Active Spectral Clustering via Iterative Uncertainty Reduction

Fabian L. Wauthier
Computer Science Division
UC Berkeley
Berkeley, CA-94720
flw@cs.berkeley.edu

Nebojsa Jojic
Microsoft Research
Redmond, WA-98052
jojic@microsoft.com

Michael I. Jordan
Computer Science Division
UC Berkeley
Berkeley, CA-94720
jordan@cs.berkeley.edu

ABSTRACT

Spectral clustering is a widely used method for organizing data that only relies on pairwise similarity measurements. This makes its application to non-vectorial data straightforward in principle, as long as all pairwise similarities are available. However, in recent years, numerous examples have emerged in which the cost of assessing similarities is substantial or prohibitive. We propose an active learning algorithm for spectral clustering that incrementally measures only those similarities that are most likely to remove uncertainty in an intermediate clustering solution. In many applications, similarities are not only costly to compute, but also noisy. We extend our algorithm to maintain running estimates of the true similarities, as well as estimates of their accuracy. Using this information, the algorithm updates only those estimates which are relatively inaccurate and whose update would most likely remove clustering uncertainty. We compare our methods on several datasets, including a realistic example where similarities are expensive and noisy. The results show a significant improvement in performance compared to the alternatives.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms

Keywords

Spectral Clustering, Active Learning

1. INTRODUCTION

Clustering is a fundamental problem involving summarizing, indexing and classifying various types of data. As data sets become larger and more complex, algorithms that depend on pairwise similarities—rather than fixed length feature vector representations—are growing increasingly popular. An important example is spectral clustering, which partitions data through a spectral analysis of the Laplacian matrix induced by the similarity graph.

Although methods based on pairwise similarities are growing in popularity, a practical difficulty is that pairwise similarities can be expensive to acquire, be it due to computational requirements, need for human input, or lack of observability. In protein clustering, for example, a computationally expensive alignment process may be necessary before two proteins can be compared. Other examples in computational biology require a combinatorial search for each pairwise similarity, even when the datapoint can be represented in a compact form (a protein sequence is usually less than 1000 letters long). A counterpart to these computational issues is that often the only practical way to obtain similarities is to query human annotators. Here, measurements are not only expensive, but frequently also noisy. In this paper we consider the task of organizing a stream of snapshots taken by a wearable camera at a rate of about one photo per 20 seconds [12]. Clustering these snapshots is beyond the capabilities of existing computer vision algorithms, so human guidance is necessary to either cluster the images, or learn improved models to do the clustering for us. As the human subject (e.g., an Alzheimer's patient), wears the camera during an open-ended observation period, it is not clear a priori what the clusters should be. A way to tackle this problem is to ask human annotators to rate how similar any two photos are and then to cluster using this data. (For example, photos may be deemed similar if they were taken in similar locations.) This is one of many examples where crowdsourcing, albeit expensive, can be used to collect pairwise similarity measurements. As a final example, in some situations the objects that are being compared can disappear over time, making retrospective comparisons difficult. Consider viral strains, which, if not preserved in a laboratory, may disappear, leaving behind only indirect assessments with other viruses. While similarities among preserved viruses can be acquired at a relatively high cost in the lab (e.g., crossreactivity of the immune responses), similarities involving the extinct strain are not available at any cost. In addition to this time barrier, geographic, legal and policy barriers can also make certain pairwise similarities inaccessible. All these examples illustrate that similarities may be noisy and arbitrarily expensive to compute, to the extreme where certain similarities are completely unavailable.

KDD'12, August 12–16, 2012, Beijing, China.

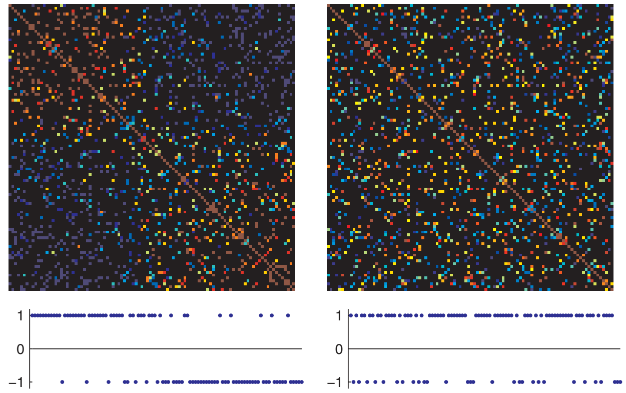


Figure 1: Two incomplete similarity matrices. The left was constructed from median image similarities, measured using Amazon Mechanical Turk (see Section 6.3). We permuted the rows and columns so that any clusters would become visible as diagonal blocks. Then we set 82% of the similarities to 0 (black regions). The right matrix was constructed similarly, but starting from similarities sampled uniformly in $[0, 1]$. The left matrix still shows clustering structure, which is also visible in the sign vector of the second Laplacian eigenvector, plotted below (see Sections 3 and 4 for details). The right matrix had almost no structure to start with, so no structure is visible in the corresponding eigenvector.

The potentially significant cost of obtaining pairwise similarities motivates the search for tradeoffs between desired clustering quality and the required amount of data. In this paper, we study this question in the setting of the spectral clustering of n objects when we do not have access to all pairwise measurements initially, but we can iteratively query the similarities from an external black-box procedure (which may impose restrictions on which of the subset of all $n(n-1)/2$ similarities can be queried). In this active learning formulation the goal is to find a good approximation to the true clustering based on as few similarity evaluations as possible, thus reducing the overall cost (i.e. compute cycles, human tasks performed, laboratory material and other data collection expenses).

A recent contribution in this direction has been made by Shamir and Tishby [18]. Their approach was designed for querying arbitrary similarity matrices, including those where no clustering structure is apparent. This paper significantly improves on their method by exploiting the fact that most realistic pairwise similarity matrices, even if incomplete, do exhibit clustering structure. Consider Figure 1, where on the left we show an incomplete matrix that we might realistically encounter and on the right a matrix constructed from random similarities. Shamir and Tishby assume no structure in the matrix (such as in the matrix on the right) and query measurements that maximally change the overall clustering solution. In contrast, we tailor our active learning algorithm to work well with *realistic* matrices (such as the one on the left) where information about the true clustering emerges quickly and can be leveraged to guide an improved query selection strategy.

