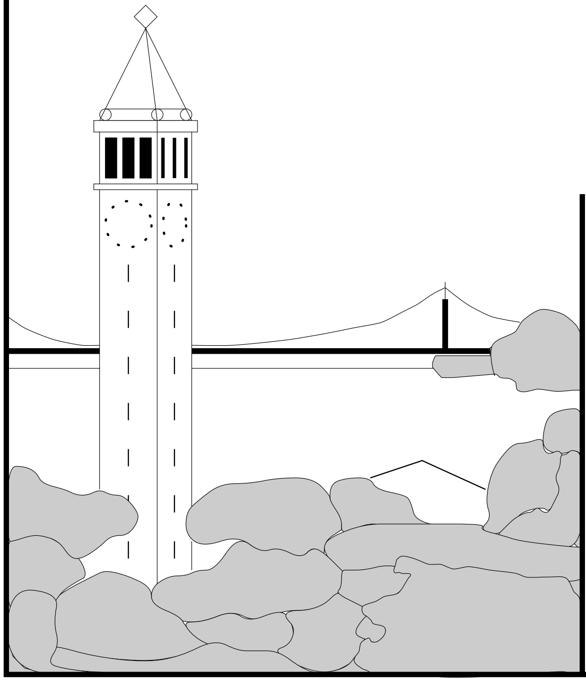
Learning Spectral Clustering

Francis R. Bach
fbach@cs.berkeley.edu
Computer Science Division
University of California
Berkeley, CA 94720, USA

Michael I. Jordan
jordan@cs.berkeley.edu
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA



Report No. UCB/CSD-03-1249

June 2003

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Learning Spectral Clustering

Francis R. Bach
fbach@cs.berkeley.edu
Computer Science Division
University of California
Berkeley, CA 94720, USA

Michael I. Jordan
jordan@cs.berkeley.edu
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA

June 2003

Abstract

Spectral clustering refers to a class of techniques which rely on the eigenstructure of a similarity matrix to partition points into disjoint clusters, with points in the same cluster having high similarity and points in different clusters having low similarity. In this paper, we derive a new cost function for spectral clustering based on a measure of error between a given partition and a solution of the spectral relaxation of a minimum normalized cut problem. Minimizing this cost function with respect to the partition leads to a new spectral clustering algorithm. Minimizing with respect to the similarity matrix leads to an algorithm for learning the similarity matrix. We develop a tractable approximation of our cost function that is based on the power method of computing eigenvectors.

1 Introduction

Spectral clustering has many applications in machine learning, exploratory data analysis, computer vision and speech processing. Most techniques explicitly or implicitly assume a metric or a similarity structure over the space of configurations, which is then used by clustering algorithms. The success of such algorithms depends heavily on the choice of the metric, but this choice is generally not treated as part of the learning problem. Thus, time-consuming manual feature selection is often a necessary precursor to the use of spectral methods.

Several recent papers have considered ways to alleviate this burden by incorporating prior knowledge into the metric, either in the setting of K-means clustering [16, 17] or spectral clustering [18, 10]. In this paper, we consider a complementary approach, providing a general framework for learning the similarity matrix for spectral clustering from examples. We assume that we are given sample data with known partitions and are asked to build similarity matrices that will lead to these partitions when spectral clustering is performed. This problem is motivated by the availability of such datasets for at least two domains of application: in vision and image segmentation, a hand-segmented dataset is now available [11], while for the blind separation of speech signals via partitioning of the time-frequency plane [2], training examples can be created by mixing previously captured signals.

Another important motivation for our work is the need to develop spectral clustering methods that are robust to irrelevant features. Indeed, as we show in Section 4.2, the performance of current spectral methods can degrade dramatically in the presence of such irrelevant features. By using our learning algorithm to learn a diagonally-scaled Gaussian kernel for generating the affinity matrix, we obtain an algorithm that is significantly more robust.

Our work is based on a new cost function $J(W, e)$ that characterizes how close the eigenstructure of a similarity matrix W is to a partition e. We derive this cost function in Section 2. As we show in Section 2.3, minimizing J with respect to e leads to a new clustering algorithm that takes the form of a weighted K-means algorithm. Minimizing J with respect to W yields an algorithm for learning the similarity matrix, as we show in Section 4. Section 3 provides foundational material on the approximation of the eigensubspace of a symmetric matrix that is needed for Section 4.

2 Spectral clustering and normalized cuts

Given a dataset $\mathcal{I}$ of P points in a space $\mathcal{X}$ and a $P \times P$ "similarity matrix" (or "affinity matrix") W that measures the similarity between the P points ($W_{pp'}$ is large when points indexed by p and p' are likely to be in the same cluster), the goal of clustering is to organize the dataset into disjoint subsets with high intra-cluster similarity and low inter-cluster similarity. Throughout this paper we always assume that the elements of W are non-negative ($W \geqslant 0$) and that W is symmetric ($W = W^\top$). We let D denote the diagonal matrix whose i-th diagonal element is the sum of the elements in the i-th row of W, i.e., $D = \text{diag}(W1)$, where 1 is defined as the vector in $\mathbb{R}^P$ composed of ones.

