Active Clustering with Model-Based Uncertainty Reduction

Caiming Xiong, David M. Johnson, and Jason J. Corso *Senior Member, IEEE*

Abstract—Semi-supervised clustering seeks to augment traditional clustering methods by incorporating side information provided via human expertise in order to increase the semantic meaningfulness of the resulting clusters. However, most current methods are *passive* in the sense that the side information is provided beforehand and selected randomly. This may require a large number of constraints, some of which could be redundant, unnecessary, or even detrimental to the clustering results. Thus in order to scale such semi-supervised algorithms to larger problems it is desirable to pursue an *active* clustering method—i.e. an algorithm that maximizes the effectiveness of the available human labor by only requesting human input where it will have the greatest impact. Here, we propose a novel online framework for active semi-supervised spectral clustering that selects pairwise constraints as clustering proceeds, based on the principle of uncertainty reduction. Using a first-order Taylor expansion, we decompose the expected uncertainty reduction problem into a gradient and a step-scale, computed via an application of matrix perturbation theory and cluster-assignment entropy, respectively. The resulting model is used to estimate the uncertainty reduction potential of each sample in the dataset. We then present the human user with pairwise queries with respect to only the best candidate sample. We evaluate our method using three different image datasets (faces, leaves and dogs), a set of common UCI machine learning datasets and a gene dataset. The results validate our decomposition formulation and show that our method is consistently superior to existing state-of-the-art techniques, as well as being robust to noise and to unknown numbers of clusters.

Index Terms—active clustering, semi-supervised clustering, image clustering, uncertainty reduction

1 Introduction

Semi-supervised clustering plays a crucial role in machine learning and computer vision for its ability to enforce top-down structure while clustering [1], [2], [3], [4], [5], [6]. In these methods, the user is allowed to provide external semantic knowledge—generally in the form of constraints on individual pairs of elements in the data—as *side information* to the clustering process. These efforts have shown that, *when the constraints are selected well* [7], incorporating pairwise constraints can significantly improve the clustering results.

In computer vision, there are a variety of domains in which semi-supervised clustering has the potential to be a powerful tool. First, in surveillance videos, there is significant demand for automated grouping of faces and actions: for instance, recognizing that the same person appears at two different times or in two different places, or that someone performs a particular action in a particular location [11]. These tasks may be problematic for traditional supervised recognition strategies due to difficulty in obtaining training data—expecting humans to label a large set of strangers' faces or categorize every possible action that might occur in a video is not realistic. However, a human probably *can* reliably determine whether two face images are of the same person [12] or two recorded actions are similar, making it quite feasible to obtain pairwise constraints in these contexts.

The problem of plant identification is similar in that even untrained non-expert humans [13] (for instance, on a low-cost crowd-sourcing tool such as Amazon's Mechanical Turk [14]) can probably generally determine if two plants are the same species, even if only an expert could actually provide a semantic label for each of those images. Thus, non-expert labor, in conjunction with semi-supervised clustering, can reduce a large set of uncategorized images into a small set of clusters, which can then be quickly labeled by an expert. The same pattern holds true in a variety of other visual domains, such as identifying animals or specific classes of man-made objects, as well as non-visual tasks such as document clustering [15].

However, even when using relatively inexpensive human labor, any attempt to apply semi-supervised clustering methods to large-scale problems must still consider the cost of obtaining large numbers of pairwise constraints. As the number of possible constraints is quadratically related to the number of data elements, the number of possible user queries rapidly approaches a point where only a very small proportion of all constraints can feasibly be queried. Simply querying random constraint pairs from this space will likely generate a large amount of redundant information, and lead to very slow (and expen-

- *C. Xiong is senior researcher at Metamind Inc, Palo Alto, CA. E-mail: cmxiong.lhi@gmail.com*
- *D. M. Johnson is with the Department of Computer Science and Engineering, SUNY at Buffalo, NY. E-mail: davidjoh@buffalo.edu*
- *J. J. Corso is with the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor. E-mail: jjcorso@eecs.umich.edu*

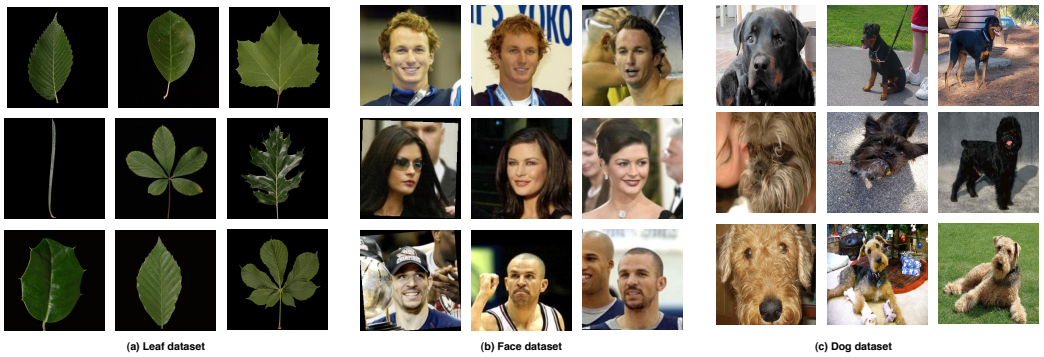


Fig. 1: Sample images from three image datasets: (a) Leaves [8]; (b) Faces [9]; (c) Dogs [10]. *Best viewed in color.*

sive) improvement in the clustering results. Worse, Davidson et al. [7] demonstrated that poorly chosen constraints can in some circumstances lead to worse performance than no constraints at all.

