
Regularized spectral learning

UW-Stat Dept TR # 465

Marina Meilă
Department of Statistics
University of Washington
Seattle, WA 98195

Susan Shortreed
Department of Statistics
University of Washington
Seattle, WA 98195

Liang Xu
Department of Mathematics
University of Washington
Seattle, WA 98195

Abstract

Spectral clustering is a technique for finding groups in data consisting of similarities S_{ij} between pairs of points. We approach the problem of learning the similarity as a function of other observed features, in order to optimize spectral clustering results on future data. This paper formulates a new objective for learning in spectral clustering, that balances a clustering accuracy term, the *gap*, and a stability term, the *eigengap* with the later in the role of a regularizer. We derive an algorithm to optimize this objective, and semiautomatic methods to chose the optimal regularization. Preliminary experiments confirm the validity of the approach.

1 Introduction

While there has been much progress in obtaining better spectral clusterings with similarities given or constructed by hand, the problem of automatically estimating the similarities from data has received less attention. This limits the application of spectral clustering to the ability of the domain experts to guess the correct features and their optimal combination for each problem. It makes the results of the clustering algorithm sensitive to the particular function chosen. Moreover, it goes against the grain of many successful approaches in machine learning, which is to consider a large number of possibly irrelevant features, within a regularized setting that lets the data select the few relevant ones.

In contrast to previous work [Meilă and Shi, 2001a, Bach and Jordan, 2004], where the focus was on defining a quality criterion to be optimized w.r.t the parameters on training data, here we take an approach closer to the principles above. We define an objective that balances a term for clustering accuracy on training data with a stability term which acts as a regularizer. While this setting is common to many problems, the specific form of the two terms is particular to spectral clustering.

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, 6 present respectively a gradient algorithm for optimizing the criterion and a method for selecting the amount of regularization. Experimental results are in section 7 and 8 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

$$D_i \equiv \text{Vol} \{i\} = \sum_{j \in V} S_{ij} \quad (1)$$

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 $\text{Vol} A = \sum_{i \in A} D_i$. W.l.o.g we assume that no node has volume 0.

The 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.

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

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

$$\pi_i = D_i/\text{Vol} V$$

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

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

$$MNCut(\mathcal{C}) = \sum_{k=1}^K \sum_{k' \neq k} \frac{Cut(C_k, C_{k'})}{Vol C_k} \quad (3)$$

where

$$Cut(A, B) = \sum_{i \in A} \sum_{j \in B} S_{ij} \quad (4)$$

The definition of *MNCut* is best motivated by the Markov random walk view. Define $P_{AB} = Pr[A \rightarrow B|A]$ 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 .

$$P_{AB} = \frac{\sum_{i \in A, j \in B} \pi_i P_{ij}}{\pi_A} = \frac{\sum_{i \in A, j \in B} S_{ij}}{Vol A} = \frac{Cut(A, B)}{Vol A} \quad (5)$$

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

$$MNCut(\mathcal{C}) = \sum_{k=1}^K \sum_{k' \neq k} P_{C_k C_{k'}} = K - \sum_{k=1}^K P_{C_k C_k} \quad (6)$$

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

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

$$MNCut(\mathcal{C}) \geq K - \sum_{k=1}^K \lambda_k(P) \quad (7)$$

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

$$gap_P(\mathcal{C}) = MNCut(\mathcal{C}) - K + \sum_{k=1}^K \lambda_k(P) \quad (8)$$

One can show [Meilă, 2002] that the gap is 0 iff P has *piecewise constant eigenvectors (PCE)* v^1, \dots, v^K w.r.t \mathcal{C} , that is $v_i^k = v_j^k$ for all $k \leq K$ whenever i, j are in the same cluster.

3 The learning problem

We assume that we have a data set of size n , for which the correct clustering \mathcal{C}^* is given. For each pair of data points i, j in the data set we also measure a set of features. The F -dimensional vector of features is denoted by

$$x_{ij} = [x_{ij,1} \ x_{ij,2} \ \dots \ x_{ij,F}]^T \quad (9)$$

The features are *symmetric*, that is $x_{ij} = x_{ji}$ for all i, j . We also assume for now that the features are non-negative $x_{ij,f} \geq 0$, $f = 1, \dots, F$ and that an increase in $x_{ij,f}$ represents a decrease in the similarity between i and 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. Our formulation however is significantly more general, in that it accomodates dissimilarity features that do not come from a Euclidian space representation of the data.

The *similarity* is an (almost everywhere) differentiable function $S(x; \theta)$ that maps a feature vector $x \in R^F$ into a non-negative scalar similarity (e.g $S(x; \theta) = e^{-\theta^T x}$) with $\theta \in R^F$ a vector of parameters. Let

$$S_{ij} = S(x_{ij}; \theta) \text{ for } i, j = 1, \dots, n \quad (10)$$

$$\mathbf{x} = [x_{ij}]_{i,j=1,\dots,n} \in R^{n \times n \times F} \quad (11)$$

$$S(\theta) \equiv S(\mathbf{x}; \theta) \equiv [S_{ij}]_{i,j=1,\dots,n} \in R^{n \times n} \quad (12)$$

Here, the letter S denotes both the similarity function $S(x; \theta)$, 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 want to learn the optimal parameters θ of the similarity function from a training set including one or more data sets together with their correct clusterings. For simplicity, we assume that we have one data set described by the features \mathbf{x} together with its correct clustering \mathcal{C}^* having K clusters. The generalization to more than one data set is immediate.