There are different variants of spectral clustering. In this paper we focus on the task of minimizing "normalized cuts." The classical relaxation of this NP-hard problem [15, 19, 3] leads to an eigenvalue problem. In this section we show that the problem of finding a solution to the original problem that is closest to the relaxed solution can be solved by a weighted K-means algorithm.

2.1 Normalized cut and graph partitioning

The clustering problem is usually defined in terms of a complete graph with vertices $V = \{1, ..., P\}$ and an affinity matrix with weights $W_{pp'}$, for $p, p' \in V$. We wish to find R disjoint clusters $A = (A_r)_{r \in \{1,...,R\}}$, where $\bigcup_r A_r = V$, that optimize a certain cost function. An example of such a function is the R-way normalized cut defined as follows [15, 8]:

$$C(A, W) = \textstyle\sum_{r=1}^R \left(\sum_{i \in A_r, j \in V \backslash A_r} W_{ij} \right) / \left(\sum_{i \in A_r, j \in V} W_{ij} \right).$$

Let e_r be the indicator vector in $\mathbb{R}^P$ for the r-th cluster, i.e., $e_r \in \{0,1\}^R$ is such that e_r has a nonzero component exactly at points in the r-th cluster. Knowledge of $e=(e_r)$ is equivalent to knowledge of $A=(A_r)$ and, when referring to partitions, we will use the two formulations interchangeably. The normalized cut is then equal to

$$C(e,W)=\sum_{r=1}^{R} e_r^\top (D-W)e_r / \, (e_r^\top D e_r).$$

2.2 Spectral relaxation and rounding

The following proposition, which extends a result of Shi and Malik [15] for two clusters to an arbitrary number of clusters, gives an alternative description of the clustering task, which will lead to a spectral relaxation:

Proposition 1 *The R-way normalized cut is equal to $R - \operatorname{tr} Y^\top D^{-1/2} W D^{-1/2} Y$ for any matrix $Y \in \mathbb{R}^{R\times P}$ such that (a) the columns of $D^{-1/2}Y$ are piecewise constant with respect to the clusters and (b) Y has orthonormal columns ($Y^\top Y = I$).*

Proof The constraint (a) is equivalent to the existence of a matrix $\Lambda \in \mathbb{R}^{R\times R}$ such that $D^{-1/2}Y = (e_1,\dots,e_R)\Lambda = E\Lambda$. The constraint (b) is thus written as $I = Y^\top Y = \Lambda^\top E^\top D E \Lambda$. The matrix $E^\top D E$ is diagonal, with diagonal elements $e_r^\top D e_r$ and is thus positive and invertible. This immediately implies that $\Lambda\Lambda^\top = (E^\top D E)^{-1}$ (because for the square matrix $M = (E^\top D E)^{1/2}\Lambda$, $M^\top M = I$ implies $MM^\top = I$). This in turn implies that $\operatorname{tr} Y^\top D^{-1/2} W D^{-1/2} Y = \operatorname{tr} \Lambda^\top E^\top W E \Lambda = \operatorname{tr} E^\top W E \Lambda\Lambda^\top = \operatorname{tr} E^\top W E (E^\top D E)^{-1}$, which is exactly the normalized cut (up to an additive constant). ■

By removing the constraint (a), we obtain a relaxed optimization problem, whose solutions involve the eigenstructure of $D^{-1/2}WD^{-1/2}$ and which leads to the classical lower bound on the optimal normalized cut [19, 3]. The following proposition gives the solution obtained from the relaxation:

Proposition 2 *The maximum of $\operatorname{tr} Y^\top D^{-1/2} W D^{-1/2} Y$ over matrices $Y \in \mathbb{R}^{P\times R}$ such that $Y^\top Y = I$ is the sum of the R largest eigenvalues of $D^{-1/2}WD^{-1/2}$. It is attained at all Y of the form $Y = UB_1$ where $U \in \mathbb{R}^{P\times R}$ is any orthonormal basis of the R-th principal subspace of $D^{-1/2}WD^{-1/2}$ and B_1 is an arbitrary rotation matrix in $\mathbb{R}^{R\times R}$.*