To overcome these problems, our community has begun exploring *active* constraint selection methods [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], which allow semi-supervised clustering algorithms to intelligently select constraints based on the structure of the data and/or intermediate clustering results. These active clustering methods can be divided into two categories: sample-based and sample-pair-based.

The sample-based methods first select samples of interest, then query pairwise constraints based on the selected sample [16], [18], [19]. Basu et al. [16] propose offline (i.e., not based on intermediate clustering results) active k-means clustering based on a two-stage process that first explores the problem space and performs user queries to initialize and grow sets of samples with known cluster assignments, and then extracts a large constraint set from the known sample sets and does semi-supervised clustering. Mallapragada et al. [18] present another active k-means method based on a min-max criterion, which also utilizes an initial "exploration" phase to determine the basic cluster structure. We have also previously proposed two different sample-based active clustering methods [19], [21]. This paper represents an improvement and extension of these works.

By contrast, the sample-pair-based methods [22], [23], [24], [25], [26] directly seek pair constraints to query. Hoi et al. [23] provide a min-max framework to identify the most informative pairs for non-parametric kernel learning and provide encouraging results. However, the complexity of that method (which requires the solution of an approximate semidefinite programming (SDP) problem) is high, limiting both the size of the data and the number of constraints that can be processed. Xu et al. [22] and Wang and Davidson [24] both propose active spectral clustering methods, but both of them are designed for two-class problems, and poorly suited to the multiclass case. Most recently, Biswas and Jacobs [12] propose a method that seeks pair constraints that maximize the *expected change* in the clustering result. This proves to be a meaningful and useful criterion, but the proposed method requires recomputing potential clustering results many times for each sample-pair selected, and is thus slow.

Both types of current approaches suffer from drawbacks: most current sample-based methods are offline algorithms that select all of their constraints in a single selection phase before clustering, and thus cannot incorporate information from actual clustering results into their decisions. Most pair-based methods are online, but have very high computational complexity due to the nature of the pair selection problem (i.e. the need to rank $\mathbf{O}(n^2)$ candidate pairs at every iteration), and thus have severely limited scalability.

In this paper, we overcome the limitations of existing methods and propose a novel sample-based active spectral clustering framework using *certain-sample sets* that performs efficient and effective sample-based constraint selection in an online iterative manner (certain-sample sets are sets containing samples with known pairwise relationships to all other items in the certain-sample sets). In each iteration of the algorithm, we find the sample that will yield the greatest predicted **reduction** in clustering **uncertainty**, and generate pairwise queries based on that sample to pass to the human user and update the certain-sample sets for clustering in the next iteration. Usefully, under our framework the number of clusters need not be known at the outset of clustering, but can instead be discovered naturally via human interaction as clustering proceeds (more details in Section 3).

In our framework, we refer to the sample that will yield the greatest expected uncertainty reduction as the **most informative sample**, and our active clustering algorithm revolves around identifying and querying this sample in each iteration. In order to estimate the uncertainty reduction for each sample, we propose a novel approximated first-order model which decomposes expected uncertainty reduction into two components: a gradient and a step-scale factor. To

estimate the gradient, we adopt matrix perturbation theory to approximate the first-order derivative of the eigenvectors of the current similarity matrix with respect to the current sample. For the step-scale factor we use one of two entropy-based models of the current cluster assignment ambiguity of the sample. We describe our framework and uncertainty reduction formulation fully in Section 3.

We compare our method with baseline and state-of-the art active clustering techniques on three mid-size image datasets (face images [9], leaf images [8] and dog images [10]), two large-scale image datasets (Caltech-101 [28] and ImageNet-100 [29]), a set of common UCI machine learning datasets [30] and a gene dataset [31]. Sample images from some of these sets can be seen in Figure 1. Our results (see Section 7) show that given the same number of pairs queried, our method performs significantly better than existing state-of-the-art techniques.

2 Background and Related Work

What is clustering uncertainty? Clustering methods are ultimately built on the relationships between pairs of samples. Thus, for any clustering method, if our data perfectly reflects the "true" relationship between each sample-pair, then the method should always achieve the same perfect result. In practice, however, data (and distance/similarity metrics) are imperfect and noisy—the relationship between some pairs of samples may be clear, but for others it is highly ambiguous. Moreover, some samples may have predominantly clear relationships to other samples in the data, while others may have predominantly ambiguous relationships. Since our goal in clustering is to make a decision about the assignment of samples to a cluster, despite the inevitable ambiguity, we can view the overall sample-relationship ambiguity in the data as the uncertainty of our clustering result.

