Journal of Machine Learning Research volume (2006) 1:20 Submitted 1/06; Published ??

Regularized Spectral Learning

Marina Meilă MMP@STAT.WASHINGTON.EDU
Department of Statistics
University of Washington
Seattle, WA 98195

Susan Shortreed SUSAN.SHORTREED@MCGILL.CA
School of Computer Science
McGill University
3480 University St
Montreal, Quebec H3A 2A7
Canada

Editor: Leslie Pack Kaelbling

Abstract

Spectral methods cluster data represented as pairwise similarities between data points. The similarities are considered as given, but in any practical application, they have to be constructed by a domain expert. In this paper we address the problem of automatically learning the similarities as a function of observed features, in order to optimize spectral clustering results on future data. We formulate a new objective for learning in spectral clustering, that balances a clustering quality term, the *gap*, and a stability term, the *eigengap*, with the later in the role of a regularizer. We prove a large eigengap corresponds to clustering stability and that using the eigengap as a regularizer is natural. We derive an algorithm to optimize this objective and choose the optimal regularization. Experiments which confirm the validity of the approach are presented.

Keywords: Spectral Clustering, Learning, Normalized Cut, Eigengap, Regularization, Cluster Stability

1. Introduction

In recent years there has been much progress made in obtaining better spectral clusterings when the similarities between pairs of points are given or constructed by hand. Unfortunately, the similarities between points are rarely known a priori, limiting the application of spectral clustering to situations in which the domain experts can correctly guess features and their optimal combination to obtain a clustering. Constructing the similarity matrix by hand is a time-consuming and subjective process. Moreover, it goes against the grain of many successful supervised approaches in machine learning, which consider a large number of possibly irrelevant features and use training data on to learn the relevant features.

This paper focuses on the problem of automatically estimating the similarity matrix for a spectral clustering task, in a supervised setting. It is assumed that for a set of data points the correct clustering is given, and that one has a set of pairwise features which may be relevant to the correct clustering.

In contrast to previous work Meilă and Shi (2001b); Bach and Jordan (2006); Cour et al. (2005), where the focus is only on defining a quality criterion to be optimized w.r.t the parameters on training data, we define a learning objective that balances fitting the data with a stability term which acts as a regularizer. While this setting is common to many machine learning problems, the form that the two terms take is specific spectral clustering. In particular, the derivation of the regularization term

We start by introducing notation and some basic facts in section 2, we define the learning problem in section 3 and introduce the new objective in 4. Sections 5 and 6 respectively present a gradient algorithm for optimizing the criterion and a method for selecting the amount of regularization. Experimental results are in section 7 and 9 concludes the paper.

2. Spectral Clustering – Notation and Background

In spectral clustering, the data is a set of *similarities* S_{ij}, satisfying $S_{ij} = S_{ji} \geq 0$, between pairs of points i, j in a set V, $|V| = n$. The matrix $S = [S_{ij}]_{i,j \in V}$ is called the similarity matrix. We denote by

$$D_i \equiv \mathit{Vol}\,\{i\} = \sum_{j \in V} S_{ij} \tag{1}$$

the volume of node $i \in V$ and by D a diagonal matrix formed with $D_i, i \in V$. The volume of a set $A \subseteq V$ is $\mathit{Vol}\ A = \sum_{i \in A} D_i$. W.l.o.g we assume that no node has volume 0.

2.1 Random Walks View

Many properties of spectral clustering are elegantly expressed in terms of the stochastic transition matrix P obtained by normalizing the rows of S to sum to 1.

$$P = D^{-1}S \quad \text{or} \quad P_{ij} = S_{ij}/D_i \tag{2}$$

This matrix can be viewed as defining a Markov random walk over V; P_{ij} being the *transition probability* $Pr[i \to j|i]$. The eigenvalues of P are $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq -1$ and the corresponding eigenvectors are $v^1, \ldots v^n$. Note that because $S = DP$ is symmetric, P has real eigenvalues and n linearly independent eigenvectors. Define $[\pi_i]_{i \in V}$ by

$$\pi_i = D_i/\mathit{Vol}\ V \tag{3}$$

It is easy to verify that $P^T\pi = \pi$ and thus that π is a stationary distribution[1] of the Markov chain. For a set $A \subseteq V$, we denote by $\pi_A = \mathit{Vol}\,A/\mathit{Vol}V$ the probability of A under the stationary distribution.

2.2 MNCut Criterion

A clustering $\mathcal{C} = \{C_1, \ldots, C_K\}$ is defined as a partition of the set V into the disjoint nonempty sets $C_1, \ldots, C_K$. The multiway normalized cut (MNCut) is a clustering criterion defined in Meilă (2002); Yu and Shi (2003) and is given by:

$$\mathit{MNCut}(\mathcal{C}) \;=\; \sum_{k=1}^{K} \sum_{k' \neq k} \frac{\mathit{Cut}(C_k, C_{k'})}{\mathit{Vol}\,C_k} \tag{4}$$

where

$$\mathit{Cut}(A, B) \;=\; \sum_{i \in A} \sum_{j \in B} S_{ij} \tag{5}$$

The definition of *MNCut* is best motivated by the Markov random walk view. Define $P_{AB} = Pr[A \to B|A]$ as the probability of the random walk going from set $A \subset V$ to set $B \subset V$ in one step if the current state is in A and the random walk is in its stationary distribution π.

$$P_{AB} \;=\; \frac{\sum_{i \in A, j \in B} \pi_i P_{ij}}{\pi_A} \;=\; \frac{\sum_{i \in A, j \in B} S_{ij}}{\mathit{Vol}\,A} \;=\; \frac{\mathit{Cut}(A, B)}{\mathit{Vol}\,A} \tag{6}$$

It follows that the multiway normalized cut represents the sum of the "out-of-cluster" transition probabilities at the cluster level.

$$\mathit{MNCut}(\mathcal{C}) \;=\; \sum_{k=1}^{K} \sum_{k \neq k'} P_{C_k C'_k} \;=\; K - \sum_{k=1}^{K} P_{C_k C_k} \tag{7}$$

If $\mathit{MNCut}(\mathcal{C})$ is small for a certain partition $\mathcal{C}$, then the probabilities of leaving C_k, once the walk is in it, is small.

In Meilă (2002) it is shown that the $\mathit{MNCut}(\mathcal{C})$ for any clustering $\mathcal{C}$ is bounded below by a function of the number of clusters $K = |\mathcal{C}|$ and of the eigenvalues of P:

$$\mathit{MNCut}(\mathcal{C}) \;\geq\; K - \sum_{k=1}^{K} \lambda_k(P) \tag{8}$$

We call the non-negative difference between the *MNCut* and its lower bound the *gap*:

$$\mathit{gap}_P(\mathcal{C}) \;=\; \mathit{MNCut}(\mathcal{C}) - K + \sum_{k=1}^{K} \lambda_k(P) \tag{9}$$

1. A word on this definition. If a Markov chain is *irreducible* and *aperiodic* it is guaranteed to have a unique stationary distribution Norris (1997). In our case the symmetry of S guarantees aperiodicity. Irreducibility is equivalent to requiring that the graph defined by the non-zero values of S_{ij} be connected. This assumption is unacceptable and, as it turns out, unnecessary in the case of clustering and we do not make it. As a consequence, the Markov chains considered here may have more than one stationary distributions. However, in all cases, the distribution π defined by (3), is one of them, and the only one of interest for this paper. With a slight abuse of terminology we therefore call π *the* stationary distribution.

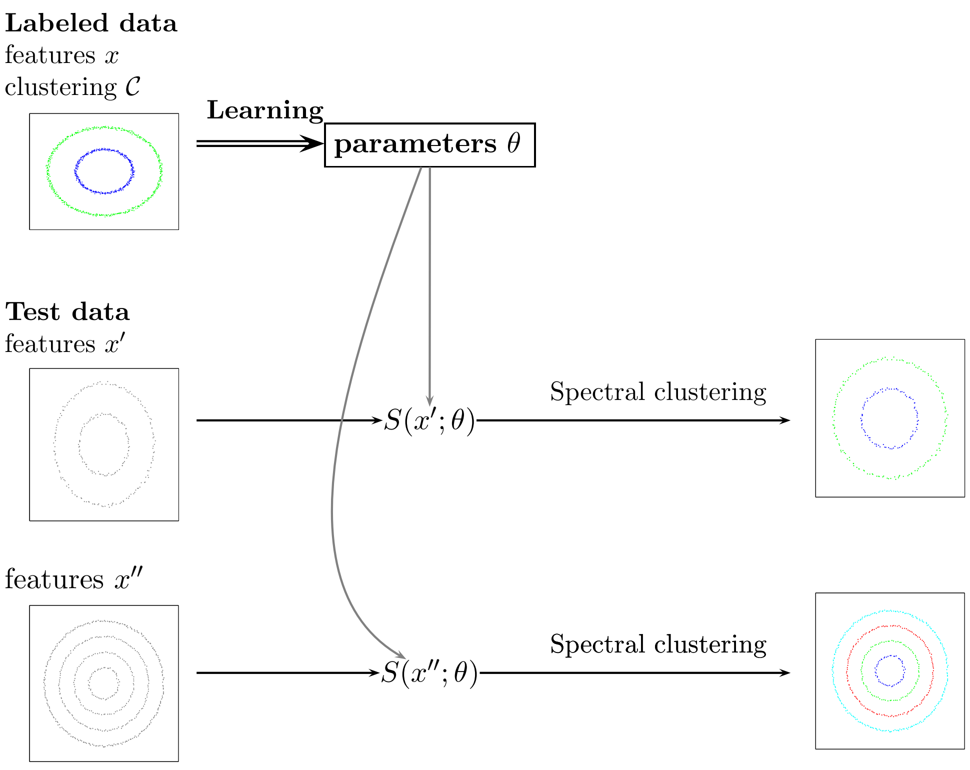


Figure 1: The supervised learning paradigm in spectral clustering. From a training set with a known clustering, one estimates the parameters θ of the similarity function $S(x;\theta)$. When a new data set is presented for clustering, the learned parameters are used to construct the S matrix for this clustering problem. Then a standard spectral clustering algorithm applied to S clusters the data.

An eigenvector v of P is said to be *piecewise constant* with respect to a clustering iff whenever i and j are in the same cluster $v_i = v_j$. One can show Meilă (2002) that the gap is 0, i.e. the *MNCut* has been minimized, iff the first K eigenvectors of P $v^1, \ldots v^K$ are piecewise constant w.r.t $\mathcal{C}$. In this case we say that P has *piecewise constant eigenvectors (PCE)*.

This fact has bearing on learning in two ways: first, being an if and only if result, shows that bringing the integrality gap to 0 is sufficient and necessary for a clustering to be good w.r.t to the *MNCut* criterion. Second, it seems to impliy that from the point of view of the clustering quality, it is the eigenvectors not the eigenvalues that are relevant. The present paper will show that, while this latter statement is not wrong, the eigenvalues are still relevant, from the point of view of clustering stability.

3. The Learning Problem

3.1 Problem Set-Up

We assume a clustering scenario where for each pair of points in the set to be clustered a set of *features* is observed. Let the data set be $\mathcal{D} = \{1, 2, \dots n\}$ and let

$$x_{ij} \;=\; [x_{ij,1}\; x_{ij,2}\; \dots x_{ij,F}]^T \tag{10}$$

denote the F-dimensional vector of features for pair $i, j \in \mathcal{D}$. The features are symmetric, that is $x_{ij} = x_{ji}$ for all i, j.

One can think of the data points as vectors in some F-dimensional space, with the features representing distances between points along the F coordinate axes. For instance, in figures 1 and 4 each pair of points is described by two features, the distances along the two coordinate axes. While using distances as pairwise features is widely spread practice in spectral clustering, our formulation is significantly more general, in that it accommodates features that do not come from a Euclidean space representation of the data. For example, in an image segmentation task, a feature $x_{ij,f}$ may indicate the presence of an edge between pixels i and j of the image. In a social network, $x_{ij,f}$ may measure degree of friendship between individuals i and j, or number of joint papers between two authors. One sees from these examples that the features can measure both similarity and dissimilarity. From the feature vector we construct the *similarity function* $S(x_{ij}; \theta)$ which summarizes in a single number the degree of similarity between i and j. This function depends on a set of parameters collected in the vector θ.