Proof Let $M = D^{-1/2}WD^{-1/2}$. The proposition is equivalent to the classical variational characterization of the sum of the R largest eigenvalues $\lambda_1(M),\dots,\lambda_R(M)$ of M (a result known as Ky Fan's theorem, see e.g. [14]):

$$\lambda_1(M) + \cdots + \lambda_R(M) = \max\{\operatorname{tr} Y^\top M Y, Y \in \mathbb{R}^{P\times R}, Y^\top Y = I\},$$

where the maximum is attained for all matrices Y of the form $Y = UB_1$, where $U \in \mathbb{R}^{P\times R}$ is any orthonormal basis of the R-th principal subspace of M and B_1 is an arbitrary rotation matrix in $\mathbb{R}^{R\times R}$. Note that the R-th principal subspace is uniquely defined if $\lambda_R \neq \lambda_{R+1}$ (i.e. there is no eigengap). ■

The solutions found by this relaxation will not in general be piecewise constant. In order to obtain a piecewise constant solution, we wish to find a piecewise constant matrix that is as close as possible to one of the possible Y obtained from the eigendecomposition. Since such matrices are defined up to a rotation matrix, it makes sense to compare the *subspaces* spanned by their columns. A common way to compare subspaces is to compare the orthogonal projection operators on those subspaces [7], that is, to compute the Frobenius norm between $UU^\top$ and $\Pi_0 = \Pi_0(W,e) \triangleq \sum_r D^{1/2} e_r e_r^\top D^{1/2} / \, (e_r^\top D e_r)$. We thus define the following cost function:

$$J(W,e) = \tfrac{1}{2} ||UU^\top - \Pi_0||_F^2. \tag{1}$$

Using the fact that both $UU^\top$ and Π_0 are orthogonal projection operators on linear subspaces of dimension R, a short calculation reveals that the cost function $J(W,e)$ is equal to $R - \operatorname{tr} UU^\top \Pi_0 = R - \sum_r e_r^\top D^{1/2} UU^\top D^{1/2} e_r / \, (e_r^\top D e_r)$.

The cost function $J(W,e)$ provides a spectral characterization of the ability of the matrix W to produce the partition e. For fixed W, minimizing J with respect to e leads to a new spectral clustering algorithm that we now present. Minimizing J with respect to the matrix W, for a given partition e, leads to an algorithm for learning the similarity matrix, as we show in Section 4.

2.3 Minimizing with respect to the partition

In this section, we consider the problem of minimizing $J(W,e)$ with respect to e. The following theorem, inspired by the spectral relaxation of K-means presented in [19], shows that the cost function can be interpreted as a weighted distortion measure, when e is varied for fixed W.[1]

Theorem 1 *Let W be an affinity matrix and let $U = (u_1, \dots, u_P)$, where $u_p \in \mathbb{R}^R$, be an orthonormal basis of its R-th principal subspace. For any partition $e \equiv A$, we have*

$$J(W,e) = \min_{(\mu_1,\dots,\mu_R) \in \mathbb{R}^{R \times R}} \sum_r \sum_{p \in A_r} d_p ||u_p d_p^{-1/2} - \mu_r||^2.$$

Proof Let $D(\mu, A) = \sum_r \sum_{p \in A_r} d_p ||u_p d_p^{-1/2} - \mu_r||^2$. Minimizing $D(\mu, A)$ with respect to μ is an unconstrained least-squares problems and we get:

$$\begin{aligned}
\min_\mu D(\mu, A) &= \textstyle\sum_r \sum_{p \in A_r} u_p^\top u_p - \sum_r || \sum_{p \in A_r} d_p^{1/2} u_p ||^2 / \, (\sum_{p \in A_r} d_p) \\
&= \textstyle\sum_p u_p^\top u_p - \sum_r \sum_{p,p' \in A_r} d_p^{1/2} d_{p'}^{1/2} u_p^\top u_{p'} / \, (e_r^\top D e_r) \\
&= R - \textstyle\sum_r e_r^\top D^{1/2} UU^\top D^{1/2} e_r / \, (e_r^\top D e_r) = J(W,e)
\end{aligned}$$

■

This theorem has an immediate algorithmic implication—to minimize the cost function $J(W,e)$ with respect to the partition e, we can use a weighted K-means algorithm. The

[1] Note that a similar equivalence holds between normalized cuts and weighted K-means for positive semidefinite similarity matrices, which can be factorized as $W = GG^\top$; this leads to an approximation algorithm for minimizing normalized cuts; i.e., we have: $C(W,e) = \min_{(\mu_1,\dots,\mu_R) \in \mathbb{R}^{R \times R}} \sum_r \sum_{p \in A_r} d_p ||g_p d_p^{-1} - \mu_r||^2 + R - \operatorname{tr} D^{-1/2} W D^{-1/2}$.

```
**Input**: Similarity matrix $W \in \mathbb{R}^{P \times P}$.
**Algorithm**:
    1. Compute first $R$ eigenvectors $U$ of $D^{-1/2} W D^{-1/2}$ where $D = \mathrm{diag}(W1)$.
    2. Let $U = (u_1, \ldots, u_P) \in \mathbb{R}^{R \times P}$ and $d_p = D_{pp}$.
    3. Weighted $K$-means: while partition $A$ is not stationary,
        a. For all $r$, $\mu_r = \sum_{p \in A_r} d_p^{1/2} u_p / \sum_{p \in A_r} d_p$
        b. For all $p$, assign $p$ to $A_r$ where $r = \arg\min_{r'} ||u_p d_p^{-1/2} - \mu_{r'}||$
**Output**: partition $A$, distortion measure $\sum_r \sum_{p \in A_r} d_p ||u_p d_p^{-1/2} - \mu_r||^2$
```