We then posit that the advantage of semi-supervised clustering is that it eliminates some amount of uncertainty, by removing all ambiguity from pair relationships on which we have a constraint. It thus follows that the goal of active clustering should be to choose constraints that *maximally* reduce the total sample-assignment uncertainty. In order to achieve this, however, we must somehow measure (or at least estimate) the uncertainty contribution of each sample/sample-pair in order to choose the one that we expect to yield the greatest reduction. In this paper, we address this problem by proposing a novel first-order model of uncertainty reduction based on matrix perturbation theory and the concept of local entropy (more details in Section 3.2).

Why *sample*-based uncertainty reduction? There are two main reasons for proposing a sample-based approach rather than a sample-pair-based one. First, an uncertain pair may be uncertain either because it contains one uncertain sample or because it contains *two* uncertain samples. In the latter case, because the constraint between these samples will not extrapolate well beyond them, it yields limited information. Second, due to the presence of n^2 pair constraints for every n samples, pair selection has an inherently higher complexity, which limits the scalability of a pair-based approach.

A naïve approach to sample-based uncertainty, however, has clear disadvantages. Querying the label of a sample rather than querying a pairwise constraint requires an understanding of the class structure of the problem, which is not available in many clustering applications. For this reason, we do not query samples directly. Rather, once a sample is selected we generate pairwise constraints between it and other representative samples, then query the user with regard to these pairs, we thus allow our sample-based method to operate using only pairwise queries. More details are introduced in section 3.

Relation to active learning. Active query selection has previously seen extensive use in the field of active learning [32], [33]. Huang et al. [34] and Jain and Kapoor [35], for example, both offer methods similar to ours in that they select and query uncertain samples. However, in active learning algorithms the oracle (the human) needs to know the class label of the queried data point. This approach is not applicable to many semi-supervised clustering problems, where the oracle can only give reliable feedback about the relationship between pairs of samples (such as the many examples we offered in the Section 1). Though we implicitly label queried samples by comparing them to a set of exemplar samples representing each cluster, we do so strictly via pairwise queries.

Additionally, for the sake of comparison we begin our experiments with an exploration phase that identifies at least one member of each cluster (thus allowing us to treat the clusters we are learning as "classes" as far as the active learning algorithms are concerned), but in real data this may not be a reliable option. There may simply be too many clusters to fully explore them initially, new clusters may appear as additional data is acquired, or certain clusters may be rare and thus not be encountered for some time. In all of these cases, our active clustering framework can adapt by simply increasing the number of clusters. In contrast, most active learning methods must be initialized with at least one sample of each class in the data, and do not allow online modification of the class structure.

3 Active Clustering Framework With Certain-Sample Sets

Recall that "certain-sample sets" are sets such that any two samples in the same certain-sample set are constrained to reside in the same cluster, and any two

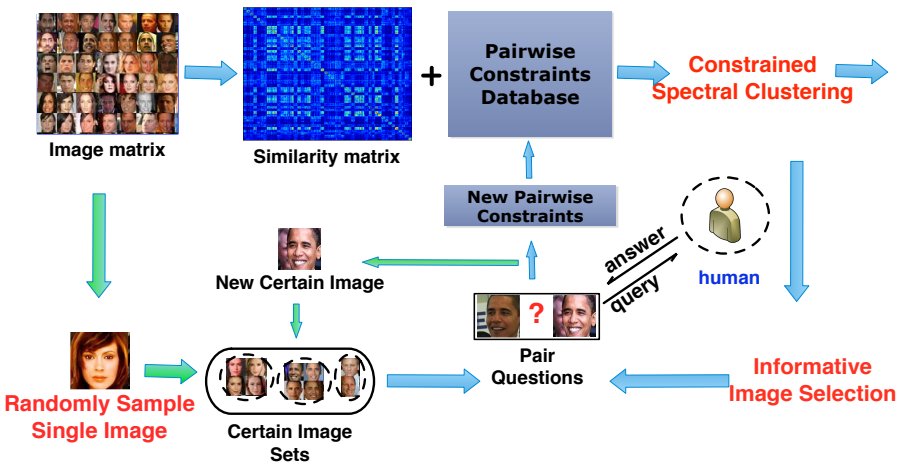


Fig. 2: Pipeline of our active clustering framework as applied to image clustering. We iteratively choose a maximally informative image, then select and query new pairwise constraints based on the chosen image, update the certain image sets, and refine the clustering results before returning to select a new most informative image.

samples from different certain-sample sets are guaranteed to be from different clusters. In the ground-truth used in our experiments, each class corresponds to a specific certain-sample set. In our framework, we use the concept of certain-sample sets to translate a sample selection into a set of pairwise constraint queries.

Given the data set $X = \{x_1, x_2, \cdots, x_n\}$, denote the corresponding pairwise similarity matrix $\mathbf{W} = \{w_{ij}\}$ (i.e. the non-negative symmetric matrix consisting of all w_{ij}, where w_{ij} is the similarity between samples x_i and x_j). Similarity is computed in some appropriate, problem-specific manner.