A similarity functions we use later in the paper is $S(x; \theta) = e^{-\theta^T x}$, with $\theta \in \mathbb{R}^F$, $\theta_f \geq 0$ and $x_{ij,f} \geq 0$ representing a dissimilarity or distance. The similarity is assumed to be an (almost everywhere) differentiable function $S(x; \theta)$ that maps a feature vector $x \in \mathbb{R}^F$ and $\theta \in \mathbb{R}^F$, a vector of parameters, into a non-negative scalar similarity[2]. In addition we require the S is an increasing function w.r.t those features $x_{ij,f}$ that represent similarity (like joint papers) and a decreasing function of the features which represent dissimilarity (like distances or intervening contour).

Let

$$\begin{aligned}
S_{ij} \;&=\; S(x_{ij}; \theta) \quad \text{for } i, j = 1, \dots n && (11)\\
\mathbf{x} \;&=\; [x_{ij}]_{i,j=1,\dots n} \;\in\; \mathbb{R}^{n \times n \times F} && (12)\\
S(\theta) \;&\equiv\; S(\mathbf{x}; \theta) \;\equiv\; [S_{ij}]_{i,j=1,\dots n} \;\in\; \mathbb{R}^{n \times n} && (13)
\end{aligned}$$

Here, the letter S denotes both the similarity function $S(x_{ij}; \theta)$, and the similarity matrix $S(\theta)$ for fixed data set $\mathbf{x}$ and parameter vector θ. The matrices obtained from $S(\theta)$ by (1,2) are respectively denoted $D(\theta)$, $P(\theta)$.

We would like to learn the optimal parameters θ of the similarity function in a supervised setting. Therefore we assume that we are given a *training set* $\mathcal{D}$ together with its correct clustering $\mathcal{C}^*$. The *learning problem* is formulated as: Given data $\mathbf{x}$, the target clustering $\mathcal{C}^*$ and the similarity function $S(.\,;\,.)$, find the optimal parameters θ w.r.t a criterion $J(\mathcal{C}^*,\, S(\mathbf{x}; \theta))$ to be defined in the next section.

2. For the rest of this paper, the dimension of the parameter vector θ equals the dimension of the feature vector x_{ij}. This is assumed only to simplfy the exposition. All our results hold for general parametric functions $S(x; \theta)$ observing the other stated conditions.

Once the parameters θ are learned, they can be applied to future test sets to obtain a good similarity matrix and corresponding clustering of those sets. This paradigm is summarized by figure 1. For simplicity we explain the algorithm using only one training data set. The generalization to more than one data set is immediate and simply requires averaging the cost function over all of the data sets.

Note that there are fundamental differences between the supervised learning problem stated here and supervised learning for classification. In a classification task, each class has a well-defined meaning (i.e class labels are not interchangeable) and the classifier encodes this information. Our $S(x;\theta)$ only aims to encode which pairs of points are similar, and it has no information on how the points should be labeled. The S function is often compared or even substituted with a kernel function. While this is possible, one should not forget that in a kernel classifier, the kernel defines a mapping but does not alone define a classifier; the w vector or alternatively the support vectors carry much of this information. It has even been argued that in many cases it is the support vectors that are more important than the kernel. By contrast, in the case of spectral clustering, what is carried on to future data sets is just the parameteric function $S(\ ;\theta)$.

This paradigm is also different from semi-supervised learning, where a large body of work relates to spectral clustering. This work is exemplified by the papers of Szummer and Jaakkola (2001); Xing et al. (2002); Zhu et al. (2002); Chapelle et al. (2003); Li and Wu (2003); Kulis et al. (2005); Shortreed (2006). In the semi-supervised framework, some points of the data set are labeled, and the rest (typically the majority), unlabeled. The task would be then to adjust the parameters *and* cluster so that the unlabeled points are clustered with those of the correct class. In contrast, in our paradigm we use a fully labeled data set, then transfer the parameters learned to new data sets.

Another aspect is that once the similarity S is learned, we need not restrict the number of clusters of the test set to equal that of the training set. In the Bull's Eye experiment presented in section 7 the training set had 2 clusters but the learned θ parameters were subsequently used to cluster data sets with 2, 3, 4 etc clusters. The same θ parameters, once learned, can be used to cluster other data sets, as long as the meaning and scaling of the features remains the same.

3.2 Previous Work

This learning problem was proposed in Meilă and Shi (2001b); there, the authors introduce a target S^* having $S^*_{ij} = 1$ if i, j are in the same cluster and $S^*_{ij} = 0$ otherwise. The parameters are then optimized by making $S(\theta)$ match S^* in a KL-divergence sense that reflects the random walk interpretation of the similarity matrix. This approach worked well on image segmentation data, but is not appropriate for general purpose learning. For any clustering there can be an infinite number of S matrices that are perfect for that clustering. Imposing any one as a target S^* will over-constrain the problem and is equivalent to introducing a bias, which may or may not fit the problem at hand.

Bach and Jordan (2006) used the angle between the subspace spanned by the true cluster indicator vectors and the subspace spanned by the eigenvectors of $P(\theta)$ as the learning criterion, thus dispensing with the target S^* and the potential bias it may introduce. The angle is minimized when $P(\theta)$ has PCE's for the given clustering, which is also when the spectral clustering algorithm returns the clustering with the minimized MNCut. This approach has shown promising results, but the need to differentiate a function of the eigenvectors of a matrix w.r.t. the matrix elements makes the approach difficult to implement and numerically unstable.

In Cour et al. (2005) the distance between the true indicator vector and the eigenvector of interest is used as the optimization criterion. They calculate the exact analytical form of the derivatives and prove that their algorithm converges. This approach should not over-constrain the parameters, but the number of parameters learned is dependent on the size of the data set. This setting makes generalization to additional is data sets difficult. While technique has shown positive results in image segmentation, the extension of the technique to other clustering applications is not straightforward.

Finally, we mention another interesting use of the eigengap in spectral clustering: Azran and Ghahramani (2006) use of the eigengaps of the transition matrix P raised to power for the purpose of finding the optimal number of clusters at different scales.

4. The objective

In this paper, we address learning by explicitly enforcing that the learned parameters induce a good and stable clustering on the training data $(\mathbf{x}, \mathcal{C}^*)$ while not over-constraining the parameter space. We do this by optimizing a clustering quality term and using a clustering stability term as a regularizer.

4.1 Clustering Quality

In the context of spectral clustering the quality of a clustering can be measured using either its *MNCut* or its integrality gap. We can explicitly enforce that the learned parameters induce a good clustering on the training data by forcing the integrality gap or *MNCut* of the true clustering $\mathcal{C}^*$ w.r.t. $S(\theta)$ to be small. For a *fixed* matrix S, a clustering that minimizes the *MNCut* also minimizes the integrality gap, so in this situation the criteria are equivalent. However, if one *learns* S, then the criteria are not equivalent: obtaining a small *MNCut* implies that the off-diagonal blocks of S are nearly 0, while a small integrality gap does not carry such an implication. Hence, the integrality gap puts fewer constraints on θ while still being an indicator of a good clustering, and we choose it as a criterion of clustering quality.

4.2 Clustering Stability

To enforce clustering stability, we will maximize the eigengap $\Delta_K = \lambda_K(P(\theta)) - \lambda_{K+1}(P(\theta))$. To motivate this choice, one can recall the fact that a large eigengap in P makes the subspace spanned by $v^1 \dots v^K$ stable to perturbations Stewart and Sun (1990). In what follows we prove an even stronger connection between the eigengap and clustering stability. Namely, if the eigengap of P is sufficiently large, then all clusterings with a small integrality gap are similar.

For two clusterings with $|\mathcal{C}| = K \leq |\mathcal{C}'| = K'$ we define the distance between $\mathcal{C}$ and $\mathcal{C}'$ by

$$d(\mathcal{C}, \mathcal{C}') \; = \; 1 - \max_{\pi \in \Pi_{K'}} \sum_{k=1}^{K} Vol(C_k \cap C'_{\pi(k)}) \tag{14}$$

where $\Pi_{K'}$ denotes the set of permutations of K' objects. This distance is inspired by the well known misclassification error cost function. In the case of clustering, as the correspondence between labels in $\mathcal{C}$ and $\mathcal{C}'$ is not known, hence the minimization over all possible 1-to-1 label mappings. This distance is a metric (see e.g Meilă (2005)) and although the maximization above is over a set of size $(K')!$, d can be computed in polynomial time by a maximum bipartite matching algorithm Papadimitriou and Steiglitz (1998).

Theorem 1 *Let $\mathcal{C}, \mathcal{C}'$ be two K-way clusterings with $gap(\mathcal{C})$, $gap(\mathcal{C}') \leq \varepsilon$ and $\delta = \frac{\varepsilon}{\Delta_K}(\sqrt{K}+1)^2$, where $p_{max} = \max_k VolC_k/VolV$. Then, whenever $\delta \leq p_{min}$ we have that $d(\mathcal{C}, \mathcal{C}') \leq \delta p_{max}$.*

Corollary 2 *Let $\mathcal{C}$ be a K-way clustering, let $\delta = \frac{gap(\mathcal{C})}{\Delta_K}(\sqrt{K}+1)^2$, and let $\mathcal{C}^* = \underset{|\mathcal{C}'|=K}{\operatorname{argmin}}\, MNCut(\mathcal{C}')$. Then $d(\mathcal{C}, \mathcal{C}^*) \leq \delta p_{max}$ whenever $\delta \leq p_{min}$.*

In words, the above theorem and its corollary show that, if we find a clustering with a small enough integrality gap relative to the eigengap, then that clustering is also "stable", in the sense that any other clustering with small gap will necessarily be close to it. If the eigengap is sufficiently large w.r.t to the best attainable integrality gap, then there is essentially a unique way of obtaining good partition in that P. Any two partitions with a small integrality gap have to be close to each other.

4.3 The learning criterion

Now we define learning in spectral clustering as solving the following optimization problem

$$(\mathcal{P}) \qquad \min\, gap_\theta(\mathcal{C}^*) \tag{15}$$
$$\text{s.t. } \Delta_K^2(P(\theta)) \;\geq\; \varepsilon \tag{16}$$

where gap_θ is a short form for $gap_{P(\theta)}$. By simultaneously achieving a small integrality gap for $\mathcal{C}^*$ and a large eigengap for $P(\theta)$, we enforce that $\mathcal{C}^*$ is both a "good" and a "stable" clustering (all other clusterings with small gap are close to $\mathcal{C}^*$). Hence, $(\mathcal{P})$ enforces that $\mathcal{C}^*$ is not only a good clustering, but it is the *only good clustering*, up to small perturbations.

There is another way of motivating this choice. Note, for instance, that if we only impose the condition that the integrality gap of $\mathcal{C}$ is small, there are many potential θ's and corresponding $S(\theta)$'s that can satisfy it, including the unit matrix I_n. For some of these, e.g for $S(\theta) \approx I_n$, there are very many clusterings different from $\mathcal{C}^*$ that have small integrality gap. By a description length argument, to specify $\mathcal{C}^*$ given that the gap is small, we need to transmit which of all the possible clustering with small integrality gap is $\mathcal{C}^*$, adding the number of bits necessary for encoding this to the description length of the solution. However, if the eigengap Δ_K is above some threshed w.r.t the gap, then by theorem 1, $\mathcal{C}^*$ is the unique clustering minimizing the integrality gap, and we save the extra bits in the description length.

The constrained optimization problem (15-16) can be turned into the unconstrained problem

$$(\mathcal{P}') \quad \min_\theta\, gap_\theta(\mathcal{C}^*) - \alpha(\Delta_K^2(P(\theta)) - \epsilon) \tag{17}$$

The parameter $\alpha \geq 0$ is a Lagrange multiplier. Its value depends on ϵ. But because the value of α above controls the trade-off between the two goals of reducing the integrality gap and enlarging the eigengap, one can also see α as a regularization parameter. The choice of this parameter will be addressed in section 6, where we give a method for semi-automatically selecting the value of α.