This problem was proposed in [Meilă and Shi, 2001a]; 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 infinity of S matrices that are perfect for that clustering. Imposing a target S^* of any form will overconstrain the problem and is equivalent to introducing a bias, which may or may not fit the problem at hand.

In [Bach and Jordan, 2004] an angle between subspaces is used as a criterion for learning the parameters of spectral clustering, thus dispensing with the target S^* . This angle is minimized when $P(\theta)$ has PCE for the given clustering. Optimizing this criterion is difficult in practice due to the need to differentiate a function of the eigenvectors of a matrix w.r.t the matrix elements.

Here, we address learning by explicitly enforcing that the learned parameters induce a good clustering on the training data $(\mathbf{x}, \mathcal{C}^*)$ while “extracting as little information from the training data as possible” as will be shown in the next section.

4 The objective

Quality of the target clustering In the context of spectral clustering, we can satisfy the first of the above requirements by enforcing the quality of the true clustering \mathcal{C}^* w.r.t $S(\theta)$. The quality of a clustering can be measured by either its *MNCut* or its gap. For a given matrix S , a clustering that minimizes the first also minimizes the second, so 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 gap does not carry such an implication. Hence, the 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.

The eigengap and the stability of the optimal clustering To enforce the second requirement, 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. The following result, proved in the appendix, gives a direct relationship between spectral clustering and the eigengap.

For two clusterings with $|\mathcal{C}| = |\mathcal{C}'| = K$ we define the χ^2 -based distance of \mathcal{C} and \mathcal{C}' by

$$d(\mathcal{C}, \mathcal{C}') = 1 - \frac{1}{K} \sum_{C_k \in \mathcal{C}} \sum_{C'_k \in \mathcal{C}'} \frac{(\text{Vol} C_k \cap C'_{k'})^2}{\text{Vol} C_k \text{Vol} C'_{k'}} \quad (13)$$

This distance is symmetric, ranges in $[0,1]$, but is not a metric. Note that $K - 1 - Kd(\mathcal{C}, \mathcal{C}')$ is Pearson's χ^2 function, [Lancaster, 1969] known in statistics as a measure of departure from independence. The distance d 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 . The unweighted d distance was also used in [Bach and Jordan, 2004].

Theorem 1 *Let $\mathcal{C}, \mathcal{C}'$ be two K -way clusterings with $\text{gap}(\mathcal{C}), \text{gap}(\mathcal{C}') \leq \varepsilon < \Delta_K$. Then, $d(\mathcal{C}, \mathcal{C}') < \frac{3\varepsilon}{\Delta_K} = \delta$.*

Corollary 2 *Let \mathcal{C} be a K -way clustering with $\text{gap}(\mathcal{C}) \leq \varepsilon < \Delta_K$ and $\mathcal{C}^* = \underset{|\mathcal{C}'|=K}{\text{argmin}} \text{MNCut}(\mathcal{C}')$. Then $d(\mathcal{C}, \mathcal{C}^*) < \delta$.*

In words, the above theorem and its corollary show that, if we find a clustering with a small enough 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 gap, then there is essential a unique way of obtaining a good partition in that P . Any two partitions with a small gap have to be close to each other.

The learning criterion Now we define learning in spec-

tral clustering as solving the following optimization problem

$$(\mathcal{P}) \quad \max \Delta_K^2(P(\theta)) \quad (14)$$

$$\text{s.t. } \text{gap}_\theta(\mathcal{C}^*) \leq \varepsilon \quad (15)$$

where gap_θ is a short form for $\text{gap}_{P(\theta)}$. By simultaneously achieving a small 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}^*); this has been known to predict good generalization performance in other learning settings.

Our formulation is reminiscent of the SVM formulation; we maximize a “stability” penalty, while ensuring that the training data are well clustered. The parameter ε controls the trade-off between the two goals; ε being a gap, its value is bounded in $[0, K]$. Thus, as a rule of thumb for the choice of ε , a value in $[10^{-2 \dots -1}, 1]$ should represent a good enough quality for \mathcal{C}^* . In section 6 we give a method for semi-automatically selecting its value.

One can also consider other formulations that balance the gap and eigengap. For example, according to theorem 1, a natural optimality criterion would be $J' = \text{gap}_\theta(\mathcal{C})/\Delta_K(P(\theta))$. 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.

5 Optimizing the parameters

Here we present the solution to problem (\mathcal{P}) . Unlike the SVM formulation, in our optimization problem neither objective nor constraints are convex. Therefore, we optimize the parameters by following the gradient, starting from small, positive, random values for the parameters θ .