An aspect of spectral clustering that is commonly ignored is that similarity measurements are generally noisy. Active spectral clustering methods have so far not taken these uncertainties into account. We extend our algorithm in this direction to allow repeat measurements of noisy similarities, and use those to compute running estimates of the true similarities, as well as estimates of their accuracy. Using this information, we extend our algorithm to measure that similarity which is relatively inaccurate and whose update would most likely remove clustering uncertainty.

The paper is organized as follows. In Section 2 we review related research. Section 3 presents some background on spectral clustering. Next, we describe our active learning algorithm in Section 4. In Section 5 we extend the algorithm to take measurement noise into account. We present experiments on synthetic and real datasets in Section 6 and conclude with final remarks in Section 7.

2. RELATED RESEARCH

Spectral Clustering.

Spectral clustering was popularized in the machine learning community by Shi and Malik [20] and shortly afterwards revisited by Ng, Jordan and Weiss [14]. Since then, it has permeated the literature and become a firm part of the practitioner's toolbox. Over the years, numerous connections to other research fields have been made, a comprehensive review of which can be found in [24]. However, very little work in the spectral clustering literature has explicitly investigated situations when only a subset of all similarities is known, possibly contaminated by noise.

Matrix Completion.

One simple approach to adapting spectral clustering to the case of missing similarities is to exploit the burgeoning literature on matrix completion methods. A particularly straightforward approach is to impute a constant value (e.g., zero) for the missing similarities. Such approaches are often used in practice, and some of its properties have been analyzed theoretically [18].

More sophisticated methods can be deployed under the assumption that entire rows or columns of the similarity matrix can be measured. The driving assumption behind these methods is that the true matrix is of low rank, so that a small subset of elements approximately captures the global matrix structure. Among these approaches, the Nyström method [6, 26] is perhaps the most well known. Other algorithms that use row/column sampling include for instance [4, 5, 9]. Applications of some of these methods to spectral clustering are given in [7, 17]. In less controlled situations, we do not have access to entire rows/columns of measurements, but only an arbitrary subset; in this setting it is still possible to exploit a low-rank assumption for matrix completion [1, 22]. There is debate about the value of these methods in the spectral clustering setting; in particular, Shamir and Tishby [18] argued that the low rank assumptions can be unrealistic in many spectral clustering applications and demonstrated that the Nyström method often performs poorly. Additionally, most of the row/column sampling methods require that the sampling distribution depends on the entire similarity matrix [4, 5, 6, 9]. Because this matrix is unknown, these methods are of limited applicability in our setting. Finally, we highlight that with the exception of Huang et al. [11], relatively little work has been done on analyzing the influence of incomplete or perturbed similarity matrices on the spectral clustering solution.

Active Learning.

Until recently, the study of active learning for spectral clustering was restricted to settings where the entire similarity matrix is known (perhaps approximately) and an external oracle can be repeatedly queried for additional *linkage constraints* between objects (of the form *must-link* or *cannot-link*). When the similarity matrix only approximately captures the desired clustering, adding such constraints iteratively can help resolve ambiguous boundary cases. Relevant examples include [2, 13, 25, 27]. We note in particular the work of Xu et al. [27], in which the constraints are absorbed by modifying the similarity matrix in a way that is akin to measuring similarities of higher quality. The focus in active learning for spectral clustering has only recently shifted to scenarios in which explicit costs are imposed on the measurement of similarities. This focus is exemplified by Shamir and Tishby [18], who propose and analyze an active learning method based on matrix perturbation theory [21].

3. SPECTRAL CLUSTERING

We begin by presenting our notation and summarizing the key ideas of spectral clustering. For further details and various interpretations of spectral clustering we refer the reader to von Luxburg [24]. Given n objects, denote by W the $n \times n$ symmetric matrix of pairwise similarities among these objects. Typically, $0 \leq w_{ij} \leq 1, i, j = 1, \ldots, n$ and $w_{ii} = 1, i = 1, \ldots, n$. Let $D = \operatorname{diag}(W\mathbf{1})$ be the diagonal matrix of row sums of W. The *unnormalized Laplacian matrix* is then given by

$$L = D - W. \tag{1}$$

Spectral clustering partitions the n objects into two groups by thresholding the second eigenvector v_2 of L. Specifically, if we let the partition be encoded by variables $c_i \in \{-1, +1\}, i = 1, \ldots, n$, then

$$c_i = 2\,[v_2(i) > 0] - 1. \tag{2}$$

Here, we use the notation $v_2(i)$ to indicate the i^{th} component of the vector v_2. Because the partitioning is trivial to compute from the second eigenvector, we will occasionally refer to the eigenvector itself as the spectral clustering solution, rather than the partitioning.

Spectral clustering only sees the data as filtered through the matrix W. Thus it is possible to adapt the spectral approach to the clustering of non-vectorial data such as graphs, sequences and sets; it suffices that similarity scores can be computed for these objects. This is generally accomplished via a kernel function, and computationally efficient kernels are available for certain kinds of structured objects [10, 19]. Unfortunately, however, kernel formulations are often too rigid to be adapted to specific needs, and often lack interpretability. As we move to more complex datasets, the notion of similarity a practitioner is interested in may not be captured by a kernel. Indeed, as highlighted in the Introduction, in many practical examples the similarities cannot be evaluated by a computer at all, but must be provided by an experiment or human annotator.

4. ACTIVE LEARNING

In this section we propose an active learning strategy that attempts to alleviate the above issues. Our work is based on a matrix perturbation argument for an intermediate estimate of the Laplacian matrix. Given incomplete measurements, we estimate the true Laplacian matrix as

$$\hat{L} = \hat{D} - \hat{W}, \tag{3}$$

where $\hat{W}$ is the matrix of all pairwise measurements with zero imputed for unknown entries, and $\hat{D} = \operatorname{diag}(\hat{W}\mathbf{1})$. The motivation for imputation with zero can be seen by rewriting the spectral clustering problem. The second eigenvector of the Laplacian $\hat{L}$ can be found as

$$\hat{v}_2 = \operatorname{argmin}_v v^\top \hat{L} v = \operatorname{argmin}_v \sum_{ij} \hat{w}_{ij}(v(i) - v(j))^2 \tag{4}$$

$$\text{s.t.} \quad v^\top v = 1 \tag{5}$$

$$v^\top \mathbf{1} = 0. \tag{6}$$

Thus, similarities act as weights on soft constraints between eigenvector components. By imputing zero for missing similarities we merely ignore those constraints which are not supported by a measurement.