Here, we also denote the set of certain-sample sets $\mathcal{Z} = \{Z_1, \cdots, Z_m\}$, where Z_i is a certain-sample set such that $Z_i \subset X$ and $Z_i \cap Z_j = \emptyset$ for all j, and define an sample set $\mathcal{O} = \bigcup_i Z_i$ containing *all* current certain sample. Our semantic constraint information is contained in the set Q, which consists of all the available pairwise contraints. Each of these constraints may be either "must-link" (indicating that two samples belong in the same semantic grouping/certain-sample set) or "cannot-link" (indicating that they do not). To initialize the algorithm, we randomly select a single sample x_i such that $Z_1 = \{x_i\}$ with $\mathcal{Z} = \{Z_1\}$, $\mathcal{O} = \{x_i\}$ and $Q = \emptyset$. As $\mathcal{Z}$, $\mathcal{O}$ and $\mathcal{Q}$ change over time, we use the notation $(\cdot)^t$ to indicate each of these and other values at the t^{th} iteration.

Assuming we begin with no pairwise constraints, if the number of clusters in the problem is not known, set the initial cluster number $n_c = 2$, otherwise set it to the given number. We then propose the following algorithm (outlined in Figure 2, more details for each step can be found in Sections 3.1–3.3):

1 **Initialization:** randomly choose a single sample x_i, assign x_i to the first certain set Z_1 and initialize the pairwise constraint set $\mathcal{Q}$ as the empty set.
2 **Constrained Spectral Clustering:** cluster all sample into n_c groups using the raw data X plus the current pairwise constraint set Q.
3 **Informative Sample Selection:** choose the most informative sample x_j based on our uncertainty reduction model.
4 **Pairwise Queries:** present a series of pairwise queries on the chosen sample x_j to the oracle until we have enough information to assign the sample x_j to a certain-sample set Z_k (or create a new certain set for the chosen sample).
5 **Repeat:** steps 2-4 until the oracle is satisfied with the clustering result or the query budget is reached.

It should be noted that, aside from the ability to collect maximally useful constraint information from the human, this algorithm has one other significant advantage: the number of clusters in the problem need not be known at the outset of clustering, but can instead be discovered naturally via human interaction as the algorithm proceeds. Whenever the queried pairwise constraints result in the creation of a new certain-sample set, we increment n_c to account for it. This allows the algorithm to naturally overcome a problem faced not just by other active clustering (and active learning) methods, but by clustering methods in general, which typically require a parameter controlling either the size or number of clusters to generate. This is particularly useful in the image clustering domain, where the true number of output clusters (e.g. the number of unique faces in a dataset) is unlikely to be initially available in any real-world application. We have conducted experiments to evaluate this method of model selection; the results, which are encouraging, are presented in Section 7.6.

Recalling the steps of our framework, from here we proceed iteratively through the three main computational steps: clustering with pairwise constraints, informative sample selection and querying pairwise constraints. We now describe them.

3.1 Spectral clustering with pairwise constraints

Spectral clustering is a well-known unsupervised clustering method [36]. Given the $n \times n$ symmetric similarity matrix $\mathbf{W}$, denote the Laplacian matrix as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{D}$ is the degree matrix such that $\mathbf{D} = \{d_{ij}\}$, where $d_{ij} = \sum_k \mathbf{W}_{ik}$ if $i = j$ and 0 otherwise. Spectral clustering partitions the n samples into n_c groups by performing k-means on the first n_c eigenvectors of L. The n_c eigenvectors can be found via:

$$\begin{aligned} \mathbf{v} = & \quad \underset{\mathbf{v}}{\operatorname{argmin}}\, \mathbf{v}^T L \mathbf{v} \\ = & \quad \underset{\mathbf{v}}{\operatorname{argmin}} \sum_{ij} w_{ij} \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \\ & s.t. \quad \mathbf{v}^T\mathbf{v} = I, \mathbf{v}^T\mathbf{1} = 0 \ . \end{aligned} \tag{1}$$

To incorporate pairwise constraints into spectral clustering, we adopt a simple and effective method called spectral learning [37]. Whenever we obtain new pairwise constraints, we directly modify the current similarity matrix $\mathbf{W}^t$, producing a new matrix $\mathbf{W}^{t+1}$. Specifically, the new affinity matrix W^{t+1} is determined via:

- Set $\mathbf{W}^{t+1} = \mathbf{W}^t$.
- For each pair of must-linked samples (i, j) assign the values $\mathbf{W}_{ij}^{t+1} = \mathbf{W}_{ji}^{t+1} = 1$.
- For each pair of cannot-linked samples (i, j) assign the value $\mathbf{W}_{ij}^{t+1} = \mathbf{W}_{ji}^{t+1} = -1$.

We then obtain the new Laplacian matrix L^{t+1} and proceed with the standard spectral clustering procedure.