The constrained optimization problem (14-15) is equivalent with minimizing

$$J_\alpha = \alpha \text{gap}_\theta(\mathcal{C}^*) - \Delta_K^2(P(\theta)) \quad (16)$$

for an α that depends on ε . If ε is known, Lagrange multiplier methods typically find α and θ simultaneously [Bertsekas, 1999]. Here however we choose to take a different approach, that will be described in the next section. For now, we will consider optimizing for θ with a fixed α .

To evaluate the gradient of J_α we write the criterion as the sum of the *MNCut* and a function of the eigenvalues of P . We further recall that [Meilă and Shi, 2001b] the eigenvalues of $P(\theta)$ are equal to the eigenvalues of the symmetric matrix $L(\theta) = D(\theta)^{-1/2}S(\theta)D(\theta)^{-1/2}$. Therefore, we drop the θ in S , L , and P to simplify notation

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

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 $\frac{\partial L_{ij}}{\partial \theta}$ also 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 that for a symmetric matrix L , the derivative of a simple eigenvalue λ w.r.t the matrix elements has the expression

$$\frac{\partial \lambda}{\partial L} = vv^T \quad (17)$$

where v is the eigenvector corresponding to λ . If λ is a multiple eigenvalue, then $\lambda(L)$ is not differentiable at L . In practice, when two of the eigenvalues of $L(\theta)$ are too close, the evaluation of the gradient becomes numerically unstable. When optimizing (16), 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 employ the robust method of [Burke et al., 2003]. 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 (16). Enforcing one large eigengap, Δ_K , for this problem, has the effect of enlarging the distances between all of $\lambda_1, \dots, \lambda_{K+1}$. In fact, in our experiments, the simple gradient given by formula (17) is stable for all but the extremely large values of α .

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 $\frac{\partial \text{MNCut}}{\partial S}$ 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 $K' = \max(2K, K+10)$. 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 by a factor of 2. We allow up to 30 reductions, but the 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. A typical method of speeding up the eigenvalue computation in spectral clustering is to use sparse similarity matrices instead of full ones or the Nyström method for approximating eigenvectors. [Fowlkes et al., 2004] These tricks can be also applied to learning spectral clustering, leading to additional time savings.

6 Selecting the regularization parameter

Choosing the amount of regularization is critical for the success of learning. By translating α into an ϵ via the optimization problem (\mathcal{P}), one gets a handle on the order of magnitude of ϵ which can be used when there is good prior knowledge about the problem or when a cheap solution is needed. Now we show a simple method for semi-automatically selecting α .

Algorithm SELECTALPHA

1. Choose a set A of α values spanning a reasonable range, e.g $[10^{-2}, 10^2]$
2. For $\alpha \in A$
 - (a) Find $\hat{\theta}_\alpha = \operatorname{argmin}_\theta J_\alpha(\theta)$
 - (b) Compute $\delta_\alpha = \Delta_K(P(\hat{\theta}_\alpha))$, $g_\alpha = \text{gap}_{\hat{\theta}_\alpha}(\mathcal{C}^*)$
3. (Manual) Choose α^* so that g_{α^*} is small, but δ_{α^*} is still reasonably large. In a plot of g_α vs δ_α , the desired α^* will be near the lower right corner of the plot.

Figure 1 shows an example of such a plot. Note that more than one α value may be near the knee of the curve. This algorithm could be refined, for example by a binary search on α , but the hope is that the problem is not so sensitive to the value of the regularizing parameter.

7 Experiments

In this section we provide the details and results of experiments run with the learning algorithm presented here.

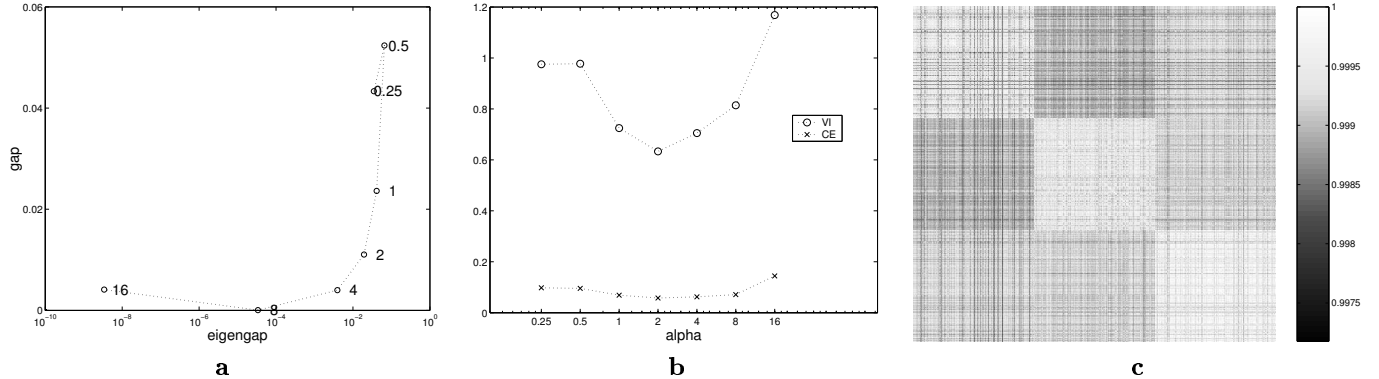


Figure 1: Selecting α on the “letters A,C,I” data set described in section 7: (a) The gap vs. eigengap plot on the training data; each point is labeled with its α value. (b) Clustering quality for the learned θ 's on an independent test set, measured by the variation of information (VI) and the classification error (CE), plotted versus the regularization parameter α . (c) The learned similarity matrix S for this dataset. Note that the off-diagonal blocks are non-zero.