For any set of similarities $\hat{W}$, the second eigenvector $\hat{v}_2$ gives the best guess for an embedding of the objects on the line. The embedding is such that two groups of similar objects are embedded away from zero, on the negative or positive orthant, respectively. Any objects that are approximately equally similar to all remaining objects are embedded near zero. This is intuitive, for these are the objects that cannot clearly be assigned to either of the two groups. Indeed, since a mean can be found by minimizing a mean squared error, we see from Eq. (4) that an object i with approximately constant similarities $\hat{w}_{ij}$ to remaining objects j should be embedded near the average of their embedding locations $\hat{v}_2(j)$. On the other hand, if the data actually clusters well and is reasonably balanced, then we expect the second eigenvector v_2 of the true Laplacian L to have elements with magnitude on the order of $1/\sqrt{n}$, since $v_2^\top v_2 = 1$. In this way, the elements of most realistic embeddings v_2 should be expected to be bounded away from zero. Spectral clustering based on incomplete data $\hat{W}$ partitions the objects by looking at the signs of the embedding $\hat{v}_2$ (Eq. (2), with threshold at zero). Consequently, objects which are embedded near zero are the objects about whose cluster label we should be most "uncertain" about.

In many practical cases, a relatively small amount of data suffices so that $\hat{v}_2$ already indicates a useful clustering. The left sign vector shown in Figure 1 demonstrates this on a real dataset. We use such partial information as a guide towards measurements that more quickly reveal the true nature of the clustering. In this approach, our earlier intuition about the magnitude of $\hat{v}_2$ components plays a crucial role. More specifically, our active learning strategy uses matrix perturbation theory to reveal that entry of $\hat{W}$ for which a constant perturbation would change the *minimum magnitude element* of $\hat{v}_2$ the most. The rationale is that by focussing on small magnitude components, we more quickly move them away from the cluster boundary (i.e. 0), and thus reduce uncertainty in the partial clustering. In effect, we try to choose measurements that help us push the embedding clusters further apart. If the data actually clusters well, this should quickly guide us to the clean clustering we expect to find.

Algorithm 1: IU-RED

```
$S = \{(i,j) : i,j \in \{1,\ldots,n\}, i < j\}$
$\hat{W} = I$
for $t = 1,\ldots,n(n-1)/2$
    $\hat{L} = \mathrm{diag}(\hat{W}\mathbf{1}) - \hat{W} = \sum_{p=1}^{n} \hat{\lambda}_p \hat{v}_p \hat{v}_p^\top$
    $k_{min} = \mathrm{argmin}_k |\hat{v}_2(k)|$    (1)
    $(i^*, j^*) = \mathrm{argmax}_{(i,j)\in S} \left| \frac{d\hat{v}_2(k_{min})}{d\hat{w}_{ij}} \right|$    (2)
        $= \mathrm{argmax}_{(i,j)\in S} \left| \sum_{p>2}^{n} \frac{\hat{v}_2^\top [\partial\hat{L}/\partial\hat{w}_{ij}] \hat{v}_p}{\hat{\lambda}_2 - \hat{\lambda}_p} \hat{v}_p(k_{min}) \right|$    (3)
    $S = S \setminus \{(i^*, j^*)\}$
    $\hat{w}_{i^*j^*} = w_{i^*j^*}, \hat{w}_{j^*i^*} = w_{j^*i^*}$
return Second eigenvector of $\hat{L} = \mathrm{diag}(\hat{W}\mathbf{1}) - \hat{W}$
```

Algorithm 2: S&T [18]

```
$S = \{(i,j) : i,j \in \{1,\ldots,n\}, i < j\}$
$\hat{W} = I$
for $t = 1,\ldots,n(n-1)/2$
    $\hat{L} = \mathrm{diag}(\hat{W}\mathbf{1}) - \hat{W} = \sum_{p=1}^{n} \hat{\lambda}_p \hat{v}_p \hat{v}_p^\top$    (1)
    $(i^*, j^*) = \mathrm{argmax}_{(i,j)\in S} \left\| \frac{d\hat{v}_2}{d\hat{w}_{ij}} \right\|_2^2$    (2)
        $= \mathrm{argmax}_{(i,j)\in S} \left\| \sum_{p>2}^{n} \frac{\hat{v}_2^\top [\partial\hat{L}/\partial\hat{w}_{ij}] \hat{v}_p}{\hat{\lambda}_2 - \hat{\lambda}_p} \hat{v}_p \right\|_2^2$    (3)
    $S = S \setminus \{(i^*, j^*)\}$
    $\hat{w}_{i^*j^*} = w_{i^*j^*}, \hat{w}_{j^*i^*} = w_{j^*i^*}$
return Second eigenvector of $\hat{L} = \mathrm{diag}(\hat{W}\mathbf{1}) - \hat{W}$
```

Suppose we have the Laplacian eigenvector decomposition $\hat{L} = \sum_{p=1}^{n} \hat{\lambda}_p \hat{v}_p \hat{v}_p^\top$ and that $\hat{\lambda}_1 \leq \hat{\lambda}_2 \ldots \leq \hat{\lambda}_n$. For spectral clustering, $\hat{\lambda}_1 = 0$ and $v_1 = \mathbf{1}/\sqrt{n}$. Matrix perturbation theory (e.g. Stewart and Sun [21], Chapter V, Section 2.3) gives the first order change of the second eigenvector as

$$\frac{d\hat{v}_2}{d\hat{w}_{ij}} = \sum_{p>2}^{n} \frac{\hat{v}_2^\top \left[\partial\hat{L}/\partial\hat{w}_{ij}\right] \hat{v}_p}{\hat{\lambda}_2 - \hat{\lambda}_p} \hat{v}_p, \tag{7}$$

provided $\hat{\lambda}_2$ has multiplicity 1. Note that $\partial\hat{L}/\partial\hat{w}_{ij} = (e_i - e_j)(e_i - e_j)^\top$, where e_i is the indicator vector of i. If $k_{min} = \mathrm{argmin}_k |\hat{v}_2(k)|$, the change to the smallest magnitude element of $\hat{v}_2$ is $d\hat{v}_2(k_{min})/d\hat{w}_{ij}$. Our proposed active learning algorithm, IU-RED, is given in Algorithm 1.