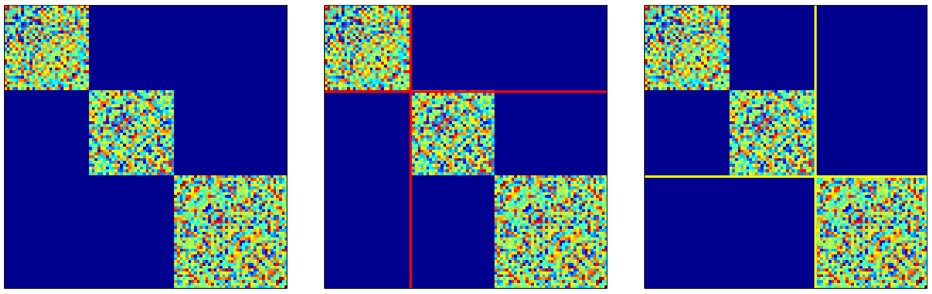


Figure 2: A similarity matrix with 3 blocks (left) and two clusterings with $K = 2$ clusters (middle and right) marked by the red, respectively yellow lines in the matrix. Both clusterings have $MNCut = 0$ but $\Delta_2 = 0$ as well.

One can also consider other formulations that balance the integrality gap and eigengap. For example, according to theorem 1, a natural optimality criterion would be $J' = gap_\theta(\mathcal{C})/\Delta_K(P(\theta))$. We chose to optimize $(\mathcal{P})$ over J' because the latter contains a division by the eigengap. Since the eigengap is typically a small number computed as the difference of two consecutive eigenvalues, such an operation is unstable numerically. We have also noticed that, even though the eigengap is typically very small (10^{-2} to 10^{-4} for the learned solutions), it can have large derivatives (see for example Figure 12); the exponent 2 in $(\mathcal{P})$ has the effect of smoothing out the derivatives, so that the effect of the eigengap is on the general location of the optimal θ, but not on their precise values, which should be tuned according to the integrality gap.

To better understand why stability and the eigengap are important in learning for spectral clustering, consider the case presented in figure 2 where Theorem 1 does not hold. The figure presents a block diagonal similarity matrix with 3 blocks, and two clusterings $\mathcal{C}, \mathcal{C}'$ with $K = 2$ clusters. Both clusterings have zero $MNCut$ and zero integrality gap, but they are very different from one another. This is possible because the eigengap Δ_2 for this matrix is 0. If K were equal to 3 then it would be impossible to find a 3-clusterings with small integrality gap, which differs much from the obvious perfect 3-clustering of this matrix.

For another example, consider the case where $S \approx I$ the unit matrix. If one tries to learn parameters that minimize the $MNCut$, it is often the case that the resulting S tends to the unit matrix. This is very good from the point of view of the optimization criterion, since *any* clustering will have a vanishingly small $MNCut$, but it is very bad because the algorithm will have failed to learn anything about the target clustering $\mathcal{C}^*$. Hence, in order to learn from $\mathcal{C}^*$ we need not only to make it be a good clustering w.r.t $S(\theta)$ but to also ensure that it is the *only* clustering (subject to small perturbations) which is good for this $S(\theta)$.

5. Optimizing the parameters

Here we present the solution to problem $(\mathcal{P}')$. Unlike the SVM formulation, in our optimization problem neither the objective nor the constraint are convex. Therefore, we optimize the parameters by following the gradient, starting from equal values for all entries in θ.

In this section, we will consider optimizing J_α in (17) w.r.t. θ holding α fixed. The next section will discuss the choice of α. The cost J_α can be expressed as

$$\begin{aligned} J_\alpha(\theta) &= \left[MNCut_{P(\theta)}(\mathcal{C}^*) - K + \sum_{k=1}^{K} \lambda_k(P(\theta)) \right] - \alpha \left[\lambda_K(P(\theta)) - \lambda_{K+1}(P(\theta)) \right]^2 \\ &= \underbrace{\sum_{k=1}^{K} \frac{\sum_{i,j \in k} S_{ij}}{\sum_{i \in k} \sum_{j=1}^{n} S_{ij}}}_{J_1(\theta)} -K + \underbrace{\sum_{k=1}^{K} \lambda_k(L(\theta)) - \alpha \left[\lambda_K(L(\theta)) - \lambda_{K+1}(L(\theta)) \right]^2}_{J_2(\theta)} \end{aligned}$$

The derivative of J_1 w.r.t to θ is straightforward, assuming that the partial derivatives $\frac{\partial S}{\partial \theta}$ can be computed tractably. In the following we show how to obtain the gradient of the second term, which involves the eigenvalues of $L(\theta)$. Evaluating $\frac{\partial L_{ij}}{\partial \theta}$ presents no problem, so we focus on the derivatives of eigenvalues and of sums of eigenvalues w.r.t to the elements of L.

It is known Magnus and Neudecker (1999) that for a symmetric matrix L, the derivative of a simple eigenvalue λ w.r.t the matrix elements has the expression

$$\frac{\partial \lambda}{\partial L} = vv^T \tag{18}$$

where v is the eigenvector corresponding to λ. If λ is a multiple eigenvalue, then $\lambda(L)$ is not differentiable at L. One then defines the *Clarke generalized gradient* Clarke (1983), which is, intuitively, the convex set of all directions of "maximum increase" of a function and coincides with the gradient whenever the function is continuously differentiable. In our case, the Clarke gradient (also denoted by $\frac{\partial \lambda}{\partial L}$) is given by the formula

$$\frac{\partial \lambda}{\partial L} = \operatorname{conv}\{uu^T \,|\, u \in R^n,\ Lu = \lambda u\} \tag{19}$$

In practice, when two of the eigenvalues of $L(\theta)$ are too close, the evaluation of the gradient becomes numerically unstable. When optimizing (17), reducing the *MNCut* term has the tendency to push the first K eigenvalues toward 1, thus sending the optimization trajectory into an instability zone. To overcome the numerical instability in our experiments, we implemented the robust method of Burke et al. (2005). This method essentially detects when the gradient is unstable and applies a sampling technique to find a robust direction of descent.

However, it is worth noting the stabilizing effect played by the eigengap term in (17). Enforcing one large eigengap, Δ_K, for this problem, has the effect of preventing all of $\lambda_1, \ldots \lambda_K$ from becoming too close to each other. As a consequence, in our experiments, the simple gradient given by formula (18) is stable for all but extremely large values of α. We stress that this is an effect of our parametrization of S and that it is not guaranteed to happen with another S parametrization.

One can also prove that preserving the eigengap Δ_K is sufficient to make the cost J differentiable.

Proposition 3 *For any symmetric matrix $L(\theta)$, if L is a continuously differentiable function of θ_f with $\frac{\partial L}{\partial \theta_f}(\theta_f^0) \neq 0$ and $\Delta_K(L(\theta)) > 0$ then the gradient $\frac{\partial J_2}{\partial \theta_f}$ exists and is continuous. Denote by V, U_K, U_{K+1} the subspace spanned respectively by the first K eigenvectors* *of L, the eigenvectors corresponding to eigenvalue λ_K, and the eigenvectors corresponding to eigenvalue λ_{K+1} of $L(\theta^0)$. Let $\beta_{1:r}$ be the eigenvalues of $U_K^T \frac{\partial L}{\partial \theta_f}(\theta^0) U_K$ and $\beta'_{1:r'}$ be the eigenvalues of $U_{K+1}^T \frac{\partial L}{\partial \theta_f}(\theta^0) U_{K+1}$.*

$$\frac{\partial J_2}{\partial \theta_f} \;=\; \operatorname{trace} V^T \frac{\partial L}{\partial \theta_f}(\theta^0) V - 2\alpha \Delta_K(\theta^0)(\min\{\beta_{1:r}\} - \max\{\beta'_{1:r'}\}) \qquad (20)$$

Proof One can recast the first term in $\frac{\partial J_2}{\partial \theta_f}(\theta^0)$ as $\operatorname{trace} V^T \frac{\partial L}{\partial \theta_f}(\theta^0) V$. Since the K-th eigengap is non-zero, the latter expression is well defined even if some of the first K eigenvalues are equal. For the second term, we need only analyze the case when there λ_K or λ_{K+1} is a multiple eigenvalue. In that case, if $\frac{\partial L}{\partial \theta_f}(\theta^0)$ is non-zero, the derivatives of the r eigenvalues equal to $\lambda_K(\theta^0)$ exist and are given by $\beta_{1:r}$ Lancaster (1969). Similarly, the derivatives of the r' eigenvalues equal to $\lambda_{K+1}(\theta^0)$ are given by $\beta'_{1:r'}$. The change in the eigengap is given by the smallest increase in λ_K minus the largest increase in λ_{K+1}, which completes the proof.

5.1 Computation

Each gradient step comprises an evaluation of $\frac{\partial J_\alpha}{\partial \theta}$ for the descent direction and several evaluations of J_α for the line search. With the features $\mathbf{x}$ given, computing S takes $\mathcal{O}(n^2 F)$ operations, computing the *MNCut* and $\frac{\partial MNCut}{\partial S}$ takes $\mathcal{O}(n^2)$ and evaluating the partial derivatives $\frac{\partial S}{\partial \theta}$, $\frac{\partial L}{\partial \theta}$ requires another $\mathcal{O}(n^2 F)$. To complete the evaluation of J_α and of its gradient, we also need the first $K+1$ eigenvalues and eigenvectors of $L(\theta)$. We compute them with the Matlab `eigs` function that calls an iterative procedure whose time per iteration is nK', where K' is the number of eigenvalues required. In our experiments, we use $K' = \max(2K, K+5)$. Thus the running time per gradient step is $\mathcal{O}(n^2 F + nK')$.

We also use line search along the direction of descent to find the optimal step size at each iteration. The procedure we use is the Armijo rule Bertsekas (1999). This takes a step of size τ the first time when $J_\alpha(\theta) - J_\alpha(\theta - \tau \frac{\partial J_\alpha}{\partial \theta}) > \beta || \frac{\partial J_\alpha}{\partial \theta} ||_2$ with $\beta = 10^{-2}$, otherwise τ is reduced. This algorithm can be easily tuned so that in practice most steps are taken after just 1–2 function evaluations. The implementation of the adaptive step size is however not superfluous, as typically at the beginning and at the end of the learning, smaller step sizes are chosen. With the adaptive step size, the gradient descent usually takes less than 100 steps to attain a good set of θ values; attaining convergence once near the optimum is slower, typical of gradient algorithms.

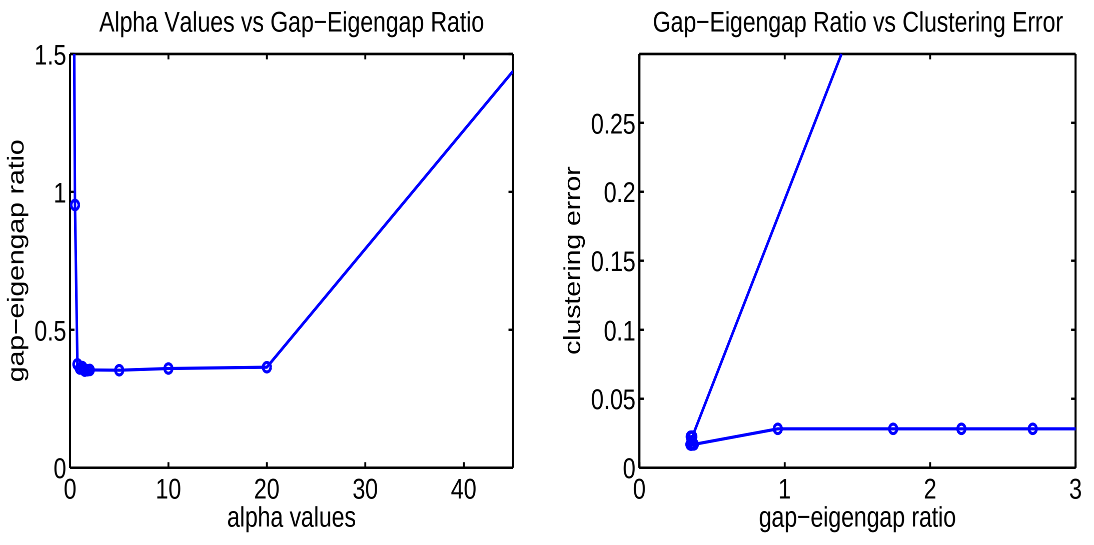
6. Selecting the regularization parameter

For the problem to be completely formulated, one needs to specify the value of ϵ in $(\mathcal{P})$, or, equivalently, the value of α in $(\mathcal{P}')$. Since ε is an integrality gap, from (8) and (9) we have that $0 \le gap_P(\mathcal{C}) \le MNCut(\mathcal{C}) \le K-1$. Thus, as a rule of thumb for the choice of ε, a value in $[10^{-4 \cdots -1}, 1]$ should represent a good enough quality for $\mathcal{C}^*$. We arrived at the lower limit 10^{-4} after observing that almost uniformly in our experiments this value was associated with good results, while smaller values typically gave bad clustering results. If ϵ is chosen too small, i.e much smaller than $10^{-}4$, $(\mathcal{P})$ may have no solution.