The similarity is defined as $S(x, \theta) = e^{-\theta^T x}$, with θ assumed positive. The initial weights were chosen to be inversely proportional to the variance of the $x_{ij,l}$. After learning for each prespecified α two optimal α values and corresponding parameters were chosen. The first optimal α was chosen based on the smallest misclassification error on the training set, with ties going to the smallest α , its corresponding parameters will be denoted $\theta_{\alpha,CE}$. The second α was chosen using the Select Alpha algorithm described in Section 6, and its parameters will be denoted $\theta_{\alpha,SA}$. Selecting a single α is less clear in this second method; in Figure 1a it can be seen that both 2 and 4 are possible candidates for an optimal α . In these situations, it appears that reasonable candidates for α lead to similar clustering results. In order to test each vector of parameters the first K eigenvectors of the corresponding $P(\theta)$ matrix were clustered using k-means, with multiple random and orthonormal initializations; for more information on initializations see [Verma and Meilă, 2003].

7.1 Bull’s-Eye

The first set of experiments was run on simulated data, a bull’s-eye in two dimensions. 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, large within cluster distances, a small number of neighbors for each point, and a high possibility of over-fitting make learning non-trivial. Also, the meaningful features each taken separately do not correlate well with the clustering, making the high weighting of both meaningful features important in the learning process. We added $Ndim$ noisy dimensions to the bull’s-eye, which were symmetric random matrices designed to have the same mean as the meaningful features. The weights were learned for a vector of α 's, by minimizing (16). The distance metric used for creating the two meaningful dimensions was $x_{ij,l} = |y_{il} - y_{jl}|$. Where y_{i1} is the x-coordinate and y_{i2} is the y-coordinate associated with

the i^{th} point. We tested the weights on ten independent samples of 300 data points.

Table 1: Results for the bull’s-eye experiments.

n_t is the size of the training data set the parameters were learned on and $Ndim$ is the number of noisy dimensions added to the data. The α presented here are those chosen based on the lowest training CE. The classification error (CE), gap_θ , Δ_k and Ncut are averaged over 10 independent test sets.

n_t	$Ndim$	α_{CE}	CE	gap_θ	Δ_k	Ncut
150	1	1.2	0	4.4e-5	1.1e-3	4.9e-3
200	2	1.2	0	9.3e-5	3.0e-4	7.0e-3
400	4	1.2	0	7.9e-5	5.4e-4	6.7e-3
700	16	2	0	1.1e-4	4.0e-4	7.2e-3

Table 1 presents the average of the classification error, gap, eigengap and normalized cut over the ten samples using $\theta_{\alpha,CE}$ as the weights applied to the features. In this situation the $\theta_{\alpha,SA}$ are identical to the $\theta_{\alpha,CE}$. In all cases the learned parameters were positive and approximately equal for the meaningful features and 0 on the noisy dimensions, which lead to great clusterings on the test data samples.

7.2 Letters

The second set of experiments is 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. Some 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 x array is defined here:

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

This distance was chosen because it scales down the features into $[0,1]$ and because it was believed that a small

difference between two large attribute values was less informative than a small difference between two small attribute values. This data set complements the bull’s eye data set because the Multi-way Normalized cut is greater than 0, the clusters are dense and there are redundant features.

Due to time and memory limitations small subsets of the letters were explored. Between the training and test sets each letter appears approximately 750 times. The training set of the ‘SM’ data consisted of 50 occurrences of each letter, the ‘WA’ and ‘EI’ training data contained 100 of each letter, and the training set for the ‘ACI’ and ‘ACIM’ data had 200 of each letter. We also chose to re-sample sets of 150 letters, 25 times from each of the test data subsets, to test the parameters on smaller data sets, because it was noted that on the artificial data that larger data sets provided for better clusterings even with the learned parameters applied.

Table 2 gives the results of the learning experiments for both $\theta_{\alpha,CE}$ and $\theta_{\alpha,SA}$ on the training and 3 gives the results for the test data. Table 4 presents the results for the re-sampled test data while 5 presents the clustering results if unlearned parameters are used. The optimal α ’s chosen by the two methods differ, which is due in part to the fact that it is rare, in either method, to have a unique optimal α and ties were decided differently between the methods.