3.2 Informative sample selection

In this section, we formulate the problem of finding the most informative sample as one of uncertainty reduction. We ultimately develop and discuss a model for this uncertainty reduction in Section 4.

Define the uncertainty of the dataset in the t^{th} iteration to be conditioned on the current updated similarity matrix $\mathbf{W}^t$ and the current certain-sample set $\mathcal{O}^t$. Thus the uncertainty can be expressed as $\mathbf{U}(X|\mathbf{W}^t, \mathcal{O}^t)$. Therefore our objective function for sample selection is as follows:

$$\begin{aligned} \mathbf{x}_j^* &= \underset{\mathbf{x}_j \in X}{\operatorname{argmax}}\, \Delta\mathbf{U}(\mathbf{x}_j) \ . \\ \Delta\mathbf{U}(\mathbf{x}_j) &= \mathbf{U}(X|\mathbf{W}^t, \mathcal{O}^t) - \mathbf{U}(X|\mathbf{W}^t, \mathcal{O}^t \cup \{x_j\}) \ . \end{aligned} \tag{2}$$

To the best of our knowledge, there is no direct way of computing uncertainty on the data. In order to optimize this objective function, we consider that querying pairs to make a chosen sample "certain" can remove ambiguity in the clustering solution and thus reduce the uncertainty of the dataset as a whole. So the expected change in the clustering solution that results from making the chosen sample "certain" can be considered as the uncertainty contribution of the sample as a result of selecting and querying that sample.

Thus, we seek samples that will have the greatest impact on the clustering solution. One strategy for finding these constraints (employed in Biswas and Jacobs [12], though with sample-pairs rather than samples) is to estimate the likely value of a constraint (i.e. cannot- or must-link) and simulate the effect that constraint will have on the clustering solution. However, this approach is computationally expensive (in the worst case requiring a simulated clustering operation for each possible constraint at each iteration of the active clusterer).

We hence adopt a more indirect method of estimating the impact of a sample query, based on matrix perturbation theory and the local cluster assignment entropy of each sample. We present the details of our method in Section 4.

3.3 Sample-based pairwise constraint queries

Before presenting our model for informative sample selection, we briefly describe how we use the selected sample. Because our active selection system is sample-based and our constraints pair-based, once we have selected the most informative sample we must then generate a set of pairwise queries related to that sample. Our goal with these queries is to obtain enough information to add the sample to the correct certain-sample set. We generate these queries as follows.

First, for each certain set Z_j, choose the single sample within the set that is closest to the selected sample $\mathbf{x}_i$ ($\mathbf{x}_l = \operatorname{argmax}_{\mathbf{x}_l \in Z_j} w_{il}$) and record this sample.

Second, since there are m certain sets, we will have recorded m sample and similarity values. We sort these samples based on their corresponding similarity, then, in order of descending similarity, query the oracle for the relation between the selected sample $\mathbf{x}_i$ and $\mathbf{x}_l$ until we find a must-link connection. We then add $\mathbf{x}_i$ into the certain-sample set containing that x_l. If all of the relations are cannot-link, we create a new certain-sample set Z_{m+1} and add $\mathbf{x}_i$ to it. This new certain set Z_{m+1} is then added to $\mathcal{Z}$. Regardless, $\mathcal{O}$ is correspondingly updated by adding $\mathbf{x}_i$. If the value of m after querying is greater than n_c, we also update n_c to reflect the newly discovered ground-truth cluster.

In Figure 3, we present a toy example to visualize the behavior of our algorithm. We discuss our method for identifying informative samples in more detail below.

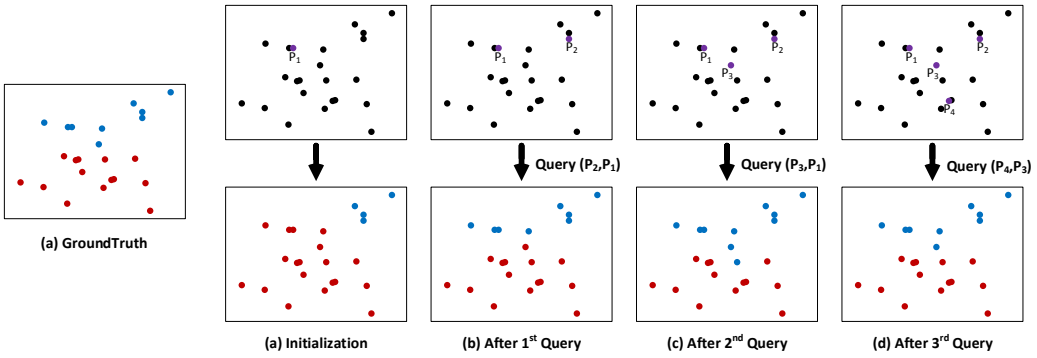


Fig. 3: A simple example of our uncertainty reducing active clustering method on toy data. The initial clustering result is poor, and the first certain sample P1 is chosen randomly. After this, however, the algorithm quickly identifies informative samples and queries the oracle to determine the ground truth relationships between each chosen point. Within 3 iterations, our method has explored the space and determined the correct borders of the two clusters.