Figure 1: Spectral clustering algorithm.

resulting algorithm is presented in Figure 1. While K-means is often used heuristically as a post-processor for spectral clustering [e.g., 13], our approach provides a mathematical foundation for the use of K-means, and yields a specific *weighted* form of K-means that is appropriate for the problem.

2.4 Minimizing with respect to the similarity matrix

When the partition e is given, we can consider minimization with respect to W. As we have suggested, intuitively this has the effect of yielding a matrix W such that the result of performing spectral clustering with that W is as close as possible to e. We now make this notion precise.

The difference between two partitions $e = (e_r)$ and $f = (f_s)$, with R and S clusters respectively, is taken to be [9]:

$$d(e,f) = \frac{1}{2} \left\| \sum_r \frac{e_r e_r^\top}{e_r^\top e_r} - \sum_s \frac{f_s f_s^\top}{f_s^\top f_s} \right\|_F^2 = \frac{R+S}{2} - \sum_{r,s} \frac{(e_r^\top f_s)^2}{(e_r^\top e_r)(f_s^\top f_s)} \qquad (2)$$

This measure is always between zero and $\frac{R+S}{2} - 1$, and is equal to zero if and only if $e \equiv f$. The following theorem shows that if we can perform weighted K-means exactly, we obtain a bound on the performance of our spectral clustering algorithm:

Theorem 2 *Let $\eta = \max_p D_{pp} / \min_p D_{pp} \geqslant 1$. If $e(W) = \arg\min_e J(W,e)$, then for all partitions e, we have $d(e, e(W)) \leqslant 4\eta J(W,e)$.*

Proof Let $S = (e_1, \ldots, e_R) \in \mathbb{R}^{P \times R}$ and $T = (f_1, \ldots, f_R) \in \mathbb{R}^{P \times R}$ where $f = e(W)$. The error measure $d(e,f)$ is the (squared) Frobenius distance $g(S,T)$ between the orthogonal projection operators on the column spaces of S and T and can thus be rewritten as $d(e,f) = g(S,T) = \frac{1}{2} ||S(S^\top S)^{-1} S^\top - T(T^\top T)^{-1} T^\top||_F^2$. Let us define a rescaled error measure $d_D(e,f)$ as the Frobenius distance between the orthogonal projection operators on the column spaces of $D^{1/2}U$ and $D^{1/2}V$:

$$d_D(e,f) = g(D^{1/2}S, D^{1/2}T) = \frac{1}{2} ||D^{1/2} S (S^\top D S)^{-1} S^\top D^{1/2} - D^{1/2} T (T^\top T)^{-1} T^\top D^{1/2}||_F^2.$$

We have, by the triangle inequality for the Frobenius norm:

$$\begin{aligned} d_D(e, e(W))^{1/2} &\leqslant \frac{1}{\sqrt{2}} ||D^{1/2} S(S^\top D S)^{-1} S^\top D^{1/2} - UU^\top ||_F \\ &\quad + \frac{1}{\sqrt{2}} ||D^{1/2} T(T^\top D T)^{-1} T^\top D^{1/2} - UU^\top ||_F \\ &= [J(W,e)]^{1/2} + [J(W, e(W))]^{1/2} \\ &\leqslant 2[J(W,e)]^{1/2} \text{ because } e(W) \text{ is optimal.} \end{aligned}$$

This in turn implies that $d_D(e, e(W)) \leqslant 4J(W,e)$. The result follows from Lemma 1. ■

Lemma 1 *Let S and T be two matrices in $\mathbb{R}^{P \times R}$, with rank R. Let D be a symmetric positive definite matrix in $\mathbb{R}^{P \times P}$. Let $g(S,T)$ denote $\frac{1}{2}||S(S^\top S)^{-1} S^\top - T(T^\top T)^{-1} T^\top||_F^2$ and $\eta = \frac{\lambda_1(D)}{\lambda_P(D)} \geqslant 1$ denote the ratio of the largest and smallest eigenvalue of D. We then have:*

$$\frac{1}{\eta} g(D^{1/2} S, D^{1/2} T) \leqslant g(S,T) \leqslant \eta g(D^{1/2} S, D^{1/2} T) \tag{3}$$