However, the above range is too large to be useful without an additional method to narrow it down. An obvious possibility is to use cross-validation for this purpose. To perform cross-validation, one needs to have one or more clustered data sets to use as validation sets. With them, the procedure is straight-forward.

In what follows, we introduce a simple alternative to cross-validation, that can be used without a validation set. This method ties in with our optimization criterion for learning the parameters, θ. We would like to select the regularizing parameter which will learn feature parameters producing a small gap and a large eigengap. Thus we choose α based on the gap-eigengap ratio. Recall that both the integrality gap *gap* and the eigengap Δ_K increase when the regularization parameter α is increased. Since the learning objective balances the two, the gap-eigengap ratio should be large for extreme values of α, both large and small, and lower at intermediate values. We will choose the regularization parameter α that minimizes this ratio.

```
Algorithm Select Alpha
  1. Choose a set $A$ of $\alpha$ values spanning a reasonable range, e.g $[10^{-2}, 10^2]$
  2. For $\alpha \in A$
     (a) Find $\hat{\theta}_\alpha = \underset{\theta}{\operatorname{argmin}}\ J_\alpha(\theta)$
     (b) Compute $r_\alpha = \frac{gap_{\hat{\theta}_\alpha}(\mathcal{C}^*)}{\Delta_K(P(\hat{\theta}_\alpha))}$
  3. Choose $\alpha^*$ to minimize $r_\alpha$, in the event of ties choose the $\alpha^*$ with the largest eigengap.
```



(a) (b)

Figure 3: (a) The gap-eigengap ratio as a function of α for the Bull's eye data (described in Section 7.2 and Figure 4); (b) The clustering error CE on a test set vs the gap-eigengap ratio on the training set, when α traverses the range in (a); the very small α's correspond to large CD. The minimal values of the test set error coincide with the minimal values of the gap-eigengap ratio in (a). Note also that the optimal range of α covers at least one order of magnitude.

The right panel of Figure 3 shows that our method gives essentially the same results at CV, since the test set error reaches its minimum in the same time as the gap-eigengap ratio. We only present one graph here, but we have seen the same phenomenon on all the data sets we experimented with.

7. Experiments

7.1 Procedure

In this section we report experimental results obtained with the learning algorithm presented. The similarity function used is defined as $S(x, \theta) = e^{-\theta^T x}$. All the x features were non-negative dissimilarity features ($x_{ij,f} \geq 0$, $x_{ij,f}$ nearer 0 meaning that i, j are more similar). Consequently, the parameters θ_f were assumed non-negative. The next section describes in more detail how the features were obtained for each experiment.

The initial parameter values were chosen to be equal for all features. The learning algorithm described in section 5 was run for a set of prespecified α values covering the range $\{0.01,\ 0.1,\ 0.2,\ 0.5,\ \ldots,\ 1000\}$. After learning, the optimal α^* value was chosen using the Select Alpha algorithm of section 6. To save time, when several identical replicates of the same experiment were run, the α selection was done only once, on the first experiment. The chosen α^* was then used in all the other replications of the experiment. The parameters θ^* corresponding to the selected α^* were output as the learned parameters from each experiment.

For the evaluation of learning we used separate test sets. Using the features of the test set and the learned parameters θ^* we constructed the matrices $S(\mathbf{x}^{test}; \theta^*)$. This similarity matrix was given as input to a standard spectral clustering algorithm. In our case this was the algorithm described in Meilă and Shi (2001a), in which the first K eigenvectors of the corresponding $P(\mathbf{x}^{test}; \theta^*)$ matrix are clustered using K-means, with multiple random and orthonormal initializations. For more information on initializations see Verma and Meilă (2003). We measure the difference between the clusterings obtained and the true clustering by *clustering error (CE)* which is the unweighted misclassification error, i.e

$$CE(\mathcal{C}, \mathcal{C}') \;=\; 1 - \max_{\pi \in \Pi_{K'}} \sum_{k=1}^{K} |C_k \cap C'_{\pi(k)}| \qquad \text{assuming } K \leq K' \tag{21}$$

7.2 Data

Bull's eye. The first experiments are performed on simulated data, the bull's-eye in two dimensions; Figure 4 displays a sample data set. The data consist of an inner ring containing approximately 40% of the data points with the remaining data points forming an outer ring. While this data is artificial, and a very easy data set to cluster with a manually chosen similarity matrix, it has several features that make learning the features a non-trivial task. Each point has only a small number of neighbors in its own cluster; the majority of within cluster distances are large, comparable to the distances to other clusters. There is a high possibility of over-fitting. The two meaningful features we used were $x_{ij,l} = |y_{il} - y_{jl}|$; where y_{i1} is the horizontal coordinate and y_{i2} is the vertical coordinate associated with the i^{th} point. To these we added a number F_{noise} of noise features, which were symmetric random matrices designed to have the same mean and variance as the meaningful features. As an added difficulty for learning this data set, the meaningful features each taken separately do

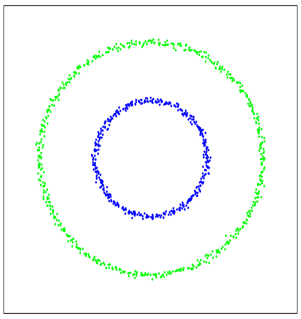


Figure 4: Sample bull's eye data set.

not correlate well with the clustering. A learner needs to output high and approximately equal weights for the two meaningful features and to approximately zero out the noise feature weights in order to produce correct clusterings.

We experimented with $F_{noise} = 1, 2, 4, 16$ noise features. For each value of F_{noise}, 10 independent runs were performed. In each run, a training set was created, the parameters θ^* were learned on it, then the parameters were evaluated on an independent test set of size $n_{test} = 300$. To save time, the value α^* was selected only once, on the first training set of the 10.

Wine. The second set of experiments was performed on the Wine dataset from the UCI ML repository Aeberhard (1991). In this data set there are 178 different wine samples which can be divided in to 3 classes. The 13 attributes measured on each wine were chemical properties of the wine such as the alcohol content, the total phenols as well as heuristics like color intensity. Pairwise features are defined as the absolute difference between the individual attributes.

We ran 25 experiments in which the training set contained 89 randomly selected points (50% of the data) and the test set contained the remaining 89 points. Then, we created 5 noise features, in a way that will be explained below, and repeated the experiment with the augmented data consisting of 13 true features plus 5 noise features.

The noise features were created in the following way. For each of the 25 repetitions of the experiment, first we chose 5 of the 13 meaningful features at random, then we permuted the values of those features so that the values are no longer associated with the correct wine sample. We use these values as the noise features. The SELECT ALPHA algorithm was performed on the entire data set in order to choose the optimal regularization parameter, which was used for all of the experiments.

Dermatology This set of experiments was performed on the Dermatology data set from the UCI ML repository Guvenir (1998). In the original data set there are 366 patients with erythemato-squamous disease, which are divided up into 6 different sub classes. There are 34 attributes measured on each of the patients. Twelve are clinical attributes such as age, family history, itching and scaling; the other attributes are histopathological and characterize the skin based on various symptoms. The pairwise features are defined as the absolute difference between the individual attributes. Eight patients had missing age information and were removed from the data set. In addition the patients with seboreic dermatitis were removed from the data set for the experiments. When these patients are

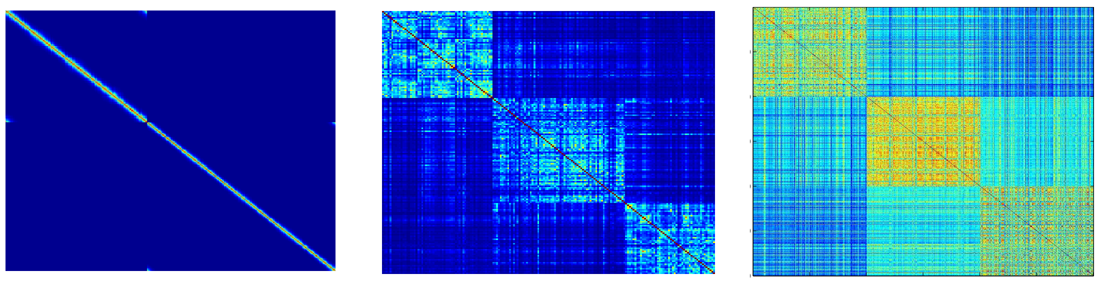


Figure 5: Sample similarity matrices after learning.

included in the data set the clustering results are unstable, the learning algorithm oscillates between weights which give good clusterings and weights which give bad clusterings. Usually the bad clusterings confuse three of the classes, the patients with seboreic dermatitis, lichen planus and pityriasis rosea. The results presented in Table 3 are on 298 of the original patients divided into 5 classes and the learning algorithm learns parameters for all 34 attributes. The regularization parameter was selected in the same manner as in the Wine experiments.

Letter. The last set of experiments was performed on the Letter Recognition data set from the UCI KDD archive Slate (1991). Each letter has sixteen features, $y_{i,1:16}$, integer valued from 0 to 16, associated with it. Examples of the attributes are the height and width of the box and the mean x and y positions of the "on"-pixels. The distance used for creating the $\mathbf{x}$ array is defined here:

$$x_{ij,f} = \begin{cases} 0 & \text{if } y_{i,f} = y_{i,f} = 0 \\ \frac{|y_{i,f} - y_{j,f}|}{y_{i,f} + y_{j,f}} & \text{otherwise} \end{cases} \tag{22}$$

This distance was chosen because it scales the features into [0,1] and because it was believed that a small difference between two large attribute alues was less informative than a small difference between two small attribute values.

The entire data set contains 750 occurrences of each letter in the alphabet. Five small subsets of letters from the alphabet were used here $'SM'$, $'AW'$, $'EI'$, $'ACI'$, $'ACIM'$.

In contrast to the previously described Bull's eye data, this data set is characterized by a Multi-way Normalized cut significantly greater than 0 even in the optimal case, redundant features, and many neighbors for each data point.

7.3 Clustering results

Tables 2, 3, 4 contain the results of using the learned parameters in each experiment to cluster new data. For the Bull's eye, Wine and Dermatology data the clustering results, shown in the columns "CE after", were very good. On the Letters data, the success of learning varies: very good for the groups AW, SM, moderate (i.e significant improvement over no learning) for ACI, ACIM, and non-existent for EI. Tables 2, 3 also show the gap and the eigengap, which can be interpreted as measures of the learning quality and stability. In all cases, these measures are relatively stable to the presence of noise features. This appears most strongly in the Bull's eye data, where the number of noise features F_{noise} grows exponentially, while the gap and eigengap hover at the same order of magnitude.

Table 1: **Summary of training data sets.** F is the number of original features in the data, F_{noise} is the number of additional noise features, n is the size of the training set.

Data	F	F_{noise}	n
Bull's eye	2	1	750
	2	2	750
	2	4	1000
	2	8	1000
	2	16	1000
	2	32	1000
Wine	13	–	89
	13	5	89
Dermatology	34	–	149
Letter			
SM	16	–	100
AW	16	–	200
EI	16	–	200
ACI	16	–	600
ACIM	16	–	800

Table 2: **Bull's eye clustering results** on test data sets. The size of the test sets is $n_{test} = 500$ for all experiments; "before"[learning] refers to using the weights with value θ_0, "after" [learning] refers to using the learned weights θ_{α^*} produced by the learning algorithm. F_{noise}, θ_0, CE, gap, Δ_K denote respectively the number of noise features, number of experiment replications, initial parameters, misclassification error, gap and eigengap of the clusterings obtained on the test sets. A number of $N_{repl} = 15$ replications were performed, and we report means and standard deviations of each value.