For the subsets ‘SM’ and ‘WA’ a fairly good clustering can be obtained with equal weights applied to the features. Some of the learned parameters equal zero which suggests that there are redundant features. The clustering error on the training data for the ‘ACI’ and ‘ACIM’ data sets is reduced by over 50% by using the learned parameters, while the clustering error for the ‘EI’ subset is reduced to less than one-third. This suggests that in addition to possible redundant features there could be noisy features which do not provide positive information about the clustering. From Table 2 we can see that the gap_{θ} is not always smaller than the Δ_k as is required for the assumptions of Theorem 1 to hold, yet the clusterings still perform well. More information about the bound and distance from the true clustering is found in Section 7.3

A priori it might be thought that the α chosen from the CE on the training set might over fit the training data. Table 3 reports that data does not provide much evidence for this, in all but one case the parameters chosen with this method have lower classification errors than the second set of learned parameters. For all but the ‘EI’ data set the learned parameters reduce the classification error by over 50% from the unlearned parameters. This maybe because the learning algorithm does not minimize the clustering error directly. Once again we notice that in many cases the gap_{θ} is larger than the Δ_k .

We seem to get better results with one larger test set than

we do with the smaller re-sampled test sets, we attribute this to the stability of the eigenvectors when n is large. The mean of the 25 data sets with the learned parameters applied are presented in Table 4 and the results before learning are given in Table 5. The clustering error for the re-sampled test data is reduce in a pattern similar to when the large test set is used and the learned parameters are applied. The standard deviation for the samples is reported in parenthesis next to the mean. While the mean of the clustering is relatively low when using the learned parameters, there appears to be quite a bit of variability between the 25 data sets.

In addition to looking at the performance of the learned parameters compared to the unlearned parameters, it is interesting to ask whether there is a common set of important features across letters. Figure 2 plots the parameters, $\theta_{\alpha,CE}$, assigned to each feature for the five different subsets tested. The $\theta_{\alpha,SA}$ are very similar and are not shown. While there is variation in the magnitude of the parameters, there seem to be some agreement in the weights selected as important. It appears that features twelve and fourteen are relatively important for all of the data sets, while one, two, five and nine do not appear to be very important for clustering these subsets. The parameters learned from the letters ‘S’ and ‘M’ appear to be the most different from the others. To further test this idea of a common set of features, the $\theta_{\alpha,SA}$ learned from the ‘SM’ subset, were applied to the other subsets and the clustering errors for the multiple disjoint test sets are reported in Table 6.

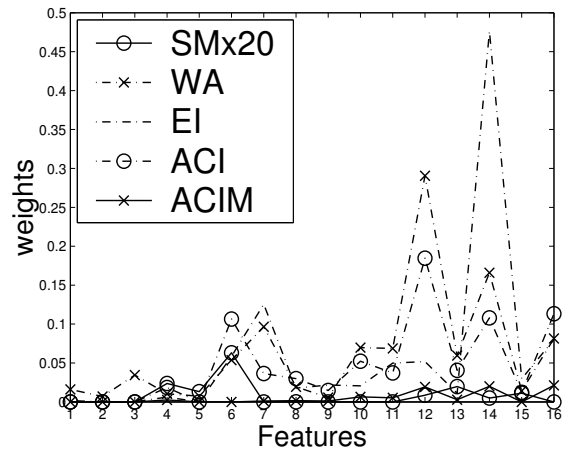


Figure 2: Comparison of learned parameters on various data sets for the letters

7.3 Stability Theorem

All of the experiments previously discussed chose clusterings by minimizing the gap, while maintaining a large eigengap. This set of experiments are designed to show that the clusterings achieved by optimizing this criterion are close to the optimal clustering. We performed

Table 2: **Training Data** The results of the learning process on the training data in the Letters experiments. The last three columns are the pre-learning results, showing that the learned parameters decrease the clustering error substantially.

	$\theta_{\alpha,CE}$				$\theta_{\alpha,SA}$				$\theta_{1:16} = 0.1$		
	α	CE	gap_{θ}	Δ_k	α	CE	gap_{θ}	Δ_k	CE	gap_{θ}	Δ_k
SM	4	0.0	3.67e-5	8.34e-5	6	11.0	7.60e-3	1.47e-2	4.0	8.87e-1	2.07e-1
WA	10	2.0	6.87e-11	3.71e-10	0.8	4.0	9.10e-3	5.13e-2	5.0	1.73e-2	4.90e-2
ACI	4	7.2	4.00e-3	3.92e-3	8	8.2	5.46e-5	3.38e-5	16.8	4.94e-2	1.33e-2
AICM	8	13.4	1.23e-3	6.60e-5	1	16.3	1.16e-4	6.78e-5	34.4	6.67e-2	7.79e-3
EI	2	8.5	3.72e-3	3.43e-3	4	8.5	1.13e-3	5.96e-4	33.5	4.68e-2	3.42e-2

Table 3: **Test Data** The results of learning on the test data. The last three columns are the pre-learning results, revealing that in the test data the learned parameters also decrease the clustering error