A recent algorithm due to Shamir and Tishby [18] has a similar structure. The main steps are shown in Algorithm 2, which we refer to as S&T throughout.[1] The algorithm chooses measurements that maximize the *global* change to $\hat{v}_2$ by maximizing the norm on line 2. The reasoning is the following: As more measurements are acquired, the estimate $\hat{v}_2$ will necessarily converge to the true eigenvector v_2, regardless of the query ordering, since only a finite number of measurements can be made. Since constant perturbations to elements of $\hat{W}$ can have varying effects on $\hat{v}_2$, we should choose to update that element where the effect is largest.

If the similarity matrices were random, we would not expect to see partial clusterings emerge in $\hat{v}_2$, even with fairly large amounts of data. Figure 1 has highlighted this. In this unstructured setting, Shamir and Tishby's method may well be the best we can do, for it targets the global change in the clustering solution. Our algorithm exploits that practical similarity matrices are highly structured even when severely subsampled and uses this partial information as a guide for query selection. Our experiments emphasize that the empirical improvements of our method are significant, even though the algorithmic differences may at first appear minor.

5. MEASUREMENT NOISE

In many settings, only noisy similarities can be measured. In the Section 6, for example, we consider a crowdsourcing application where similarities are manually assigned by human labelers. A significant factor there is that even cooperative workers may disagree on similarity scores. Noisy similarities can be a fundamental problem, yet their impact on spectral clustering has not been thoroughly understood. Huang et al. [11] are among the few to investigate the effects of perturbations when all similarities are known. To our knowledge, active spectral clustering with costly and noisy measurements has not been considered.

The simplest way to deal with noise is to take the mean or median of multiple repeated measurements. However, given m repeated measurements of normally distributed similarities, both the mean and median have a standard deviation that is only a factor of $O(1/\sqrt{m})$ smaller than that of a single measurement. Thus, to halve the standard deviation we need about four times as many measurements. Thus, measuring every similarity multiple times is a fairly expensive way to reduce the effects of noise. It is especially wasteful since it is likely that only similarities of objects close to the cluster boundaries need to be known accurately to resolve the decision boundary. This section gives an active learning algorithm that asks for repeat measurements only when the measurement can significantly change the current solution and when our uncertainty in the true similarity is large.

We have extended IU-RED to maintain a "running median" estimate for each similarity. At any time, the true similarity is estimated as the median of any repeat measurements. Additionally, we maintain for each similarity estimate $\hat{w}_{ij}$ an estimate of its standard deviation, $\hat{\sigma}_{ij}$. When no measurements were made, we let the standard deviation be that of a uniform on [0, 1]; i.e., $\hat{\sigma}_{ij} = \sqrt{1/12} \approx 0.2887$. After a single measurement we set $\hat{\sigma}_{ij} = s$, an estimate of the population standard deviation, and we let $\hat{\sigma}_{ij} = s/\sqrt{m}$ for m repeat measurements.[2] IU-RED was then modified to choose that measurement (possibly a repeat) where the product $\hat{\sigma}_{ij}|d\hat{v}_2(k_{min})/d\hat{w}_{ij}|$ is maximal. This amounts to

[1] The full algorithm requires a "budget" parameter b which specifies the maximum number of measurements that can be requested from an oracle. For this paper, we set $b = n(n-1)/2$. Another version of their algorithm interleaves active selection with random selection, which we consider in our experiments.

[2] Several variations of this theme could be considered. Given enough samples, frequentist confidence intervals of the median could be estimated using the bootstrap. Alternatively, given prior information one could use Bayesian techniques to estimate posterior means and variances.