			CE		gap_θ		Δ_K	
F_{noise}	θ_0	α^*	before	after	before	after	before	after
1	0.5	1	0.47(0.02)	0(0)	$3.4e^{-2}(8.6e^{-3})$	$8.2e^{-7}(3.0e^{-7})$	$1.1e^{-3}(6.0)e^{-4})$	$1.2e^{-3}(2.5e^{-4})$
2	0.5	1	0.44(0.02)	0(0)	$5.9e^{-3}(5.1e^{-3})$	$3.1e^{-7}(4.7e^{-7})$	$4.8e^{-4}(4.0e^{-4})$	$8.3e^{-4}(3.0e^{-4})$
4	0.25	2	0.44(0.02)	0(0)	$1.6e^{-2}(3.1e^{-2})$	$2.9e^{-5}(1.6e^{-5})$	$6.3e^{-3}(6.4e^{-3})$	$1.5e^{-3}(3.6e^{-3})$
8	0.15	1.25	0.43(0.003)	0(0)	$5.1e^{-2}(4.7e^{-2})$	$3.9e^{-5}(3.0e^{-5})$	$2.9e^{-2}(2.4e^{-2})$	$1.4e^{-4}(5.2e^{-4})$
16	0.10	1.75	0.43(0.015)	0(0)	$1.4e^{-2}(3.5e^{-2})$	$5.7e^{-5}(2.7e^{-5})$	$5.2e^{-2}(5.3e^{-2})$	$9.0e^{-4}(4.3e^{-4})$
32	0.075	1.5	0.43(0.003)	0(0)	$7.1e^{-3}(7.1e^{-3})$	$6.0e^{-5}(1.5e^{-5})$	$3.9e^{-2}(2.8e^{-2})$	$7.1e^{-4}(3.5e^{-4})$

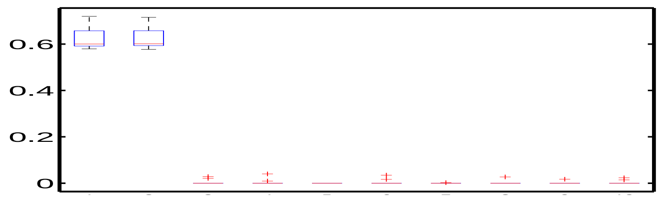
Table 3: **Wine and Dermatology clustering results** on test data sets. The size of the test sets is n_{test}, "before learning" refers to using the weights with value θ_0, "after learning" refers to using the learned weights θ_{α^*} produced by the learning algorithm. F_{noise}, N_{repl}, θ_0, CE, gap, Δ_K denote respectively the number of noise features, number experiment replications, initial parameters, misclassification error, gap and eigengap of the clusterings obtained on the test sets. Where multiple replications were performed, we report means and standard deviations.

					CE		gap_θ		Δ_K	
F_{noise}	n_{test}	N_{repl}	θ_0	α	before	after	before	after	before	after
Wine:0	89	25	1	1.5	0.03(0.02)	0.04(0.03)	0.11(0.01)	0.12(0.03)	0.15(0.05)	0.24(0.07)
5	89	25	1	1.5	0.23(0.16)	0.03(0.01)	0.07(0.02)	0.12(0.02)	0.05(0.02)	0.23(0.06)
Derm:0	149	25	0.75	1.25	0.18(0.08)	0.05(0.07)	0.05(0.02)	0.04(0.03)	0.04(0.02)	0.11(0.03)

Table 4: **Letter clustering results** on test data sets. The size of the test sets is n_{test}, "before" [learning] refers to using the weights with value θ_0, "after" [learning] refers to using the learned weights θ_{α^*} produced by the learning algorithm. Where multiple replications were performed, we report means and standard deviations.

					CE	
Subset	n_{test}	N_{repl}	θ_0	α	before	after
SM	1400	1	0.1	0.06	0.25(0.13)	0.03
	150	25	0.1	0.06		0.06(0.02)
AW	1300	1	0.1	0.8	0.09(0.03)	0.05
	150	25	0.1	0.8		0.06(0.02)
EI	1400	1	0.1	4	0.20(0.05)	0.44
	150	25	0.1	4		0.43(0.07)
ACI	1650	1	0.1	8	0.20(0.10)	0.08
	150	25	0.1	8		0.12(0.05)
ACIM	2200	1	0.1	1	0.34(0.07)	0.14
	150	25	0.1	1		0.15(0.03)

Figure 6: **Bull's eye. Box plots of learned parameters** for 15 experiments when $F_{noise} = 8$ noise features are added. The meaningful features are first. The line connects the medians.



Figures 6, 7, 8, and 9 display the learned weights for the four data sets. It can be seen that in all cases, the noise features were identified as irrelevant and their weights set to 0. In addition, in all three UCI data sets, some other features are found as irrelevant and are assigned zero weight. For the Bull's eye data, the meaningful features are correctly given equal weights. It is interesting to compare the learned weights for the 5 groups of letters in the last experiment. The sizes of the weights differ, with the four letter group having the smallest weights and the two letter groups displaying the largest ones. But the relative sizes of the θ parameters between different features is approximately constant. In particular, features 1, 2, 4, 5, 9, 13, 15 are deemed irrelevant in all learning situations. Note that the parameters for the group EI do not differ markedly from the parameters learned for the other groups. Therefore it is possible that the failure to generalize for this group is only partly because of learning a "bad" set of parameters, and partly because the pair EI is harder to separate. In other words, it is possible that the learning succeeds, but the spectral clustering algorithm fails because the letters are not sufficiently separated.

In this context, we stress that a data set for which classification error is reasonable can be a very difficult data set for clustering, by any method. If two classes overlap, a classifier can still be trained to separate them with near optimal error, but this may not always happen for a clustering algorithm on the same data. In particular for the letters problem: a classifier for letters will be defined in terms of feature values, and examines each data point separately. In contrast, the clustering algorithm only sees *differences* between features, both in the training and in the testing phase, so it has considerably different information. This is why pairwise clustering results should not be expected to be similar to classification results on the same data set.

We also find it remarkable that, even though we allow α to vary over 5 orders of magnitude, in all but one of the many experiments the optimal parameter α^* is close to 1 (the exception is the letter group SM). In addition, the range of good α values is large (1–2 orders of magnitude) in all experiments.

7.4 Other evidence of generalization

Figure 9 shows that the weights learned on different letter groups are similar in (relative) values. But how similar are they from the point of view of clustering performance? To establish this, we used the parameters of the letter group 'SM', denoted here θ^*_{SM} in conjunction with test data for all the other letter groups. We chose the weights θ^*_{SM} because of the learned weight vectors corresponding to the different groups of letters, θ^*_{SM} is the most outlying.

Figure 7: **Wine. Box plots of learned parameters** for 15 experiments without (above) or with (below) $F_{noise} = 5$ noise features added. The meaningful features are first. The line connects the medians.

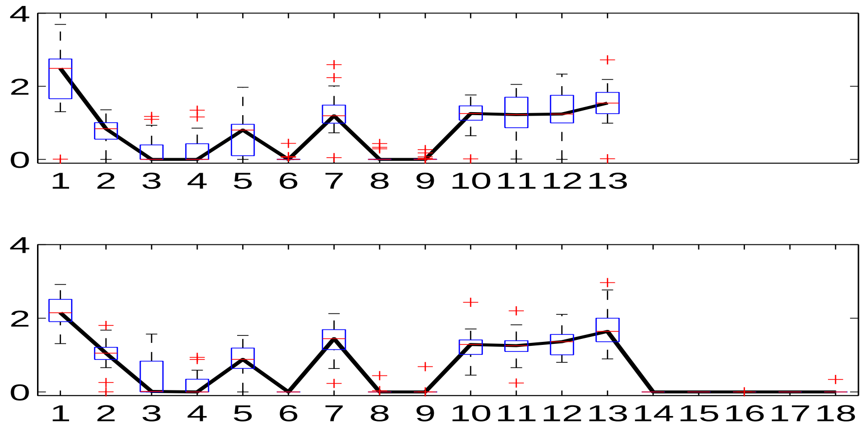


Figure 8: **Dermatology. Box plots of learned parameters.** The line connects the medians.

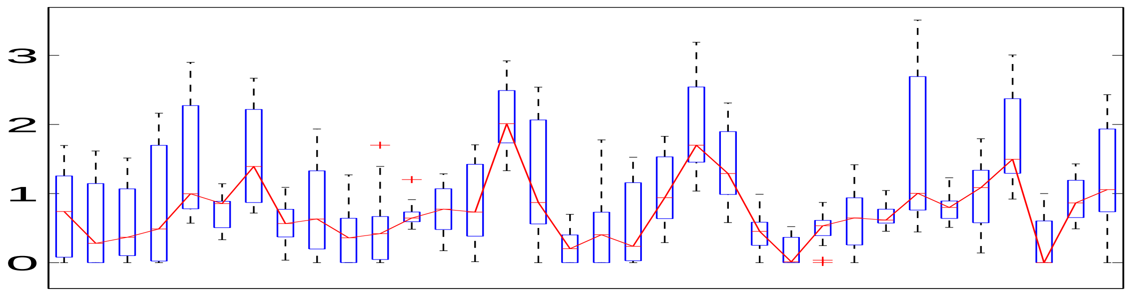


Figure 9: **Letter. Plots of learned parameters** for different groups of letters.

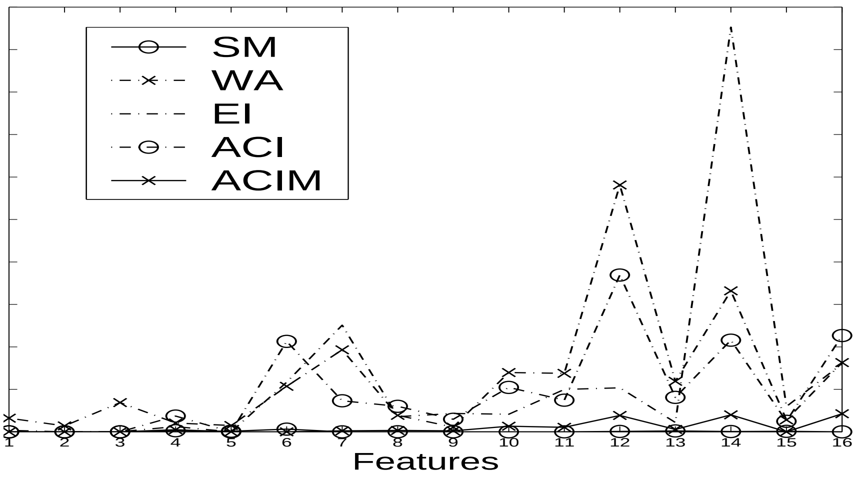


Table 5: **Generalization in the Letter** data. CE after learning is the test set error after learning on the respective letter set copied from table 4; CE with θ^*_{SM} is error on the same test sets, using the parameters learned on the SM training set.

Subset	CE after learning	CE with $\theta *_{SM}$
SM	0.06(0.02)	NA
AW	0.06(0.02)	0.05(0.02)
EI	0.43(0.07)	0.23(0.02)
ACI	0.12(0.05)	0.10(0.03)
ACIM	0.15(0.03)	0.32(0.10)

For each of the groups 'AW', 'EI', 'ACI', 'ACIM' we reclustered the size 150 training sets using the parameters θ^*_{SM} to create new similarity matrices. Table 5 presents the clustering results using these matrices. It can be seen that the θ^*_{SM} parameters result in the same or better clustering performance on 3 out of the 4 letter groups. For the fourth group, ACIM, the performance is unstable, sometimes equal to the performance with θ^*_{ACIM} and sometimes bad.

The second generalization experiment involves the Bull's eye type data. The original Bull's eye distributions consist of two clusters. However, once the correct θ parameters are learned one can use them to cluster data with $K \neq 2$ clusters. This is what we did.