	$\theta_{\alpha,CE}$				$\theta_{\alpha,SA}$				$\theta_{1:16} = 0.1$		
	α	CE	gap_{θ}	Δ_k	α	CE	gap_{θ}	Δ_k	CE	gap_{θ}	Δ_k
SM	2	2.8	3.60e-5	1.07e-4	6	1.0	7.48e-6	2.84e-5	28.5	8.58e-1	2.26e-1
WA	10	3.2	1.54e-9	5.78e-9	0.8	5.3	1.04e-2	4.88e-2	8.5	2.05e-2	4.05e-2
ACI	4	7.1	4.00e-3	4.41e-3	8	7.9	5.43e-5	3.65e-5	15.5	4.87e-2	1.67e-2
AICM	8	12.1	1.19e-3	7.12e-5	1	14.0	1.04e-4	5.99e-5	33.2	6.67e-2	1.47e-2
EI	2	17.9	5.63e-3	4.8e-3	4	44	3.36e-3	2.50e-3	15.4	4.91e-2	3.67e-2

Table 6: **Generalization of parameters**

	$\theta_{SM,CE}$		
	CE	gap_{θ}	Δ_k
WA	4.8 (1.9)	9.0e-3 (1.8e-3)	3.9e-2 (3.4e-3)
ACI	9.7 (2.9)	1.9e-2 (3.0e-3)	2.0e-2 (3.1e-3)
AICM	32.3 (9.6)	6.9e-2 (8.5e-3)	1.4e-2 (4.0e-3)
EI	19.3 (10.7)	6.2e-3 (1.1e-3)	4.4e-3 (2.0e-3)

a set of experiments using an artificially constructed S matrix with $n=100$ and $K=5$, with clusters of sizes 10, 20, 30, 20 and 20. This matrix is not block diagonal, the node volumes are unequal ($\max D_i / \min D_i = 15$) and unevenly distributed (smallest D_i is in the smallest cluster). This matrix is perfect for the clustering K , resulting in a block-stochastic matrix P . Symmetric i.i.d. noise was added to the S matrix such that $S_{ij} = S_{ij} + noise_{ij} \sim \epsilon/n \times \text{uniform}[0, 1)$ and the corresponding P matrix was constructed. Figure 3 plots the value of the bound in Theorem 1 averaged over 10 noise realizations as a function of this noise, for the clustering obtained by the Meilă-Shi spectral algorithm [Meilă and Shi, 2001b]; $K = 5$ for this data set. As one can see, the bound is informative up to significant noise levels (SNR of about 1). Since the node volumes are known, in the best cases the bound can represent a proof that the optimal clustering for this data set has been actually found.

Table 7 reports the bound obtained by Theorem 1 and the true χ^2 -based distance, defined in (13). The true distance is usually much lower than the bound, especially, in the cases when gap_{θ} is larger than Δ_k and the bound is no longer meaningful. As can be seen in Table 7 for these experiments the bound is usually meaningless, i.e. greater

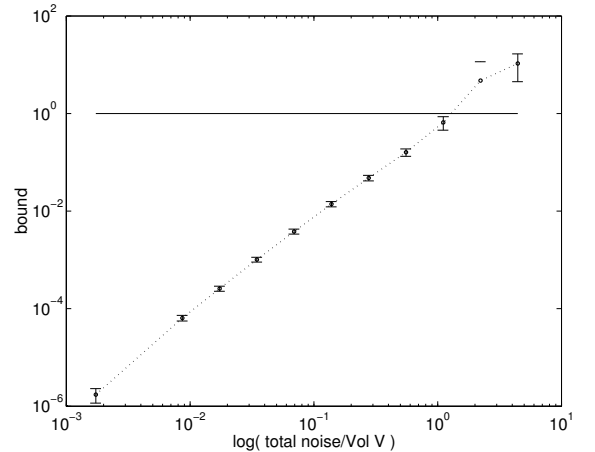


Figure 3: Bound value in Theorem 1 averaged over 10 noise realizations with bars representing plus and minus one standard deviation. The solid (blue) line is the the noise divided by average degree versus the bound.

than 1, but the true distance is very low. This suggests that while having a small gap_{θ} in relation to the Δ_k is a sufficient condition for a close to optimal clustering it is not necessary.

8 Discussion and conclusions

We have 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,

Table 4: **Re-sampling Data** Reports the results of learning on the re-sampled smaller test data sets. The mean of 25 data sets of 150 letter is reported with the standard deviation in parenthesis.

	$\theta_{\alpha, CE}$				$\theta_{\alpha, SA}$			
	α	CE	gap_{θ}	Δ_k	α	CE	gap_{θ}	Δ_k
SM	2	6.8 (4.8)	4.8e-5 (1.3e-5)	1.1e-4 (2.7e-5)	6	5.7 (2.5)	5.9e-3 (8.0e-4)	1.7e-2 (1.8e-3)
WA	10	4.6 (1.3)	7.6e-11 (1.3e-11)	3.4e-10 (3.1e-11)	0.8	5.8 (1.8)	1.1-2 (2.1-3)	4.9e-2 (3.1e-3)
ACI	4	8.1 (2.1)	4.2e-3 (6.2e-4)	4.2e-3 (7.5e-4)	8	12.0 (5.3)	6.1e-5 (9.8e-6)	3.5e-5 (8.3e-6)
AICM	8	17.3 (6.9)	1.3e-3 (1.9e-4)	2.3e-4 (1.2e-4)	1	15.2 (3.0)	1.1e-4 (1.2e-5)	4.8e-5 (1.2e-5)
EI	2	15.4 (9.2)	5.9e-3 (1.2e-3)	4.8e-3 (1.1e-3)	4	43.3 (7.1)	3.5e-3 (7.9e-4)	2.4e-3 (8.3e-4)

Table 5: **Re-sampling Data Cont'd** The clustering error results before learning. The mean of 25 data sets of 150 letter is reported with the standard deviation in parenthesis.