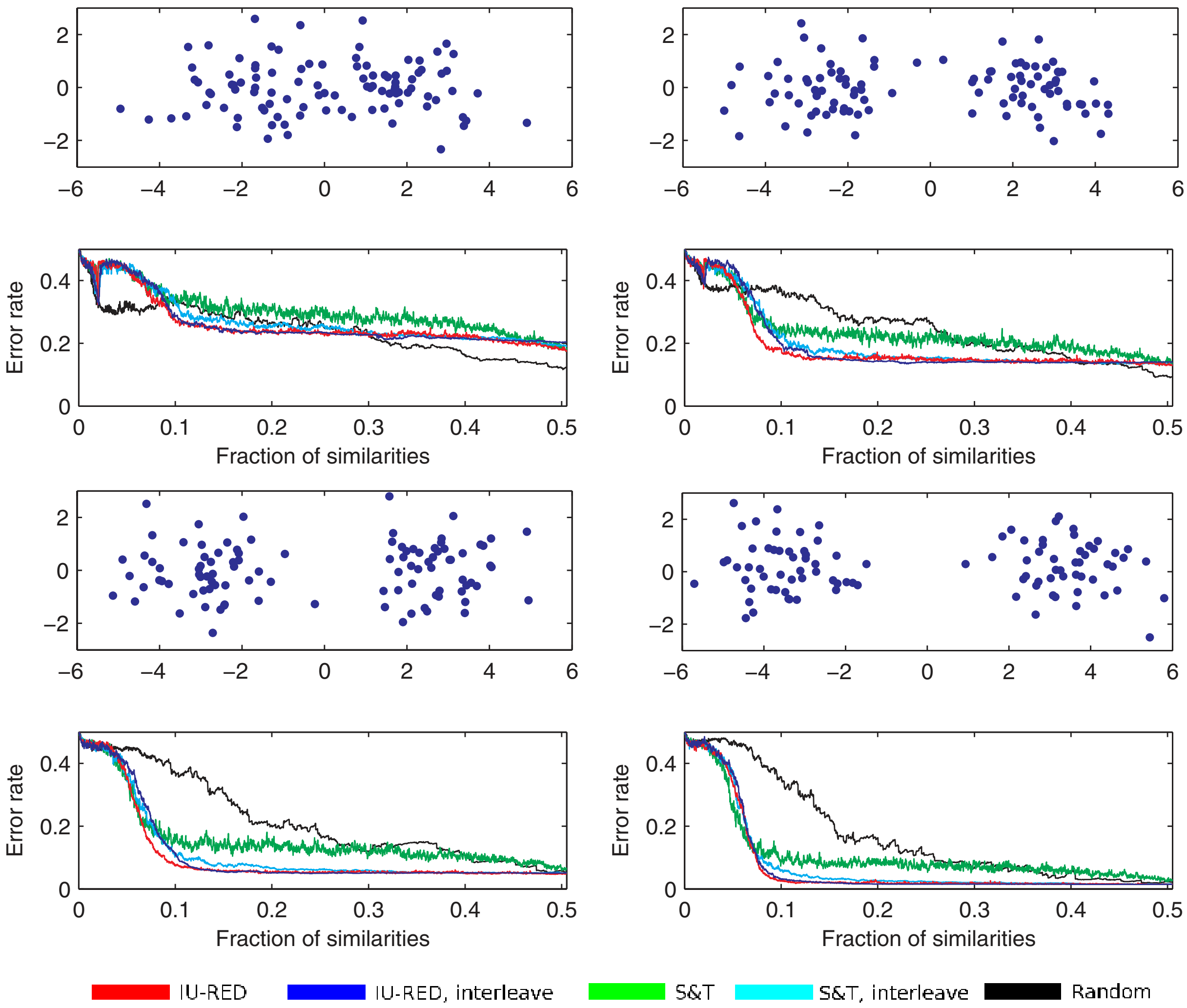


Figure 2: Results on synthetic datasets. For each dataset we sampled a total of 200 points from two Gaussians of increasing separation. Methods are evaluated on the average misclustering error rate, relative to a spectral clustering solution with complete data. Our proposed methods are "IU-RED" and "IU-RED, interleave".

choosing that measurement where our current estimate is uncertain *and* where the uncertainty matters. The S&T algorithm can be similarly modified and is considered in that form during our experiments.

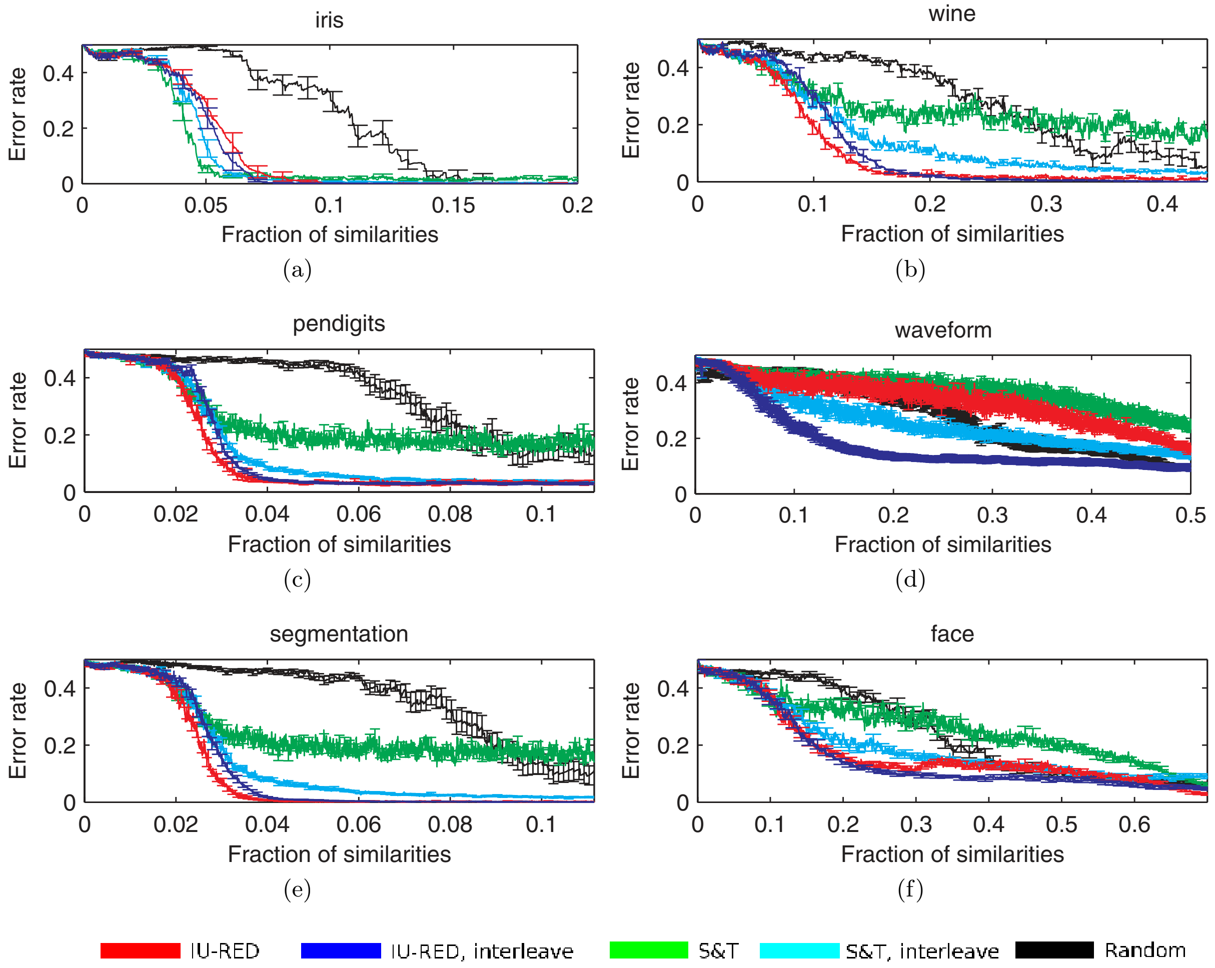
6. RESULTS

We begin this section by evaluating the basic IU-RED algorithm of Section 4 on synthetic and real datasets. Subsection 6.3 then considers the extension to noisy measurements.