We generated data sets containing $K = 3, 4, 5$ equally spaced concentric rings and addded 4 noisy dimensions. We then clustered them using the weights learned on the 2 rings Bull's eye problem. The results, presented in Table 6 show that the learned weights generalize well. The worse results, for $K = 5$, are because the points in the outer ring become too sparse (we maintained a number of 250 points in each ring).

In fact, the knowledge summarized by the weights learned is that the first two features have relatively equal importance and that the remaining features are irrelevant. With this information the algorithm can generalize well beyond concentric rings to well separated clusters in the plane as long as the distances are of about the same scale as the Bull's eye data.

7.5 Stability Theorem

Here we verify that there exist matrices for which the conditions of Theorem 1 hold. For this, we tested the applicability of Theorem 1 on the all the data sets used in the previously described experiments. On the three UCI data sets the gap/eigengap ratio was too large for Theorem 1 to apply, but on the Bull's eye data we obtained the informative bounds in Table 7.

In addition, we calculated the bounds for the generalization experiments whose results are in Table 6. The last 2 columns of this table show that bounds exist and are informative for values of K larger than 2. Significantly, in the $K = 5$ experiments, clustering fails most of the time. But in the few cases where it succeeds, the bound is able to indicate that.

In Bach and Jordan (2006), Theorem 6 gives an alternative bound for the distance to the optimal clustering, w.r.t the costs considered there. For comparison, in the first experiment we also computed these bounds, and present them in the last row of Table 7.

Table 6: **Bull's eye clustering results for different values of K.** The data had $F_{noise} = 4$ noisy features, and $K = 2, 3, 4, 5$ clusters. The size of the test sets is n_{test}, "before"[learning] refers to using the weights with value $\theta_0 = 0.25$, "after" [learning] refers to using the learned weights θ_{α^*} from Table 2, corresponding to $F_{noise} = 4$. Each experiment was replicated $N_{repl} = 15$ times. The misclassification error is CE and *bound* represents the maximum distance to the optimal clustering given by Theorem 1. This is calculated only when the theorem applies, and $\exists$*bound* denotes how many times this event occurs in the $N_{repl} = 15$ samples. The last column gives the CE restricted to the cases when Theorem 1 applies. Where multiple replications were performed, we report means and standard deviations.

		CE		$\exists$ *bound*			CE after
K	n_{test}	before	after	before	after	*bound* after	if $\exists$ *bound*
2	500	0.44(0.02)	0(0)	2	15	0.07(0.04)	0(0)
3	750	0.66(0.003)	0(0)	0	15	0.008(0.01)	0(0)
4	1000	0.72(0.02)	0.026(0.10)	0	12	0.04(0.03)	0(0)
5	1250	0.74(0.03)	0.22(0.22)	0	4	0.03(0.009)	0(0)

Table 7: **Bull's eye. Bounds from Theorem 1** for the test clusterings used in Table 2; F_{noise} is the number of noise features in the experiment, $\exists$*bound* is the number of clusterings on which Theorem 1 applied, *bound* represents the mean and standard deviation of the bound in the cases the theorem applied. The total number of samples is 15 and the other relevant parameters are given in Table 2. The true value of the error is 0 in all cases.

F_{noise}	1	2	4	8	16	32
$\exists$*bound*	15	15	15	13	12	14
$(gap/\Delta_k) * (\sqrt{k} + 1)^2$	0.004(0.0017)	0.0019(0.0025)	0.18(0.15)	0.20(0.25)	0.41(0.25)	0.51(0.23)
Ortho Projection Bound$*p_{max}$	0.12(0.0072)	0.063(0.034)	0.11(0.03)	0.11(0.024)	0.17(0.078)	0.18(0.075)

Note that in Bach and Jordan (2006) the bounds are given w.r.t another distance. To make them comparable with our bounds, we multiply them by p_{max} the normalized size of the maximum cluster, using the result of Meilă (2009). The resulting bounds are somewhat tighter than ours, and have lower variance.

7.6 Comparisons with other methods

We compared the new learning algorithm with two other spectral learning algorithms: the algorithm of Meilă and Shi (2001b) which we will call here *Target Similarity* and the algorithm of Bach and Jordan (2006) which will be referred to as the *Orthogonal Projection.* Our own method will be designated as *Gap-Eigengap.* We do this by running all algorithms on the Bull's eye, Wine and Dermatology data sets described in section 7.2 and comparing the resulting misclassification errors and weights. The *Target Similarity* algorithm was started with the same intial values as those given in Tables 2 and 3. For the *Orthogonal Projection* we had to chose smaller initial values; these are given in Table 8.

Table 8: **Summary of training data sets.** Initial weight values for the *Orthogonal Projection* experiments.

Data	F_{noise}	θ_0
Bull's eye		
	1	1.0
	2	1.5
	4	3.0
	8	8.5
	16	1.0
	32	12
Wine	0	2.5
	5	5.5
Dermatology	3.5	

Table 9 summarizes the performance of the three learning algorithms on all data sets and Figure 10 displays the learned feature weights for the Bull's Eye data with eight noise features. The results for other numbers of noise features are similar and have not been shown. As we have mentioned, the challenge of the Bull's Eye data set for learning is to identify the two relevant features and to ensure that they have *equal* and significantly higher weights than the noise features.

The regularized Gap-Eigengap algorithms succeeds in doing so. The Target Similarity method works under the implicit assumption that a point's similarity to every other point in its own cluster is approximately the same. This assumption holds for the UCI data sets, but is inappropriate for the Bull's Eye data, where each point is similar to only a small number of other points in its own cluster. As a consequence, the Target Similarity algorithm fails completely on the Bull's Eye data, but works reasonably well on the other data sets. The Orthogonal Projections algorithm works well when there are fewer spurious features but fails when their number increases.

Figure 10: **Bull's eye** weights learned by three different learning algorithms. N_{repl}=15, n_{test} = 1000, F_{noise} = 16

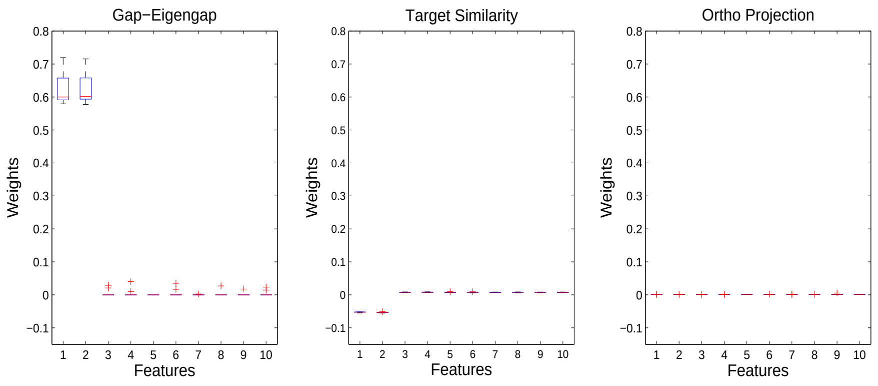
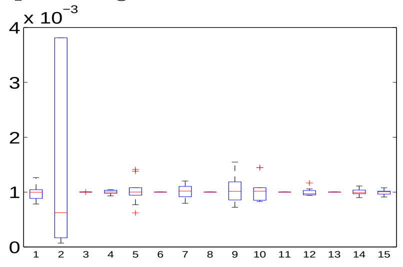


Table 9: Mean and standard deviation of the clustering error for the Gap-Eigengap, Target similarity and Orthogonal Projection algorithms on the Bull's Eye, Wine and Dermatology data sets. The number of training-test set replications for all methods are as reported in Tables 2 and 3

Data	F_{noise}	**Gap-Eigengap**	**Target** S	**Orthogonal projection**
Bull's eye	1	0	0.50(0.002)	0
	2	0	0.50(0.003)	0
	4	0	0.40(0.005)	0
	8	0	0.50(0.003)	0.48(0.013)
	16	0	0.48(0.015)	0.49(0.006)
	32	0	0.49(0.004)	0.49(0.008)
Wine	–	0.030(0.020)	0.046(0.022)	0.49(0.14)
	5	0.040(0.05)	0.045(0.015)	0.51(0.15)
Dermatology	–	0.053(0.071)	0.028(0.016)	0.10(0.19)

Figure 11: **Bull's eye** Close up of weights learned from Orthogonal Projection algorithm



7.7 A Close Look at the Cost Function

This section examines the shape of the cost function being optimized. It is known that this function is non-convex and therefore little can be said about its local minima in general.

We empirically study the shape of $\mathcal{J}$ on the Bull's Eye data, which for the purpose of learning represents a difficult situation will try to assess how harmful is the non-convexity in this problem.

In Figure 12 the values of the cost function, the gap and the eigengap are plotted w.r.t the θ_1, θ_2 weights (of the meaningful features) with θ_3 fixed at 0.1. The figure shows that while the gap keeps decreasing towards 0 as θ_1, θ_2 increase, the minimum value for the cost function occurs at $\theta_1 = \theta_2 = 0.89$. This is due to the regularising effect of the eigengap, which has a local maximum near this point, at $\theta_1 = \theta_2 = 0.73$. This is only a *local minimum*; another local minimum of J is visible very near 0. The latter has a much smaller basin of attraction compared with the "good" local minimum. What this plot shows corresponds to our observations on other data sets: that starting with weights approximatively equal and away from 0 is robust.

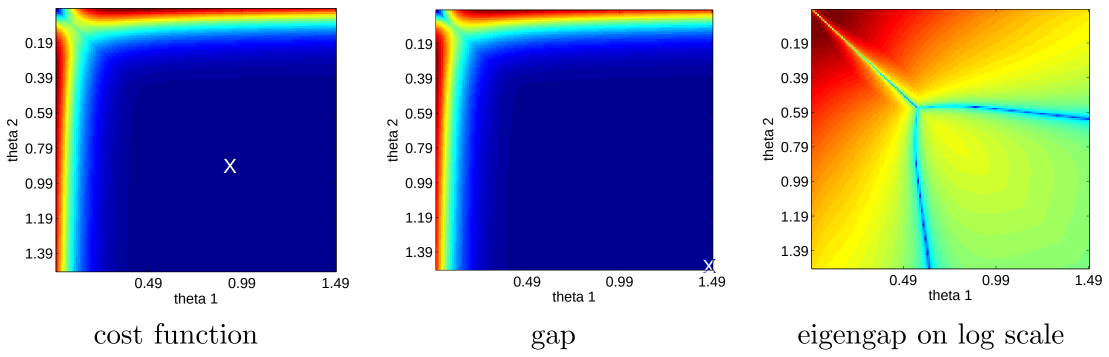


Figure 12: Values of cost function $J_{\alpha=1}$, gap and eigengap for the Bull's Eye data with one noisy feature. The weight θ_3 for this noisy feature is held constant at 0.10, while θ_1, θ_2 the weights of the informative features are varied. Red colors indicate high values; blue indicates zero for the cost and gap plots. Minimum value for cost function and gap marked by white X. The range of values of the eigengap is smaller than for the gap (resulting in the values of J almost equal to the gap). The log scale for the eigengap allows one to see the local maximum at $\theta_1 = \theta_2 = 0.73$.

The largest values for the eigengap occur when both weights are small, but this situation result in a large gap. The other region of large eigengap can be seen as an island in the right most image of Figure 12, centered at weight values for both features of 0.73. It is in this "island" that the optimum of $\mathcal{J}_{\alpha=1}$ falls.