	$\theta_{1:16} = 0.1$		
	CE	gap_{θ}	Δ_k
SM	25.2 (12.8)	1.5e-2 (4.7e-3)	1.2e-2 (7.0e-3)
WA	9.3 (2.7)	2.1e-2 (3.9e-3)	4.0e-2 (9.5e-3)
ACI	20.5 (9.8)	5.0e-2 (8.8e-3)	1.8e-2 (6.8e-3)
AICM	34.1 (7.2)	7.1e-2 (8.2e-3)	1.4e-2 (4.4e-3)
EI	19.7 (5.4)	5.1e-2 (6.5e-3)	3.7e-2 (4.8e-3)

Table 7: **Bound Info for the letters** The bound from Theorem 1 and the distance defined in (13) of the clustering found with the learned weights from the true clustering are reported for the training and test data sets and both optimal weights.

	Training data				Test data			
	Bound	$\theta_{\alpha, CE}$ $d(C_{true}, C)$	Bound	$\theta_{\alpha, SA}$ $d(C_{true}, C)$	Bound	$\theta_{\alpha, CE}$ $d(C_{true}, C)$	Bound	$\theta_{\alpha, SA}$ $d(C_{true}, C)$
SM	1.32	0	4.55	0.19	1.01	0.05	0.79	0.02
WA	0.46	0.04	0.53	0.04	0.80	0.06	0.64	0.10
ACI	2.42	0.13	4.84	0.15	2.72	0.13	4.50	0.16
AICM	6.76	0.24	5.14	0.28	50.3	0.21	5.21	0.24
EI	3.25	0.14	6.69	0.14	3.51	0.19	4.02	0.49

and by adding the squared eigengap as a regularization term. One of the difficulties of learning in spectral clustering is the numerical optimization of the chosen criteria, as they often depend on θ through functions of the eigenvalues and vectors of P or another matrix. The gradients of such functions are expensive to compute and often unstable. Our choice of objective function also performs well in this respect, in that it can be optimized by a rather unsophisticated gradient descent algorithm. This is partly due to the eigengap term which has the effect of enhancing the numerical stability of the problem.

The amount of regularization is selected (semi-) automatically on the training set alone, with no further adjustments on the test set. Of course, some obvious variations are possible, like using multiple training sets, examining the $\delta_{\alpha}, g_{\alpha}$ graph on the test data, 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 are very promising, as the algorithm clusters both sparse and blocky data, and

eliminates the noisy features flawlessly.

The stability theorem 1 and its corollary can be used outside of spectral learning. For instance, the bound can tell one how far a given clustering is w.r.t the unknown optimal clustering on a data set, and even, in the luckiest cases, prove that the best clustering was found. The theorem makes no explicitly assumptions about the similarity matrix S . However, one should be aware that not every S will have a clustering good enough to satisfy the bound.

We conclude by remarking that from the perspective of learning, this work is just a beginning. A frame perhaps solid enough to allow one to think of the yet unanswered questions at the core of statistical learning, like sample complexity, prior knowledge, generalization bounds and so on. We hope that our future work will contribute to these areas.

Acknowledgments

The authors gratefully acknowledge Jim Burke for his invaluable advice on difficult optimizations. This work was

partially supported by NSF VIGRE grant DMS-9810726, NSF ITR grant 0313339 and the University of Washington RRF grant 2684.

References

- [Bach and Jordan, 2004] Bach, F. and Jordan, M. I. (2004). Learning spectral clustering. In Thrun, S. and Saul, L., editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA. MIT Press.
- [Bertsekas, 1999] Bertsekas, D. P. (1999). *Nonlinear programming*. Athena Scientific, Cambridge, MA, 2 edition.
- [Burke et al., 2003] Burke, J. V., Lewis, A. S., and Overton, M. L. (2003). A robust gradient sampling algorithm for non-smooth, nonconvex optimization. Technical report, Courant Institute of Mathematical Sciences. To appear in SIAM J. Optimization, 2005.
- [Fowlkes et al., 2004] Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225.
- [Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2:193–218.
- [Lancaster, 1969] Lancaster, H. (1969). *The Chi-Squared Distribution*. Wiley.
- [Meilä, 2002] Meilä, M. (2002). The multicut lemma. Technical Report 417, University of Washington.
- [Meilä and Shi, 2001a] Meilä, M. and Shi, J. (2001a). Learning segmentation by random walks. In Leen, T. K., Dietterich, T. G., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13, pages 873–879, Cambridge, MA. MIT Press.
- [Meilä and Shi, 2001b] Meilä, M. and Shi, J. (2001b). A random walks view of spectral segmentation. In Jaakkola, T. and Richardson, T., editors, *Artificial Intelligence and Statistics AISTATS*.
- [Slate, 1991] Slate, D. (1991). UCI repository of machine learning databases.
- [Verma and Meilä, 2003] Verma, D. and Meilä, M. (2003). A comparison of spectral clustering algorithms. TR 03-05-01, University of Washington. (submitted).
- [Yu and Shi, 2003] Yu, S. X. and Shi, J. (2003). Multiclass spectral clustering. In *International Conference on Computer Vision*.

