint. IBM Journal of Research and Development **16**(4), 407–411 (1972)

Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In: Advances in neural information processing systems, pp. 1537–1544 (2004)

Yan, D., Huang, L., Jordan, M.I.: Fast approximate spectral clustering. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 907–916. ACM (2009)

Ye, Q.: Relative perturbation bounds for eigenvalues of symmetric positive definite diagonally dominant matrices. SIAM Journal on Matrix Analysis and Applications **31**(1), 11–17 (2009)

Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in neural information processing systems, pp. 1601–1608 (2004)

Zhang, B. Dependence of clustering algorithm performance on clustered-ness of data. Tech. rep., Technical Report, 20010417. Hewlett-Packard Labs (2001)

Zhang, K., Tsang, I.W., Kwok, J.T.: Maximum margin clustering made practical. Neural Networks, IEEE Transactions on **20**(4), 583–596 (2009)

Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: ACM SIGMOD Record, vol. 25, pp. 103–114. ACM (1996)

Zhao, Y., Karypis, G.: Empirical and theoretical comparisons of selected criterion functions for document clustering. Machine Learning **55**(3), 311–331 (2004)