Proof We can expand the Frobenius and rewrite $g(S,T)$ as

$$\begin{aligned} g(S,T) &= \frac{1}{2} \operatorname{tr} \Big\{ S(S^\top S)^{-1} S^\top S (S^\top S)^{-1} S^\top \\ &\quad -2 S(S^\top S)^{-1} S^\top T (T^\top T)^{-1} T^\top + T(T^\top T)^{-1} T^\top T (T^\top T)^{-1} T^\top \Big\} \\ &= \operatorname{tr} \Big\{ I - S(S^\top S)^{-1} S^\top T (T^\top T)^{-1} T^\top \Big\} \\ &= \operatorname{tr} \Big\{ I - (S^\top S)^{-1/2} S^\top T (T^\top T)^{-1} T^\top S (S^\top S)^{-1/2} \Big\} \\ &= \operatorname{tr} \Big\{ (S^\top S)^{-1/2} (S^\top S - S^\top T (T^\top T)^{-1} T^\top S)(S^\top S)^{-1/2} \Big\} \end{aligned}$$

The matrix $S^\top S - S^\top T(T^\top T)^{-1} T^\top S$ is exactly the Schur complement of the top left block in the matrix $M = (S\ T)^\top (S\ T) = \begin{pmatrix} S^\top S & S^\top T \\ T^\top S & T^\top T \end{pmatrix}$. Let $M_D = (S\ T)^\top D (S\ T)$. We have the following inequalities between matrices: $M_D \preceq \lambda_1(D) M$ and $M_D \succeq \lambda_P(D) M$, which implies the same inequality for the top left blocks ($S^\top S$ and $S^\top D S$) and their Schur complements ($N = S^\top S - S^\top T(T^\top T)^{-1} T^\top S$ and $N_D = S^\top D S - S^\top D T (T^\top D T)^{-1} T^\top D S$). We then have:

$$(S^\top S)^{-1/2} N (S^\top S)^{-1/2} \preceq \eta (S^\top D S)^{-1/2} N_D (S^\top D S)^{-1/2}$$

$$(S^\top S)^{-1/2} N (S^\top S)^{-1/2} \succeq \frac{1}{\eta} (S^\top D S)^{-1/2} N_D (S^\top D S)^{-1/2},$$

which implies Eq. (3) by taking the trace. ■

3 Approximation of the cost function

In order to minimize the cost function $J(W, e)$ with respect to W, which is the topic of Section 4, we need to optimize a function of the R-th principal subspace of the matrix $D^{-1/2}WD^{-1/2}$. In this section, we show how we can compute a differentiable approximation of the projection operator on this subspace.

3.1 Approximation of eigensubspace

Let $X \in \mathbb{R}^{P \times P}$ be a real symmetric matrix. We assume that its eigenvalues are ordered by magnitude: $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots \geqslant |\lambda_P|$. We assume that $|\lambda_R| > |\lambda_{R+1}|$ so that the R-th principal subspace E_R is well defined, with orthogonal projection operator Π_R.

Our approximations are based on the power method to compute eigenvectors. It is well known that for almost all vectors v, the ratio $X^q v / ||X^q v||$ converges to an eigenvector corresponding to the largest eigenvalue [7]. The same method can be generalized to the computation of dominant eigensubspaces: If V is a matrix in $\mathbb{R}^{P \times R}$, the subspace generated by the R columns of $X^q V$ will tend to the principal eigensubspace of X. Note that since we are interested only in subspaces, and in particular the orthogonal projection operators on those subspaces, we can choose any method for finding an orthonormal basis of range$(X^q V)$. The QR decomposition is fast and stable and is usually the method used to compute such a basis (the algorithm is usually referred to as "orthogonal iteration" [7]). However this does not lead to a differentiable function. We develop a different approach which does yield a differentiable function, as made precise in the following proposition:

Proposition 3 *Let* $V \in \mathbb{R}^{P \times R}$ *such that* $\eta = \max\limits_{u \in E_R(X)^\perp,\, v \in \text{range}(V)} \cos(u, v) < 1$. *Then the function* $Y \mapsto \widetilde{\Pi}_R(Y) = M(M^\top M)^{-1} M^\top$, *where* $M = Y^q V$, *is* C^∞ *in a neighborhood of* X, *and we have:* $||\widetilde{\Pi}_R(X) - \Pi_R||_2 \leqslant \frac{\eta}{(1-\eta^2)^{1/2}} (|\lambda_{R+1}|/|\lambda_R|)^q$.

Proof It is shown in [7] that $M = X^q V$ has always rank R, so that $M^\top M$ is invertible. Thus in a neighborhood of X, $\widetilde{\Pi}_R(Y)$ is C^∞ since matrix inversion and multiplication are C^∞. The bound is proved in [7] for the QR orthogonal iterations, and since the subspaces computed by the two methods are the same, the bound also holds here. The derivative can easily be computed using the chain rule. ■

This proposition shows that as q tends to infinity, the range of $X^q V$ will tend to the principal eigensubspace. The rate of convergence is determined by the (multiplicative) *eigengap* $|\lambda_{R+1}|/|\lambda_R| < 1$: it is usually hard to compute principal subspace of matrices with eigengap close to one. Note that taking powers of matrices without care can lead to disastrous results [7]. By using successive QR iterations, the computations can be made stable and the same technique can be used for the computation of the derivatives.