Proof of Theorem 1

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 $C_k \in \mathcal{C}$, $y^k = D^{1/2}e^k$, $Y = [y^1 \dots y^K]$, $\mu_k = 1 - \lambda_k(L)$, $U =$ the orthonormal matrix formed with the eigenvectors of L as columns. Thus, $I - L = U \text{diag}(\mu_1, \dots, \mu_n) U^T$. 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

$$A = \begin{bmatrix} \tilde{A} \\ E \end{bmatrix} \quad A' = \begin{bmatrix} \tilde{A}' \\ E' \end{bmatrix} \quad (18)$$

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

Lemma 3 *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 .

$$\sum_{k=1}^K y^{kT} (I - L) y^k = \sum_{k=1}^K A_{\cdot k}^T U^T (I - L) U A_{\cdot k} \quad (19)$$

$$= \sum_{k=1}^K \sum_{j=1}^n A_{jk}^2 \mu_j \quad (20)$$

$$\geq \sum_{k=1}^K \sum_{j=1}^K A_{jk}^2 \mu_j + \underbrace{\mu_{K+1} \sum_{k=1}^K \sum_{j=K+1}^n A_{jk}^2}_{\|E\|_F^2} \quad (21)$$

Now, using the hypothesis, we have

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

$$\mu_{K+1} \|E\|_F^2 \leq \sum_{k=1}^K \mu_k \left(1 - \sum_{j=1}^K A_{kj}^2\right) + \varepsilon \quad (22)$$

$$\leq \mu_K \left(K - \sum_{k=1}^K \sum_{j=1}^K A_{kj}^2 \right) + \varepsilon \quad (23)$$

$$= \mu_K \|E\|_F^2 + \varepsilon \quad (24)$$

■

Lemma 4 *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*

$$\tilde{A}^T \tilde{A} = \tilde{V}_1^T \text{diag}\{\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2\} \tilde{V}_1 \quad (25)$$

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

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

$$\tilde{V}_2^T \tilde{A}^T \tilde{A} \tilde{V}_2 = I - \text{diag}\{e_1^2, \dots, e_K^2\} \quad (26)$$

Since $\|E\|_F^2 < \delta$ we have that $\sum_{k=1}^K e_k^2 < \delta$ and therefore $e_k^2 < \delta$. From (26) 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 $k = 1, \dots, K$. ■

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

Lemma 5 *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*

$$\|Y^T Y'\|_F^2 \geq K - (\sqrt{K} + 1)^2 \delta \quad (27)$$

Proof Let

$$\|Y^T Y'\|_F = \|A^T A'\|_F \quad (28)$$

$$= \|\tilde{A}^T \tilde{A}' + E^T E'\|_F \quad (29)$$

$$\geq \left| \|\tilde{A}^T \tilde{A}'\|_F - \|E^T E'\|_F \right| \quad (30)$$

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

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

By virtue of the singular value decomposition in (26) \tilde{A} can be written as

$$\tilde{A} = \tilde{V}_3 \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_K\} \tilde{V}_4 \quad (31)$$

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

$$\|\tilde{A}^T \tilde{A}'_{\cdot k}\|_2^2 = \|\tilde{V}_4^T \text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_K\} \tilde{V}_3^T \tilde{A}'_{\cdot k}\|_2^2 \quad (32)$$

$$= \|\text{diag}\{\sigma_1, \sigma_2, \dots, \sigma_K\} \tilde{V}_3^T \tilde{A}'_{\cdot k}\|_2^2 \quad (33)$$

$$\geq (1 - \delta) \|\tilde{V}_3^T \tilde{A}'_{\cdot k}\|_2^2 \quad (34)$$

$$= (1 - \delta) \|\tilde{A}'_{\cdot k}\|_2^2 \quad (35)$$

Then, using equation (??) and lemma 3 we obtain

$$\|\tilde{A}^T \tilde{A}'\|_F^2 \geq (1 - \delta) \sum_{k=1}^K \|\tilde{A}'_{\cdot k}\|_2^2 \quad (36)$$

$$\geq (1 - \delta)(K - \delta) \quad (37)$$

Using now equation (30) above we obtain

$$\|Y^T Y'\|_F^2 \geq \left(\|\tilde{A}^T \tilde{A}'\|_F - \|E^T E'\|_F \right)^2 \quad (38)$$

$$\geq (\sqrt{(1 - \delta)(K - \delta)} - \delta)^2 \quad (39)$$

$$\geq K - (\sqrt{K} + 1)^2 \delta \quad (40)$$

As $d = 1 - \frac{1}{K} \|Y^T Y'\|_F^2$ and $(\sqrt{K} + 1)^2 / K < 3$ the proof of the theorem is finished.