Our methods include IU-RED and a version of IU-RED where active selection is interleaved with random selection. We compared against three alternatives: S&T, S&T with interleaved random selection, and random selection only. The last three methods have been previously evaluated in Shamir and Tishby [18] using a similar evaluation methodology and on similar datasets as we consider here. In particular, we consider binary classification, which can be extended to more than two clusters by recursive splitting or by expanding the reasoning outlined here to more eigenvectors. Shamir and Tishby also evaluated an algorithm based on the Nyström method [7], but often found performance to be poor if the low rank assumption was not met. We evaluate all methods on the misclustering error, relative to a spectral clustering solution with complete dat and give average results over 20 runs.

6.1 Synthetic Datasets

We first present results on a simple clustering task in which the data is drawn from mixtures of Gaussians with two components. The data and results are shown in Figure 2. For each dataset we sampled a total of 200 points from two Gaussians of increasing separation. We normalized the data to lie in the unit hypercube and used a standard radial basis function kernel to compute similarities. As expected, random selection usually performs worst, except when the cluster separation is minimal. Compared to our two algorithms, S&T does poorly even on easy problems. As reported in [18], interleaving with random selection significantly boosts performance. IU-RED outperforms both



(a) (b) (c) (d) (e) (f)

Figure 3: Results on several real datasets. Methods are evaluated on the average misclustering error rate, relative to a spectral clustering solution with complete data. Our proposed methods are "IU-RED" and "IU-RED, interleave". Note that the scaling of the x-axis changes between plots.

versions of S&T early on, but ties with them towards the end. Interestingly, while interleaving IU-RED appears to eventually stabilize the error rates, it slightly hurts performance early on.

6.2 Real Datasets

We have also evaluated our algorithm on a variety of real datasets. Five of the sets in this subsection are from the UCI repository [8] (iris, wine, pendigits, waveform, segmentation). An additional dataset concerning the similarity of faces is available from the University of Washington [15]. We followed [11, 18] and normalized the data to lie in the unit hypercube and used the Gaussian kernel to compute similarities.[3] Figures 3(a)–(e) show the results on the UCI datasets. Figure 3(f) shows results on the face dataset. Note that the scaling of the x-axis changes between plots. The difference between methods is amplified on these datasets. Except for the iris dataset in Figure 3(a), S&T performs relatively poorly, and is eventually outperformed by random selection. On three of five UCI datasets IU-RED outperforms all other methods early on. Interleaving usually increases the error initially, but eventually leads to a more stable algorithm with marginally lower error. The exception is the waveform dataset in Figure 3(d) where interleaving helps significantly. Lastly, on the face dataset IU-RED also outperforms S&T early on, with slight gains for interleaving.

We conducted a further experiment to assess whether the superior performance of IU-RED over S&T can be seen after a single active learning step, or only emerges after many such steps. In this experiment, we sampled a subset of similarities of approximately constant size uniformly at random and measured the decrease in error rate over one active selection step. These results were averaged over 30000 restarts. The first five rows of Table 1 show results for UCI data, the sixth row for the face data, and the last row for similarity matrices with off-diagonal entries uniform in [0, 1]. On four out of five UCI datasets and on the face dataset, IU-RED decreases the error more than S&T. The one dataset where we perform

[3] Although the UCI datasets were previously analyzed in [18], the results are not directly comparable, since their evaluations used different kernel parameters. Also, since each of these datasets contains more than two classes, it is possible that we chose two different classes for evaluating the spectral clustering.

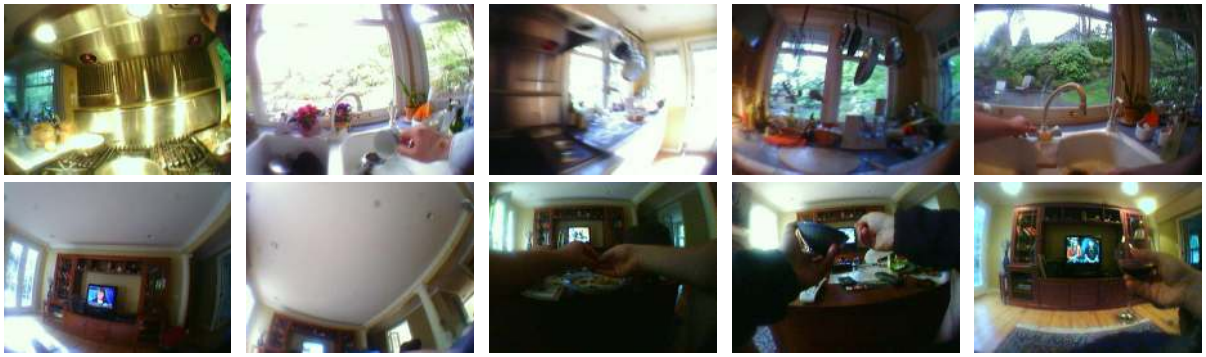


Figure 4: Spectral clustering of photos with complete data. The top row shows example views of the kitchen, and the bottom row example views of the living room. Humans can easily determine that photos in each row were probably taken in the same room, but a computer algorithm would have difficulty solving this task.

worse is the waveform dataset, which Figure 3(d) shows to be challenging for all methods. Overall, our method performs much better than S&T on structured similarity matrices. On random matrices both algorithms perform poorly, but now S&T performs better than IU-RED.