3.2 Potentially hard eigenvalue problems

In most of the literature on spectral clustering, it is taken for granted that the eigenvalue problem is easy to solve. It turns out that in many situations, the (multiplicative) eigengap is very close to one, making the eigenvector computation difficult (examples are given in the next section). We acknowledge this potential problem by averaging over several initializations of the original subspace V. More precisely, let $(V_m)_{m=1,\dots,M}$ be M subspaces of dimension R. Let $B_m = \Pi(\text{range}((D^{-1/2}WD^{-1/2})^q V_m))$ be the approximations of the projections on the R-th principal subspace[2] of $D^{-1/2}WD^{-1/2}$. The cost function that we use is the average error

$$F(W, \Pi_0(e)) = \frac{1}{2M}\sum_{m=1}^{M} ||B_m - \Pi_0||_F^2. \tag{4}$$

This cost function can be rewritten as the distance between the average of the B_m and Π_0 plus the variance of the approximations, thus explicitly penalizing the non-convergence of the power iterations. We choose V_i to be equal to $D^{1/2}$ times a set of R indicator vectors corresponding to subsets of each cluster. In simulations, we used $q = 128$, $M = R^2$, and subsets containing $2/(\log_2 q + 1)$ times the number of original points in the clusters.

3.3 Empirical comparisons

In this section, we study the ability of various cost functions to track the gold standard error measure in Eq. (2) as we vary the parameter α in the similarity matrix $W_{pp'} = \exp(-\alpha||x_p - x_{p'}||^2)$. We study the cost function $J(W, e)$, its approximation based on the power method presented in Section 3, and two existing approaches, one based on a Markov chain interpretation of spectral clustering [12] and one based on the alignment [4] of $D^{-1/2}WD^{-1/2}$ and Π_0. We carry out this experiment for the simple clustering example shown in Figure 2(a). This apparently simple toy example captures much of the core difficulty of spectral clustering—nonlinear separability and thinness/sparsity of clusters (any point has very few near neighbors belonging to the same cluster, so that the weighted graph is sparse). In particular, in Figure 2(b) we plot the eigengap of the similarity matrix as a function of α, noting that at the optimum, this gap is very close to one, and thus the eigenvalue problem is hard to solve.

In Figure 2(c) and (d), we plot the four cost functions against the gold standard. The gold standard curve shows that the optimal α lies near 2.5 on a log scale, and as seen in Figure 2(c), the minima of the new cost function and its approximation lie near to this value. As seen in Figure 2(d), on the other hand, the other two cost functions show a poor match to the gold standard, and yield minima far from the optimum.

The problem with the alignment and Markov-chain-based cost functions is that these functions essentially measure the distance between the similarity matrix W (or a normalized version of W) and a matrix T which (after permutation) is block-diagonal with constant blocks. Unfortunately, in examples like the one in Figure 2, the optimal similarity matrix is very far from being block diagonal with constant blocks. Rather, given that data points that lie in the same ring are in general far apart, the blocks are very sparse—not constant and full. Methods that try to find constant blocks cannot find the optimal matrices in these cases. In the language of spectral graph partitioning, where we have a weighted graph with weights W, each cluster is a connected but very sparse graph. The power W^q corresponds

[2] The matrix $D^{-1/2}WD^{-1/2}$ has always the same largest eigenvalue 1 with eigenvector $D^{1/2}1$ and we could consider instead the $(R-1)$-st principal subspace of $D^{-1/2}WD^{-1/2} - D^{1/2}11^\top D^{1/2}/(1^\top D1)$.

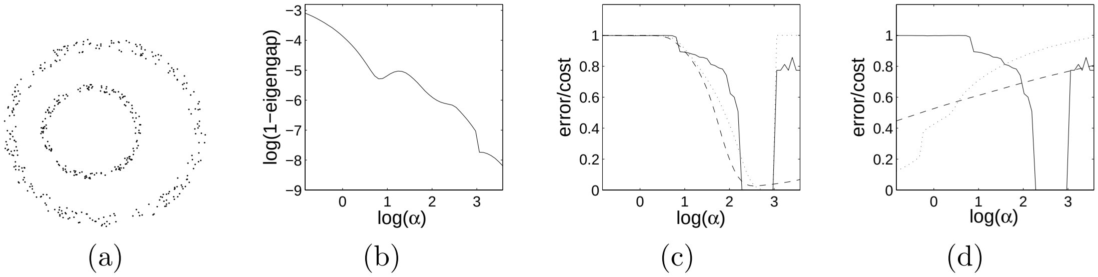


Figure 2: Empirical comparison of cost functions. (a) Data. (b) Eigengap of the similarity matrix as a function of α. (c) Gold standard clustering error (solid), spectral cost function J (dotted) and its approximation based on the power method (dashed). (d) Gold standard clustering error (solid), the alignment (dashed), and a Markov-chain-based cost, divided by 16 (dotted).

to the q-th power of the graph; i.e., the graph in which two vertices are linked by an edge if and only if they are linked by a path of length no more than q in the original graph. Thus taking powers can be interpreted as "thickening" the graph to make the clusters more apparent, while not changing the eigenstructure of the matrix (taking powers of symmetric matrices only changes the eigenvalues, not the eigenvectors).