4 Uncertainty Reduction Model for Informative Sample Selection

As described in Section 3.1, we use spectral learning [37] as our clustering algorithm. In spectral learning [37], the clustering result arises from the values of the first n_c eigenvectors of the current similarity matrix. Therefore, the impact of a sample query on the clustering result can be approximately measured by estimating its impact on V^{n_c} (the first n_c eigenvectors v_k):

$$\begin{aligned}\Delta\mathbf{U}(x_j) \approx & \Delta\mathbf{V}^{n_c}(x_j) \\ = & \sum_{k=0}^{n_c} \Delta v_k(x_j) \ . \end{aligned} \tag{3}$$

In order to measure $\Delta\mathbf{V}^{n_c}(x_j)$, based on a first-order Taylor expansion, we decompose the change in the eigenvectors into a gradient and a step-scale factor:

$$\Delta\mathbf{V}^{n_c}(x_j) = \frac{\partial V^{n_c}(x_j)}{\partial H(x_j)} \Delta H(x_j) \ , \tag{4}$$

where $H(x_j)$ represents the assignment-ambiguity of x_j, and $\Delta H(x_j)$ represents the reduction in this ambiguity after querying x_j. $\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)}$ is a first-order derivative of the changes in the eigenvectors as a result of this ambiguity reduction. We describe how to estimate this gradient and ambiguity reduction in Sections 4.1 and 4.2, respectively.

4.1 Estimating the uncertainty reduction gradient

In order to solve (4) we must first evaluate $\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)}$. We know that in spectral learning (Section 3.1) the information obtained from the oracle queries is expressed via changes in the similarity values for the queried point contained in $\mathbf{W}^t$. Given this, changes in ambiguity are always mediated by changes in $\mathbf{W}^t$, so we can approximate $\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)}$ via

$$\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)} \approx \frac{\partial V^{n_c}(x_j)}{\partial \mathbf{W}^t_{x_j}} \ , \tag{5}$$

where $\partial\mathbf{W}^t_{x_j}$ represents an incremental change in the similarity values of sample x_j.

Thus, we must begin by computing $\frac{\partial V^{n_c}(x_j)}{\partial \mathbf{W}^t_{x_j}}$ for each x_j, for which we propose a method based on matrix perturbation theory [38]. First note that the graph Laplacian at iteration t can be fully reconstructed from the eigenvectors and corresponding eigenvalues via $L^t = \sum_{i=1}^{n} \lambda_i v_i v_i^T$. Then, given a small constant change in a similarity value w^t_{jk}, the first-order change of the eigenvector v_i can be calculated as:

$$\frac{dv_i}{dw^t_{jk}} = \sum_{p \neq i} \frac{v_i^T \left(\partial L^t / \partial w^t_{jk}\right) v_p}{\lambda_i - \lambda_p} v_p \tag{6}$$

Note that $\partial L^t / \partial w^t_{jk} = (e_j - e_k)(e_j - e_k)^T$, where e_q is the n-length indicator vector of index q.

For the chosen sample x_j we take n_c samples $X_{n_c} = \{x_{j_1}, x_{j_2}, \cdots, x_{j_{n_c}}\}$, one sampled from each certain set $Z_i \in \mathcal{Z}$. If we decide to query the oracle for x_j, the relation of x_j to each sample in X_{n_c} will become known, and the corresponding w^t_{jk} in W^t will be updated during spectral learning. Therefore, to estimate the influence of sample x_j on the gradient of the eigenvectors, we can simply sum the influences of the relevant w^t_{jk} values based on Eq. 6. We thus define our approximate model for the derivative of uncertainty reduction as:

$$\begin{aligned}\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)} \approx & \sum_{i=1}^{n_c} \left| \sum_{x_k \in X_{n_c}} \frac{dv_i}{dw^t_{jk}} \right| \\ = & \sum_{i=1}^{n_c} \left| \sum_{x_k \in X_{n_c}} \sum_{p \neq i} \frac{v_i^T [\partial L^t / \partial w^t_{jk}] v_p}{\lambda_i - \lambda_p} v_p \right| \ . \end{aligned} \tag{7}$$

Note that we operate only over a subset of certain samples in order to both save on computation and avoid redundancy. We could simply use the entirety of $\mathcal{O}$ in place of X_{n_c}, but this would likely distort the results. Intuitively, the effect of a must-link constraint is to shift the eigenspace representations of the two constrained samples together. The samples in a certain set should thus have very similar eigenspace representations, so we expect additional constraints between them and x_j to have diminishing returns.

4.2 Estimating the step scale factor for uncertainty reduction

The second component of our uncertainty reduction estimation is $\Delta H(x_j)$—the change in the ambiguity of the sample x_j as a result of querying that sample. This component serves as the step scale factor for the gradient $\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)}$. According to the assumptions in Section 3.3, after a sample is queried the ambiguity resulting from that sample is reduced to 0. This leads to the conclusion that

$$\Delta H(x_j) = H(x_j) \ . \tag{8}$$

Therefore, the problem of estimating the change in ambiguity of a sample reduces to the problem of estimating the current ambiguity of that sample. While this problem still cannot be solved precisely, we present two reasonable heuristics for estimating the ambiguity of a sample. Both are based on the concept of entropy—specifically, the entropy over probability distributions of local cluster labels (an uncertainty estimation strategy that has shown good results in active learning [32]).