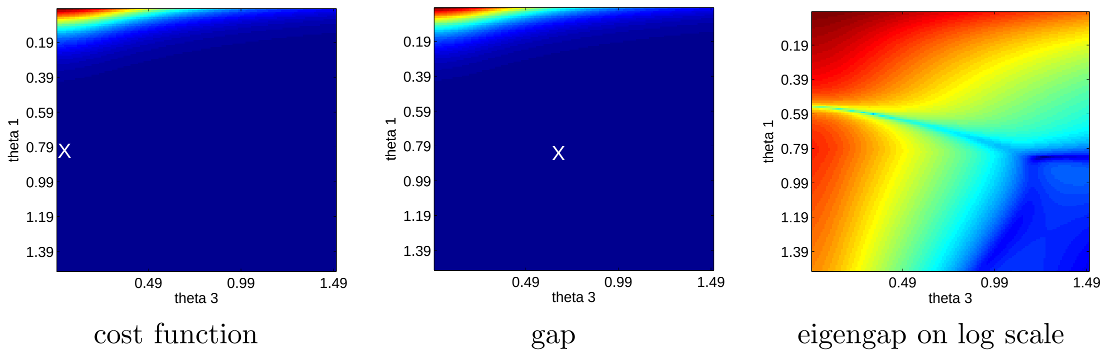


Figure 13: Results of cost function, gap and eigengap on Bull's Eye data with one noisy feature. The weight for the feature assiciated with the vertical axis is kept constant at a near optimal value of 0.8, while the weights for the first feature, which is informative to the clustering, and the third feature which is purely noise vary as indicated by the plot axes. Red colors indicate high values and blue indicates zero. Minimum value for cost function and gap indicated by white X.

Figure 13 presents the value of the cost function and its components, if the weight corresponding to the vertical axis in the Bull's Eye data is held at $\theta_2 = 0.8$, a constant value near the optimal weight value. The weights associated with the first informative feature, the horizontal axics, and the third feature, which is uninformative to the clustering, are varied. The minimum value for the cost function (marked with and X) occurs when the weight associated with the informative feature is at $\theta_1 = 0.81$ and the value for the weight associated with the noisy feature is zero. The minimum for the gap occurs at $\theta_1 = 0.83$, $\theta_2 = 0.63$ for the noisy feature. The plot of the eigengap once agains shows the regularizing behavior of the eigengap. It has high values when both weights are very small, but in addition there is an island with its peak at a weight value of 0.81 for the informative feature and a weight value of 0 for the noisy feature.

This experiment suggests that at least for some nontrivial learning problems, the problem of local minima is mild. This is supported by our experience in running the algorithm, where we found consistently that the results were robust to variations in the initialization (we always initialized with equal θ values) as long as the norm of the initial θ vector was in a reasonable range, sufficiently far away from 0 but not so large as to cause the S to collapse into the unit matrix[3].

8. Stability theorem and relation to clustering stability to perturbations

The Stability Theorem 1 and its corollary can be used in spectral clustering outside of the learning context.

Since the theorem only requires knowledge of the similarity matrix S and of a clustering $\mathcal{C}$, it can be applied to any clustering of any data set (defined by S). Whether the theorem produces a bound or not depends on the clustering $\mathcal{C}$ being a "good" clustering, but it also depends on the input data S. If the best clustering of S does not have an integrality gap small enough, then we cannot obtain an error bound via theorem 1.

There are two ways of using the results of the stability theorem. The practical one is to apply the theorem to the output $\mathcal{C}$ of a spectral clustering algorithm. If the conditions of the theorem are satisfied (i.e $gap(\mathcal{C})/\Delta_K(S)$ sufficiently small) then one obtains a bound on how far the found $\mathcal{C}$ is w.r.t the unknown optimal clustering of S. In the luckier cases, when this bound is very low, below the volume of a single point, one obtains a certificate that the best clustering was found.

A second way is purely theoretical. The theorem also implies that if for some S the optimal (unknown) clustering $\mathcal{C}^*$ is "good" in the sense of satisfying the stability Theorem 1, then for that data set the *MNCut* cost function has a global minimum (represented by $\mathcal{C}^*$) which is much deeper than the other local minima. The rest of this section shows how this interpretation can bridge to another important field of research on clustering algorithms.

8.1 Stability to perturbations and the unique minimizer property

In this paper we have used the term stability to refer to the property of the integrality gap to have a unique deep minimum. In other words, we proved that that small perturbations in the *MNCut* cost around its minimum always imply small perturbations of the clustering around the optimal clustering $\mathcal{C}^*$. In optimization, this is known as *algorithmic stability*[4]. But in the clustering literature, the term "stability" often has another meaning. For instance, in Ben-Hur et al. (2001); Lange et al. (2004); Ben-David et al. (2006) a clustering $\mathcal{C}$ is called stable when it is the output of a clustering algorithm and robust to perturbations (e.g in the

3. For θ very large, the S matrix becomes the unit matrix; for θ near zero, the S_{ij} values become nearly 1. Both cases are singularities that make learning difficult if started from them.
4. In our case the "algorithm" would be the idealized algorithm that picks the minimum of the cost function.

data). The clustering algorithm can be an "ideal algorithm", like for instance picking the optimal clustering w.r.t a cost function. In optimization terms this amounts to algorithmic well conditioning. To distinguish between the two notions of stability, we will call the latter *stability to perturbations*. Note that neither notions of stability refer to a clustering; the former as defined here is a property of the cost function $MNCut(\mathcal{C})$ *on the actual data* and the latter is the property of the same cost function (or of a clustering algorithm) *under perturbations of the data*.

Stability to perturbations has been much discussed as a tool for selecting the number of clusters K, with promising empirical results like Ben-Hur et al. (2001). On the other hand, the theoretical result of Ben-David et al. (2006) (abbreviated to BDLP in the forthcoming) states that stability to perturbations should not inform on the number of clusters. It is not the place to discuss this result and its implications in detail. However, we mention it because our Theorem 1 is strongly related to the technical conditions under which the main result in BDLP is true. We quote the main from BDLP, replacing the cost with the *MNCut* cost, and then explain the concepts involved:

> Theorem 10 of BDLP (Stability theorem) If P has unique minimizer $\mathcal{C}^*$, then any *MNCut*-minimizing clustering algorithm which is risk converging is stable on P.

In the above P is a continuous measure, and the *MNCut* is computed from a given similarity function and P. For more details on the similarity and *MNCut* under a continuous measure see von Luxburg et al. (2005). The terms "risk converging" and "unique minimizer" are defined as

> Definition 8 of BDLP (Risk convergence). Let A be an *MNCut*-minimizing clustering algorithm. We say that algorithm A is risk converging, if the *MNCut* of A's output on a sample of size m converges in measure to the optimal *MNCut* under the distribution P when the sample size tends to infinity.
>
> Definition 9 of BDLP. (Unique minimizer) Let d be a clustering distance. We say that a probability distribution P has unique minimizer $\mathcal{C}^*$ if $(\forall \eta > 0)(\exists \epsilon > 0)(MNCut(P, \mathcal{C}) < \min MNCut(P) + \epsilon) \Rightarrow d_P(\mathcal{C}, \mathcal{C}^*) < \eta$

The property of risk convergence is standard in machine learning, and has been proved for the *MNCut* cost function by von Luxburg et al. (2005). The unique minimizer property however, to the best of our knowledge, has not been proved before for any clustering cost function.

Moreover, the reader will notice immediately that the notion of unique minimizer is identical to the (algorithmic) stability concept studied in this paper, with one difference: we look at stability on a finite sample, while BDLP's Theorem 10 considers the property at the population level. Hence, our Stability Theorem 1 is a proof *in the finite sample case* that the *MNCut* cost has a unique minimizer. To complete the proof of the preconditions of BDLP's theorem, one would have to prove our stability result in the infinite sample case. We conjecture that this is possible, for instance under the conditions in von Luxburg et al. (2005). We mention that the property of unique minimizer for the k-means cost function, was proved in the finite sample case by Meilă (2006); von Luxburg (2008) confirms that the result can be extended to the infinite sample case. The stability results of Bach and

Jordan (2006) also belong to this category, proving the unique minimizer property for the cost functions defined in their paper.

Our stability theorem holds only under certain conditions, in particular when the data are well clustered and the correct number of clusters is known. While it is not an if and only if result, it hints at the possibility that a unique minimizer for the *MNCut* may not exist in a sizable number of cases[5].

9. Discussion and conclusions

In this paper we have made two main contributions. We proved that the *MNCut* cost function has a unique deep minimum in the favorable case when a "good clustering" exists in the data. Then, based on this property, we introduced a new criterion for learning the similarity in spectral clustering.

The criterion optimizes the quality of the target clustering, while constraining the parameters θ as little as possible in the process. This is achieved by choosing the gap as the clustering quality, and by adding the eigengap (squared) as regularization term. One of the difficulties in learning in spectral clustering is the numerical optimization of the chosen criteria, as it often depends on θ through functions of the eigenvalues and vectors of a matrix. The gradients of such functions are expensive to compute and often unstable.

Our choice of objective function performs well in this respect. First, although the clustering is done via eigenvectors, the cost function for learning only depends on the eigenvalues, which by itself removes some potential problems[6] and it does not require that the matrix be positive semidefinite at all times as Bach and Jordan (2006). Second, the eigengap term in the cost function has the effect of enhancing the numerical stability of the problem. This is partly why the cost can be optimized (locally) by a rather unsophisticated gradient descent algorithm.

The amount of regularization can be selected either by standard cross-validation or automatically on the training set alone. Of course, some obvious variations and extension of our learning algorithm are possible, like using multiple training sets, examining the gap-eigengap ratio on the test set or other permissible tunings on the test data or on an independent validation set. We have avoided these here, as our focus was to validate the power of the regularization using training data alone. The experimental results show that the algorithm clusters both sparse and blocky data and eliminates the noisy features flawlessly.

Selecting the number of clusters K is another important problem in spectral clustering. In the supervised learning paradigigm we propose, the problem of selecting K is not an issue because K is known as a byproduct of the labeling. We have also shown that once the similarity function is learned, one can use it to cluster data sets with numbers of clusters different from the training set. This means that finding the correct number of clusters remains a concern for a spectral clustering algorithm, whether it uses our method to learn the weights or not. But learning the weights in supervised mode circumvents the model selection problem.

5. We know already that isolated counterexamples exist, like that in figure 2.
6. Gradients w.r.t. eigenvectors are more difficult numerically and theoretically, see Cour et al. (2005)

Acknowledgments

The authors gratefully acknowledge Jim Burke for his advice on difficult optimizations. Liang Xu proved Lemma 8. Francis Bach provided the code of the Orthogonal Projection algorithm. This work was partially supported by NSF VIGRE grant DMS-9810726, NSF ITR grant 0313339 and the University of Washington RRF grant 2684.

Proof of Theorem 1

We begin by introducing another distance

$$d^2_{\chi^2}(\mathcal{C},\mathcal{C}') \;=\; \frac{K+K'}{2} - \sum_{C_k\in\mathcal{C}}\sum_{C'_k\in\mathcal{C}'}\frac{(VolC_k\cap C'_{k'})^2}{VolC_kVolC'_{k'}} \tag{23}$$

This (squared) distance ranges in [0,K-1]. Note that $K-1-d_{\chi^2}(\mathcal{C},\mathcal{C}')$ is Pearson's χ^2 function, Lancaster (1969) known in statistics as a measure of departure from independence. The distance d_{χ^2} is equivalent with a criterion proposed by Hubert and Arabie (1985), with the modification that each data point is weighted by its volume D_i. This distance was also used in Bach and Jordan (2006).

We now state and prove a slightly stronger results *: theorem 4 proved in the following, and theorem 5 proved in Meilă (2009). From the two theorems the desired result follows immediately. Note also that a more complicated but slightly tighter bound for theorem 5 can be found in Meilă (2009).*

Theorem 4 *Let $\mathcal{C},\mathcal{C}'$ be two K-way clusterings with gap($\mathcal{C}$), gap($\mathcal{C}'$) $\leq \varepsilon < \Delta_K$. Then, $d(\mathcal{C},\mathcal{C}') < \frac{\varepsilon}{\Delta_K}(\sqrt{K}+1)^2 = \epsilon'$.*

Theorem 5 *Meilă (2009) For two clusterings $\mathcal{C},\mathcal{C}'$ with $|\mathcal{C}| = |\mathcal{C}'| = K$ denote $p_{min} = \min_k VolC_k/VolV$, $p_{max} = \max_k VolC_k/VolV$. Then, if $d_{\chi^2}(\mathcal{C},\mathcal{C}') \leq \varepsilon \leq p_{min}/p_{max}$ then $d(\mathcal{C},\mathcal{C}') \leq \varepsilon p_{max}$.*

Proof of theorem 4
For any clustering $\mathcal{C}$ and fixed S matrix, with L defined as in section 5 we denote by e^k the indicator vector of cluster $C_k \in \mathcal{C}$, $y^k = D^{1/2}e^k$, $Y = [y^1 \ldots y^K]$, $\mu_k = 1-\lambda_k(L)$, $U =$ the orthonormal matrix formed with the eigenvectors of L as columns. Thus, $I - L = U\text{diag}(\mu_1,\ldots\mu_n)U^T$. It is easy to show that $MNCut(\mathcal{C}) = \sum_{k=1}^K y^{kT}(I-L)y^k$ and $gap(\mathcal{C}) = MNCut(\mathcal{C}) - \sum_{k=1}^K \mu_k$. For two clusterings $\mathcal{C}$, $\mathcal{C}'$ we express their respective Y, Y' in the basis defined by U as $Y = UA$, $Y' = UA'$ with A, A' being $n\times K$ matrices of coefficients. Let

$$A \;=\; \begin{bmatrix}\tilde{A}\\ E\end{bmatrix} \quad A' \;=\; \begin{bmatrix}\tilde{A}'\\ E'\end{bmatrix} \tag{24}$$

with $\tilde{A},\tilde{A}'$ $K\times K$ matrices.