4 Learning the similarity matrix

We now turn to the problem of learning the similarity matrix from data. We assume that we are given one or more sets of data for which the desired clustering is known. The goal is to design a "similarity map," that is, a mapping from datasets of elements in $\mathcal{X}$ to the space of symmetric matrices with nonnegative elements. To turn this into a parametric learning problem, we focus on similarity matrices that are obtained as Gram matrices of a kernel function $k(x, y)$ defined on $\mathcal{X}\times\mathcal{X}$. In particular, for concreteness and simplicity, we restrict ourselves in this paper to the case of Euclidean data ($\mathcal{X} = \mathbb{R}^F$) and a diagonally-scaled Gaussian kernel $k_\alpha(x, y) = \exp(-(x-y)^\top \operatorname{diag}(\alpha)(x-y))$, where $\alpha \in \mathbb{R}^F$—while noting that our methods apply more generally.

In order to apply our algorithms to domains where the number of data points may be large, efficient implementations are necessary. In Appendix A, we show how the structure of the similarity matrices—sparsity or low-rank approximation—can be used to deal with those potentially large matrices.

4.1 Learning algorithm

We assume that we are given N datasets $\mathcal{D}_n$ of points in $\mathbb{R}^F$. Each dataset $\mathcal{D}_n$ is composed of P_n points x_{np}, $p \in \{1, \ldots, P_n\}$. Each dataset is segmented, that is, for each n we know the partition e_n, so that the "target" matrix $\Pi_0(e_n, \alpha)$ can be computed for each dataset. For each n, we have a similarity matrix $W_n(\alpha)$. The cost function that we use is $H(\alpha) = \frac{1}{N}\sum_n F(W_n(\alpha), \Pi_0(e_n, \alpha)) + C||\alpha||_1$, where $F(W_n(\alpha), \Pi_0(e_n, \alpha))$ is the cost for each dataset, defined in Eq. (4). The ℓ_1 penalty serves as a feature selection term, tending to make the solution sparse. The learning algorithm is the minimization of $H(\alpha)$ with respect to $\alpha \in \mathbb{R}_+^F$, using the method of conjugate gradient with line search.

Since the complexity of the cost function increases with q, we start the minimization with small q and gradually increase q up to its maximum value. We have observed that for small q, the function to optimize is smoother and thus easier to optimize—in particular, the long plateaus of constant values are less pronounced.

Testing The output of the learning algorithm is a vector $\alpha \in \mathbb{R}^F$. In order to cluster previously unseen datasets, we compute the similarity matrix W and use the algorithm of Figure 1. In order to further enhance performance, we can also adopt an idea due to [13]—we hold the direction of α fixed but perform a line search on its norm. This yields the real number λ such that the weighted distortion obtained after application of the spectral clustering algorithm of Figure 1, with the similarity matrices defined by $\lambda\alpha$, is minimum.

4.2 Simulations

We performed simulations on synthetic datasets in two dimensions, where we consider datasets similar to the one in Figure 2, with two rings whose relative distance is constant across samples (but whose relative orientation has a random direction). We add D irrelevant dimensions of the same magnitude as the two relevant variables. The goal is thus to learn the diagonal scale $\alpha \in \mathbb{R}^{D+2}$ of a Gaussian kernel that leads to the best clustering on unseen data. We learn α from N sample datasets ($N=1$ or 10), and compute the clustering error of our algorithm with and without adaptive tuning of the norm of α during testing (as described in Section 4.1) on ten previously unseen datasets. We compare to an approach that does not use the training data: α is taken to be the vector of all ones and we again search over the best possible norm during testing (we refer to this method as “no learning”). We report results in Table 1. Without feature selection, the performance of spectral clustering degrades very rapidly when the number of irrelevant features increases, while our learning approach is very robust, even with only one training dataset.

5 Conclusion

We have presented two algorithms—one for spectral clustering and one for learning the similarity matrix. These algorithms can be derived as the minimization of a single cost function with respect to its two arguments. This cost function depends directly on the eigenstructure of the similarity matrix. We have shown that it can be approximated efficiently using the power method, yielding a method for learning similarity matrices that can cluster effectively in cases in which non-adaptive approaches fail. Note in particular that our new approach yields a spectral clustering method that is significantly more robust to irrelevant features than current methods.