6.3 Wearable Camera Dataset

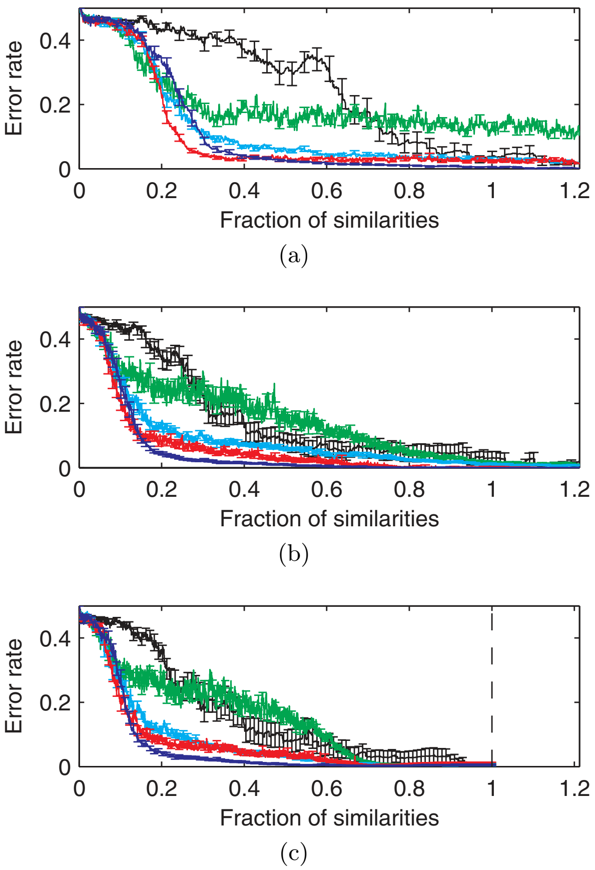
Our next experiment focuses on the realistic example outlined in the Introduction where similarities are hard to compute and noisy. The data of interest is a photo stream, acquired by a wearable camera at a rate of about one photo per 20 seconds. Clustering the data by the location at which an image was taken may be useful in a variety of applications, ranging from health (e.g., in diagnosis and life quality improvement for Alzheimer's patients) to summarization of personal memories. Because the data is collected in an entirely unconstrained way, analyzing it is beyond the capabilities of current unsupervised algorithms. The top row of Figure 4, for example, shows five images taken in the same kitchen. Clustering clearly requires some human input; at the very least a preliminary annotation that could be used to train supervised computer vision algorithms. It is impractical to ask annotators to simply label images by their location, since salient locations and their number only become evident as the stream progresses and may change from week to week, and from subject to subject. Also, locations may be interconnected, and multiple locations might be visible from the same viewpoint. In our data, for example, an open kitchen connects to the living room so that large parts of the space can be perceived to belong to both rooms. It is much more natural (and cost effective) to collect similarities between images and to infer a clustering from that data.

We took this approach in order to cluster 100 images taken from the photo stream described above [12]. Of these, about 50 were taken in an open kitchen, and 50 were taken in the adjacent living room. Some example images are shown in Figure 4. We asked workers on Amazon Mechanical Turk to rate, on a scale from 1 to 10, how likely it was that a given pair of images was taken in the same room, with 10 indicating certainty. The user ratings were divided by 10 and then used as similarities. Humans are adept at matching rooms by a loose jumble of visible objects, making this task fairly realistic. Indeed, the two rows of Figure 4 show representative examples from a partition that was found by spectral clustering using the complete median similarity matrix, with the median running over three repeat measurements. A subsampled version of the median similarity matrix was shown in Figure 1 on the left. To collect one similarity for each pair of photos costs a total of US$74, so the three repeats required for the median cost a total of US$222.

We first evaluated our algorithms on the median similarity matrix using the same methods as before. The results are shown in Figure 5(a). The legend is the same as in Figure 2. Note that the x-axis is scaled to extend beyond 1.0, to account for the three repeat measurements necessary to compute one median similarity. A fraction of f indicates that a total of $fn(n-1)/2$ pairwise measurements were made. As before, our method outperforms a number of competitors early on. The results can be also interpreted in terms of the amount of money that must be expended to achieve a clustering result of fixed quality. Each image comparison cost us US$0.045 on Amazon Mechanical Turk. Table 2 shows the resulting approximate cost in US$ for each algorithm in order to achieve an error rate of 0.05. IU-RED is at least 4 times cheaper than S&T, which costs more than 30% of the complete-labelling cost. This difference can easily render larger image clustering tasks than ours impractical.

We also evaluated how IU-RED and S&T compare over only one active choice. The result is shown in the row of Table 1 labeled "photos." As before, our active learning framework outperforms that of Shamir and Tishby.

Next, we consider the extension of IU-RED to deal with noisy similarities, as outlined in Section 5. Figure 6 illustrates the type of noise encountered in this labelling task. We allow up to three repeat measurements of similarities that are known with high uncertainty and which can potentially change the current solution. The results are shown in Figure 5(b). For comparison, Figure 5(c) shows results when no repeat measurements are allowed and the standard deviation is not estimated. The latter is the extreme counterpart to measuring every similarity three times. All error rates are relative to a spectral clustering computed from the complete median similarity matrix, averaged over 20 runs. As before, we scaled the x-axes to show the *effective* fraction of pairwise measurements that was made. For the no-repeats framework in Figure 5(c) this fraction cannot be larger than 1.0. Another important consequence of this measuring framework is that all algorithms should yield approximately the same



(a)

(b)

(c)

Figure 5: Results on the "photos" dataset. Figure (a) shows results when true similarities are estimated as the median of three measurements. Figure (b) shows results when the algorithm is allowed choose which repeat measurements to make. Up to three repeats are allowed. Figure (c) shows results when similarities are estimated by a single noisy measurement. The legend is the same as in Figure 3.

average error rate at a fraction of 1.0, since at that point every algorithm has observed one complete set of (noisy) similarities. The differences between algorithms are thus only appreciated in the first half of the Figure 5(c).

Both versions of IU-RED continue to beat the remaining algorithms in either of the two new settings. However, random interleaving now helps slightly, where it was detrimental before. With the exception of S&T, most methods improve early on compared to Figure 5(a). On the one hand this is intuitive, since as long as the similarities are not extremely noisy, three measurements do not convey three times as much information as one. One the other hand, it suggests that moderately noisy measurements can still be quite informative. Even so, we emphasize again that all algorithms should perform identically at a fraction of 1.0 in Figure 5(c). Maintaining running estimates and their accuracies is therefore preferable as it does not force performance to equalize once a fixed measurement quantum has been reached. Indeed, both IU-RED and IU-RED with random interleaving perform marginally better in Figure 5(b) than Figure 5(c) once a fraction of 1.0 measurements is reached.

Dataset	IU-RED	S&T
iris	0.0078 ± 0.0002	0.0016 ± 0.0001
wine	0.0103 ± 0.0003	-0.0010 ± 0.0003
pendigits	0.0067 ± 0.0002	-0.0007 ± 0.0002
waveform	-0.0018 ± 0.0002	0.0008 ± 0.0001
segmentation	0.0104 ± 0.0002	0.0002 ± 0.0002
face	0.0009 ± 0.0001	0.0001 ± 0.0001
photos	0.0126 ± 0.0003	-0.0021 ± 0.0002
uniform	-0.0057 ± 0.0009	-0.0037 ± 0.0007

Table 1: Average decrease in error rate across one selection step. IU-RED generally decreases the error more than S&T.