From here, a series of three stages, stated as lemmas (with proofs immediately following), leads us to the desired result.

Lemma 6 *If gap($\mathcal{C}$) $< \varepsilon$, then $||E||_F^2 < \delta$, where $\delta = \varepsilon/\Delta_K$ and $||E||_F^2$ represents the Frobenius norm.*

Proof Denote by $A_{\cdot k}$ the k-th column of A.

$$\sum_{k=1}^{K} y^{kT}(I-L)y^k = \sum_{k=1}^{K} A_{\cdot k}^T U^T (I-L) U A_{\cdot k} \tag{25}$$

$$= \sum_{k=1}^{K}\sum_{j=1}^{n} A_{jk}^2 \mu_j \tag{26}$$

$$\geq \sum_{k=1}^{K}\sum_{j=1}^{K} A_{jk}^2 \mu_j + \mu_{K+1} \underbrace{\sum_{k=1}^{K}\sum_{j=K+1}^{n} A_{jk}^2}_{||E||_F^2} \tag{27}$$

Now, using the hypothesis, we have

$$\sum_{k=1}^{K}\sum_{j=1}^{K} A_{jk}^2 \mu_j + \mu_{K+1}||E||_F^2 \leq \sum_{k=1}^{K} \mu_k + \varepsilon$$

$$\mu_{K+1}||E||_F^2 \leq \sum_{k=1}^{K} \mu_k (1 - \sum_{j=1}^{K} A_{kj}^2) + \varepsilon \tag{28}$$

$$\leq \mu_K \left(K - \sum_{k=1}^{K}\sum_{j=1}^{K} A_{kj}^2 \right) + \varepsilon \tag{29}$$

$$= \mu_K ||E||_F^2 + \varepsilon \tag{30}$$

■

Lemma 7 *Assume that the conditions of theorem 1 hold and let $Y, A, \tilde{A}, E$ be as defined before. Let the SVD of $\tilde{A}$ be given by*

$$\tilde{A}^T \tilde{A} = \tilde{V}_1^T \mathrm{diag}\{\sigma_1^2, \sigma_2^2, \ldots \sigma_K^2\} \tilde{V}_1 \tag{31}$$

with $\tilde{V}_1$ a unitary matrix. Then $\sigma_k^2 > 1 - \delta$ for $k = 1, \ldots K$.

Proof The columns of A are orthonormal. Therefore

$$A^T A = I = \tilde{A}^T \tilde{A} + E^T E$$

or

$$\tilde{A}^T \tilde{A} = I - E^T E$$

Let $e_k,\ k = 1, \ldots K$ be the singular values of E. Then, there is a unitary matrix $\tilde{V}_2$ such that

$$\tilde{V}_2^T \tilde{A}^T \tilde{A} \tilde{V}_2 = I - \mathrm{diag}\{e_1^2, \ \ldots \ e_K^2\} \tag{32}$$

Since $||E||_F^2 < \delta$ we have that $\sum_{k=1}^{K} e_k^2 < \delta$ and therefore $e_k^2 < \delta$. From (32) we also have that $\sigma_k^2 = 1 - e_k^2$ and $\tilde{V}_2 = \tilde{V}_1$ which implies $\sigma_k^2 > 1 - \delta$ for all $k = 1, \ldots K$. ■

Note also that if $||E||_F^2, ||E'||_F^2 \leq \delta$, then by the Cauchy-Schwartz inequality $||E^T E'|| \leq \delta$.

Lemma 8 *Assume that the conditions of theorem 1 hold and let* Y, A, $\tilde{A}$, E, Y', A', $\tilde{A}'$, E' *and* δ *be as defined before. Then*

$$||Y^TY'||_F^2 \;\geq\; K - (\sqrt{K}+1)^2\delta \tag{33}$$

Proof (by Liang Xu) Let

$$\begin{aligned} ||Y^TY'||_F &= ||A^TA'||_F && (34)\\ &= ||\tilde{A}^T\tilde{A}' + E^TE'||_F && (35)\\ &\geq \Big|\,||\tilde{A}^T\tilde{A}'||_F - ||E^TE'||_F\Big| && (36)\end{aligned}$$

Let us look at the first term of the difference above.

$$||\tilde{A}^T\tilde{A}'||_F^2 = \sum_{i=1}^K\sum_{k=1}^K\left(\sum_{j=1}^K A_{ji}A'_{jk}\right)^2 = \sum_{k=1}^K ||\tilde{A}^TA'_{\cdot k}||_2^2 \tag{37}$$

By virtue of the singular value decomposition in (32) $\tilde{A}$ can be written as

$$\tilde{A} \;=\; \tilde{V}_3\mathrm{diag}\{\sigma_1,\sigma_2,\ldots\;\sigma_K\}\tilde{V}_4 \tag{38}$$

with $\tilde{V}_3, \tilde{V}_4$ complex unitary matrices. Therefore

$$\begin{aligned} ||\tilde{A}^T\tilde{A}'_{\cdot k}||_2^2 &= ||\tilde{V}_4^T\mathrm{diag}\{\sigma_1,\sigma_2,\ldots\;\sigma_K\}\tilde{V}_3^T\tilde{A}'_{\cdot k}||_2^2 && (39)\\ &= ||\mathrm{diag}\{\sigma_1,\sigma_2,\ldots\;\sigma_K\}\tilde{V}_3^T\tilde{A}'_{\cdot k}||_2^2 && (40)\\ &\geq (1-\delta)||\tilde{V}_3^T\tilde{A}'_{\cdot k}||_2^2 && (41)\\ &= (1-\delta)||\tilde{A}'_{\cdot k}||_2^2 && (42)\end{aligned}$$

Then, using equation (37) and lemma 6 we obtain

$$\begin{aligned} ||\tilde{A}^T\tilde{A}'||_F^2 &\geq (1-\delta)\sum_{k=1}^K||\tilde{A}'_{\cdot k}||_2^2 && (43)\\ &\geq (1-\delta)(K-\delta) && (44)\end{aligned}$$

Using now equation (36) above we obtain

$$\begin{aligned} ||Y^TY'||_F^2 &\geq \Big(||\tilde{A}^T\tilde{A}'||_F - ||E^TE'||_F\Big)^2 && (45)\\ &\geq (\sqrt{(1-\delta)(K-\delta)}-\delta)^2 && (46)\\ &\geq K-(\sqrt{K}+1)^2\delta && (47)\end{aligned}$$

■

As $d_{\chi^2} = K - ||Y^TY'||_F^2$ the proof of the theorem is finished. ■

References

Stefan Aeberhard. UCI repository of machine learning databases, July 1991. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Arik Azran and Zoubin Ghahramani. Spectral methods for automatic multiscale data clustering. In *Computer Vision and Pattern Recognition*, pages 190–197. IEEE Computer Society, 2006.

Francis Bach and Michael I. Jordan. Learning spectral clustering with applications to speech separation. *Journal of Machine Learning Research*, 7:1963–2001, 2006.

Shai Ben-David, Ulrike von Luxburg, and David Pal. A sober look at clustering stability. In *19th Annual Conference on Learning Theory, COLT 2006*. Springer, 2006.

A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001. URL `citeseer.ist.psu.edu/hur01support.html`.

Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, Cambridge, MA, 2 edition, 1999.

Jim Burke, Adrian Lewis, and Michael Overton. A robust gradient sampling algorithm for nonsmooth, non-convex optimization. *Society for Industrial and Applied Mathematics Journal of Optimization*, 15(3):751–779, 2005.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 15, 2003.

F. H. Clarke. *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York, 1983.

Timothe Cour, Nicolas Gogin, and Jianbo Shi. Learning spectral graph segmentation. In Robert Cowell and Zoubin Ghahramani, editors, *Artificial Intelligence and Statistics Workshop (AISTATS05)*, 2005.

H. Altay Guvenir. UCI repository of machine learning databases, January 1998. URL `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. Semi-supervised graph clustering: A kernel approach. In *22nd International Conference on Machine Learning*, 2005. URL `http://www.cs.utexas.edu/users/ml/papers/kernel-kdd-05.pdf`.

H.O Lancaster. *The Chi-Squared Distribution*. Wiley, 1969.

Tilman Lange, Volker Roth, Mikio L. Braun, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Comput.*, 16(6):1299–1323, 2004. ISSN 0899-7667. doi: http://dx.doi.org/10.1162/089976604773717621.

Chun-hung Li and Zhi-Li Wu. Spectral energy minimization for semi-supervised learning. Technical Report COMP-03-019, Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong, 2003.

Jan R. Magnus and Hienz Neudecker. *Matrix Differential Calculus*. John Wiley & Sons, 1999.

Marina Meilă. Comparing clusterings – an axiomatic view. In Stefan Wrobel and Luc De Raedt, editors, *Proceedings of the International Machine Learning Conference (ICML)*. ACM Press, 2005.

Marina Meilă. Local equivalence of distances between clusterings. Technical Report 553, UW Statistics, 2009.

Marina Meilă. The uniqueness of a good optimum for K-means. In Andrew Moore and William Cohen, editors, *Proceedings of the International Machine Learning Conference (ICML)*, pages 625–632. International Machine Learning Society, 2006.

Marina Meilă. The multicut lemma. Technical Report 417, University of Washington, 2002. URL `www.stat.washington.edu/reports`.

Marina Meilă and Jianbo Shi. A random walks view of spectral segmentation. In T. Jaakkola and T. Richardson, editors, *Artificial Intelligence and Statistics AISTATS*, 2001a.

Marina Meilă and Jianbo Shi. Learning segmentation by random walks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 873–879, Cambridge, MA, 2001b. MIT Press.

J.R. Norris. *Markov Chains.* Cambridge University Press, 1997.

Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization. Algorithms and complexity.* Dover Publication, Inc., Minneola, NY, 1998.

Susan Shortreed. *Learning in Spectral Clustering.* PhD thesis, University of Washington, 2006.

David Slate. UCI repository of machine learning databases, January 1991. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Gilbert W. Stewart and Ji-guang Sun. *Matrix perturbation theory.* Academic Press, San Diego, CA, 1990.

Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in Neural Information Processing Systems*, volume 14, 2001. URL `citeseer.ist.psu.edu/szummer02partially.html`.

Deepak Verma and Marina Meilă. A comparison of spectral clustering algorithms. TR 03-05-01, University of Washington, May 2003.

Ulrike von Luxburg. personal communication, 2008. Marina's Kmeans theorem can be extended to continuous measures.

Ulrike von Luxburg, Olivier Bousquet, and Mikhail Belkin. Limits of spectral clustering. In Lawrence K. Saul, Yair Weiss, and Leon Bottou, editors, *Advances in Neural Information Processing Systems*, number 17. MIT Press, 2005.

E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In S. Becker et al., editor, *Advances in Neural Information Processing Systems (NIPS)*, number 16, pages 521–528. MIT Press, 2002.

Stella X. Yu and Jianbo Shi. Multiclass spectral clustering. In *International Conference on Computer Vision*, 2003.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Machine Learning. Proceedings of the Nineteenth International Conference*, 2002.