We are currently applying our algorithm to problems in speech separation and image segmentation, in particular with the objective of selecting features from among the numerous features that are available in these domains [2, 15].

Table 1: Performance on synthetic datasets: clustering errors (multiplied by 100) for method without learning (but with tuning) and for our learning method with and without tuning, with $N=1$ or 10 training datasets; D is the number of irrelevant features.

D	no	learning w/o tuning		learning with tuning	
	learning	N=1	N=10	N=1	N=10
0	0	15.5	10.5	0	0
1	60.8	37.7	9.5	0	0
2	79.8	36.9	9.5	0	0
4	99.8	37.8	9.7	0.4	0
8	99.8	37	10.7	0	0
16	99.7	38.8	10.9	14	0
32	99.9	38.9	15.1	14.6	6.1

A Efficient implementation

In applications, such as vision or speech separation, the number of data points can be very large (think of an image with 256 rows and 256 columns of pixels, or a spectrogram with 512 frequency components and 1000 time frames). In order to deal with such matrices, we make use of two possible sources of structure, sparsity or low-rank. Note that the only operations that are required are the products of matrices with vectors. If the matrix is naturally sparse, as is the case with a Gaussian kernel with large α, then matrix products can be done efficiently. If the matrix is not sparse, we then approximate it with a matrix with low-rank, by using only some of its columns to generate the entire matrix. Different techniques can be used to select the columns that span the entire matrix. In this paper, we use the technique of [6], which is based on random selection of columns and Schur complements[3].

Selection of approximation In order to decide which approximation to use (sparse or low-rank), we denote $y_i = \text{diag}(\alpha)^{1/2} x_i$, and we fit a Gamma distribution (using the first two moments) to the set of samples $\{||y_i - y_j||^2, j \neq i\}$ and use the cumulative distribution of the Gamma to predict the degree of sparsity. Note that the first two moments can be computed as: $\mu = 2\alpha^\top \Sigma$, $\sigma^2 = 2\alpha^\top K \alpha - 2(\alpha^\top \operatorname{diag} \Sigma)^2 + 4\alpha^\top(\Sigma \otimes \Sigma)\alpha$, where μ, Σ are the first two moments of (x_i), K is a fourth order moment $K_{rs} = \frac{1}{P}\sum_i (x_{ri} - \mu_r)^2 (x_{si} - \mu_s)^2$, and $\otimes$ denotes the element-wise product of matrices.

Derivatives and line searches The cost function computations can be made linear in the number of points P with the use of low-rank decomposition or sparsity. Assuming that we have the derivative of W with respect to parameters, we immediately get derivatives of all other quantities by using the chain rule. If we are using sparsity, then the derivative

[3] Note that since our aim is to find a low-rank approximation of $D^{-1/2} W D^{-1/2}$ (and not of W), a greedy approach for selecting the columns, such as incomplete Cholesky decomposition [7, 5], would not necessarily select the right columns.

is also sparse, while if low-rank approximations are used, we can use the decomposition as a difference of two low-rank matrices, as used in [1], to get an efficient implementation using low-rank approximations. For line searches, where several evaluations have to be performed in a small neighborhood, we can locally freeze the sparsity pattern or the generating columns, in order to speed up computations.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642), the Multidisciplinary Research Program of the Department of Defense (MURI ONR-N00014-01-1-0890), and a grant from Intel Corporation.

References

[1] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[2] G. J. Brown and M. P. Cooke. Computational auditory scene analysis. *Computer Speech and Language*, 8:297–333, 1994.

[3] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral K-way ratio-cut partitioning and clustering. *IEEE Trans. CAD*, 13(9):1088–1096, 1994.

[4] N. Cristianini, J. Shawe-Taylor, and J. Kandola. Spectral kernel methods for clustering. In *NIPS 14*, 2002.

[5] S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.

[6] C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the Nyström method. In *CVPR*, 2001.

[7] G. H. Golub and C. F. V. Loan. *Matrix Computations.* Johns Hopkins University Press, 1996.

[8] M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for K-way graph clustering and bi-clustering. Technical report, Penn. State Univ, Computer Science and Engineering, 2001.

[9] L. J. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[10] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *IJCAI*, 2003.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[12] M. Meila and J. Shi. Learning segmentation by random walks. In *NIPS 13*, 2002.

[13] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *NIPS 14*, 2001.

[14] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrics. *Mathematical Programming*, 62:321–357, 1993.

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

[16] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained K-means clustering with background knowledge. In *ICML*, 2001.

[17] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS 15*, 2003.

[18] S. X. Yu and J. Shi. Grouping with bias. In *NIPS 14*, 2002.

[19] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for K-means clustering. In *NIPS 14*, 2002.