Nonparametric structure model for cluster probability First, consider the current clustering result $C^t = \{c_1, c_2, \cdot, c_{n_c}\}$, where c_i is a cluster and n_c is the number of clusters. We can then define a simple nonparametric model based on similarity matrix W for determining the probability of x_j belonging to cluster c_i:

$$P(c_i|x_j) = \frac{\sum_{x_l \in c_i} w_{jl}}{\sum_{x_l \in X} w_{jl}} \tag{9}$$

Because only local nearest neighbors have large similarity values in relation to a given sample, we can use the k-nearest neighbors (kNN) of each point to efficiently approximate the entropy. These neighbors need only be computed once, so this ambiguity estimation process is fast and scalable. In our experiments, we use $k = 20$.

Parametric model for cluster probability Alternately, we can simply use the eigenspace representation of our data produced by the most recent semi-supervised spectral clustering operation to compute a probabilistic clustering solution. We elect to learn a mixture model (MM) on the embedded eigenspace of the current similarity matrix W^t for this purpose:

$$p(\mathbf{x}_j|\{\alpha_c\},\theta_c\}) = \sum_{c=1}^{n_c} \alpha_c f(\mathbf{x}_j;\theta_c) \ , \tag{10}$$

where $\{\alpha_c\}$ are the mixing weights and $(\theta_c\})$ are the component parameters. Then, the probability of each data point given each cluster c is computed via:

$$P(c|\mathbf{x}_j) = \frac{\alpha_c f(\mathbf{x}_j;\theta_c)}{\sum_{c=1}^{n_c} \alpha_c f(\mathbf{x}_j;\theta_c)} \ . \tag{11}$$

In our experiments, we assume a Gaussian distribution for each component, yielding a Gaussian Mixture Model (GMM).

Entropy-based ambiguity model Whether using the parametric or nonparametric cluster probability model, the ambiguity of sample x_j can be defined, based on entropy, as:

$$H(x_j) = -\sum_{i=1}^{n_c} P(c_i|x_j) \log P(c_i|x_j) \tag{12}$$

We then use this value to approximately represent $\Delta H(x_j)$. In combination with the approximate uncertainty gradient $\frac{\partial V^{n_c}(x_j)}{\partial H(x_j)}$ computed as in Section 4.1, this allows us to evaluate (4) and effectively estimate the uncertainty reduction for every point x_j. From there, solving our sample selection objective (2) is a simple argmin operation.

5 Complexity Analysis

At each iteration, we must select a query sample from among $\mathbf{O}(n)$ possibilities, applying our uncertainty reduction estimation model to each potential sample. Computing the gradient component of the uncertainty model takes $\mathbf{O}(m{n_c}^2 n)$ time for each sample, where m is number of certain sets and n_c is the number of clusters/eigenvectors. $m \leq n_c$, so the complexity of the uncertainty gradient evaluation at each iteration is $\mathbf{O}(n_c^3 n^2)$. Computing all the step scale factors costs $\mathbf{O}(n_c k n)$ (where k is the number of nearest neighbors) if the nonparametric method is used, or $\mathbf{O}(n_c^3 n)$ for the parametric method. $k \ll n$, so regardless the total complexity of the active selection process at each iteration is $\mathbf{O}(n_c^3 n^2)$.

In order to reduce this cost, we adopt a slight approximation. In general, the samples with the largest uncertainty reduction will have both a large step scale and a large gradient. With this mind, we first compute the step scale for each sample (this is cheaper than computing the gradient, particularly if the nonparametric model is used), then only compute the gradient for the b samples with the largest step scales. Assuming $b \ll n$, this yields an overall complexity of $\mathbf{O}(n_c^3 n)$. Note that all results for our method shown in this paper were obtained using this fast approximation, except those for URASC-GO (one of the variants of

TABLE 1: UCI machine learning and gene data sets

Dataset	Size	Dim.	No. Classes
Balance	625	4	3
BUPA Liver Disorders	345	6	2
Diabetes	768	8	2
Sonar	208	60	2
Wine	178	13	3
Cho's gene	307	100	5

our method discussed in section 6.3). Also note that for large data, the cost of the method will generally be dominated by the spectral clustering itself, which is $\mathbf{O}(n^3)$ in the worst case (though potentially significantly cheaper, possibly even $\mathbf{O}(n)$ [39], [40], depending on the eigendecomposition method used and the sparseness of the similarity matrix).

6 Experimental Setup

6.1 Data

We evaluate our proposed active framework and selection measures on three medium-size image datasets (leaves, dogs and faces—see Figure 1), two large image datasets (Caltech-101 [28] and ImageNet-100 [29]), one gene dataset [31] and five UCI machine learning datasets [30]. We seek to demonstrate that our method is generally effective for different types of data/applications with a wide range of cluster numbers.