Random	S&T	S&T, inter.	IU-RED	IU-RED, inter.
$53	> $70	$32	$17	$21

Table 2: Approximate labelling costs in US$ to achieve a 0.05 error rate on the photos dataset.

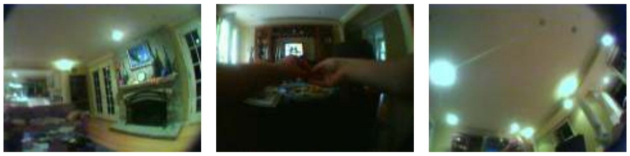


Figure 6: Worker disagreement for two image comparisons on Amazon Mechanical Turk. For the left two photos, workers agreed they were certainly taken in the same room; for the right two, one worker asserted they were definitely not.

7. CONCLUSION

In this paper we have presented and evaluated an active learning algorithm for spectral clustering. Our main insight is that similarity matrices are not random, but usually exhibit clear clustering structure. Even when observing only a small fraction of the data, this structure becomes evident. Furthermore, assuming that the data clusters well, the final v_2 will usually have elements well away from zero. Motivated by these observations, our algorithm uses the current estimate $\hat{v}_2$ to choose measurements that will be most useful in removing elements close to zero, i.e., to push the two clusters in $\hat{v}_2$ apart. We have applied this algorithm to a range of datasets and showed that it generally outperforms a related algorithm by Shamir and Tishby.

The effects of costly and noisy similarities have so far been ignored in the active learning setting. We propose an extension of our algorithm that maintains running estimates of the true similarities as well as their accuracies. By taking these accuracies into account during query selection, we can potentially avoid unnecessary repeat measurements and speed up the learning process in noisy settings.

Rahimi and Recht [16] previously showed that a version of spectral clustering related to normalized cuts [20] clusters data by finding a hyperplane that cuts the data in a lifted space. The signed distances of points to the hyperplane are given by rescaled elements of an Laplacian eigenvector, and the partitioning can be done by taking the sign of the distances. If we have that $W\mathbf{1} = c\mathbf{1}$, for some $c > 0$, then their result implies that our version of spectral clustering also finds such a hyperplane, and that the signed distances are proportional to the second eigenvector v_2. Our active learning approach can then be interpreted as choosing measurements that can maximally perturb the margin between a hyperplane and the lifted datapoints. Rahimi and Recht's observation has recently been used to derive an active learning rule for the spectral graph transducer [2]. Here W is completely known, but for each object an additional binary class label can be queried. The rule chooses to label that point next which is currently closest to a hyperplane. Similar heuristics have also been employed in a number of other classifiers and clustering frameworks [3, 13, 23, 27]. In all these methods, however, the pairwise similarities are assumed to be known initially (either implicitly or explicitly) but additional labels or constraints can be queried. In contrast, our setting allows for incomplete similarities which we can (perhaps only noisily) measure at high cost.

8. REFERENCES

[1] D. Achlioptas and F. McSherry. Fast Computation of Low-Rank Matrix Approximations. *Journal of the ACM*, 54, 2007.

[2] Z. Bodó, Z. Minier, and L. Csató. Active Learning with Clustering. *Journal of Machine Learning Research*, 16:127–139, 2011.

[3] C. Campbell, N. Cristianini, and A. Smola. Query Learning with Large Margin Classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 111–118. Morgan Kaufmann, 2000.

[4] P. Drineas and R. Kannan. Pass-Efficient Algorithms for Approximating Large Matrices. In *Proceedings of the Annual Symposium on Discrete Algorithms*, pages 223–232, 2003.

[5] P. Drineas, R. Kannan, and M. W. Mahoney. Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix. *SIAM Journal on Computing*, 36:158–183, 2006.

[6] P. Drineas and M. W. Mahoney. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral Grouping using the Nyström Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[8] A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010.

[9] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo Algorithms for Finding Low-Rank Approximations. *Journal of the ACM*, 51:1025–1041, 2004.

[10] T. Gärtner. A Survey of Kernels for Structured Data. *SIGKDD Explorations*, 2003.

[11] L. Huang, D. Yan, M. I. Jordan, and N. Taft. Spectral Clustering with Perturbed Data. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 705–712, 2009.

[12] N. Jojic, A. Perina, and V. Murino. Structural Epitome: A Way to Summarize One's Visual Experience. In *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 1027–1035. 2011.

[13] P. K. Mallapragada, R. Jin, and A. K. Jain. Active Query Selection for Semi-Supervised Clustering. In *ICPR*, pages 1–4. IEEE, 2008.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 849–856. MIT Press, 2002.

[15] University of Washington Information Design Lab. [idl.ee.washington.edu/SimilarityLearning/Applications/Datasets/]. Face Similarity Data, 2012.

[16] A. Rahimi and B. Recht. Clustering with Normalized Cuts is Clustering with a Hyperplane. In *Statistical Learning in Computer Vision*, 2004.

[17] T. Sakai and A. Imiya. Fast Spectral Clustering with Random Projection and Sampling. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 372–384, Berlin, Heidelberg, 2009. Springer-Verlag.

[18] O. Shamir and N. Tishby. Spectral Clustering on a Budget. *Proceedings of the Infernational Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[19] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[20] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[21] G. W. Stewart and J. Sun. *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Academic Press, 1990.

[22] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Investigation of Various Matrix Factorization Methods for Large Recommender Systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 6:1–6:8, New York, NY, USA, 2008. ACM.

[23] S. Tong and D. Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research*, 2:45–66, 2002.

[24] U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17:395–416, 2007.

[25] X. Wang and I. Davidson. Active Spectral Clustering. In *IEEE International Conference on Data Mining*, pages 561–568, 2010.

[26] C. K. I. Williams and M. Seeger. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 682–688. MIT Press, 2001.

[27] Q. Xu, M. desJardins, and K. L. Wagstaff. Active Constrained Clustering by Examining Spectral Eigenvectors. In *Proceedings of the International Conference on Discovery Science*, pages 294–307, 2005.