Face dataset: all face images are extracted from a face dataset called PubFig [9], which is a large, real-world face dataset consisting of 58,797 images of 200 people collected from the Internet. Unlike most other existing face datasets, these images are taken in completely uncontrolled settings with non-cooperative subjects. Thus, there is large variation in pose, lighting, expression, scene, camera, and imaging conditions. We use two subsets: **Face-1** (500 images from 50 different people) and **Face-2** (200 images from 20 different people).

Leaf dataset: all leaf images are iPhone photographs of leaves against a monochrome background, acquired through the Leafsnap app [8]. We use the same subset (1042 images from 62 species) as in [12]. The feature representations and resulting similarity matrices for the leaf and face datasets are all from [12].

Dog dataset: all dog images are from the Stanford Dogs dataset [10], which contains 20,580 images of 120 breeds of dogs. We extract a subset containing 400 images from 20 different breeds and compute the features used in [29]. Affinity is measured via a χ^2 kernel.

Gene and UCI machine learning datasets: we choose five datasets from the UCI repository and **Cho**'s [31] gene dataset (details in Table 1). Affinity is measured via a Gaussian kernel.

We also evaluate our methods on two well-known large-scale datasets: Caltech-101 [28] and ImageNet-100 [29].

Caltech-101 image dataset: contains 101 object categories with 40 to 800 images per category, with more than 8,000 images in total. For each image, we compute the PHOW [41] feature via VLFEAT [43].

ImageNet-100 image dataset: ImageNet [29] is a well-known large-scale image dataset that includes millions of annotated images. We select 100 categories, containing 10,000 images in total, from the ImageNet database. For each image, we use the feature representation from [42].

6.2 Evaluation protocols

We evaluate all cluster solutions via two commonly used cluster evaluation metrics: the Jaccard Coefficient [44] and V-measure [45].

The **Jaccard Coefficient** is defined by $\text{JCC} = \frac{SS}{SD+DS+SS}$, where:

- **SS**: represents the total number of pairs that are assigned to the same cluster in both the clustering results and the ground-truth.
- **SD**: represents the total number of pairs that are assigned to the same cluster in the clustering results, but to different clusters in the ground-truth.
- **DS**: represents the total number of pairs that are assigned to different clusters in the clustering results, but to the same cluster in the ground-truth.

V-Measure is an alternate metric for determining cluster correspondence between a set of ground-truth classes C and clusters K, which defines entropy-based measures for the completeness and homogeneity of the clustering results, and computes the harmonic mean of the two.

6.3 Baseline and state-of-the-art methods

To evaluate our active clustering framework and proposed active constraint selection strategies, we test the following set of methods, including a number of variations on our own proposed method, as well as a baseline and multiple state-of-the-art active clustering and learning techniques. From this point forward we refer to our proposed method as Uncertainty Reducing Active Spectral Clustering (URASC). The variants of URASC:

- **URASC+N:** Proposed model for uncertainty reducing active clustering with gradient and non-parametric step scale estimation.
- **URASC+P:** Proposed model for uncertainty reducing active clustering with gradient and parametric step scale estimation.
- **URASC-GO:** Our model without step scale estimation—only the gradient estimation for each sample is used.

- **URASC-NO:** Our model without gradient estimation—only the nonparametric step scale is used.
- **URASC-PO:** Our model without gradient estimation—only the parametric step scale is used.

Our baselines and comparison methods include state-of-the-art pair-based active clustering methods and two active learning methods:

- **Random**: A baseline in which pair constraints are randomly sampled from the available pool and fed to the spectral learning algorithm.
- **Active-HACC:** [12] An active hierarchical clustering method that seeks pairs that maximize the expected change in the clustering.
- **CAC1:** [25] An active hierarchical clustering method that heuristically seeks constraints between large nearby clusters.
- **FFQS** [16]: An offline active k-means clustering method that uses certain-sample sets to guide constraint selection (as in our method), but selects samples to query either through a farthest-first strategy or at random.
- **ASC** [24]: A binary-only pair-based active spectral clustering method that queries pairs that will yield the maximum reduction in expected pair value error.
- **QUIRE** [34]: A binary-only active learning method that computes sample uncertainty based on the informativeness and representativeness of each sample. We use our certain-sample set framework to generate the requested sample labels from pairwise queries.
- **pKNN+AL** [35]: A minmax-based multi-class active learning method. Again, we use our framework to translate sample label requests into pairwise constraint queries.

7 Results

We run our method and its variants on all of the listed datasets and compare against baselines and competing state-of-the-art techniques.

7.1 Variant methods and baseline

In Figure 4, we compare our parametric and nonparametric methods, as well as the three "partial" URASC procedures, on three image sets and two UCI sets at varying numbers of constraints. We show results in terms of both Jaccard coefficient and V-measure, and witness similar patterns for each. In all cases, our parametric and nonparametric methods perform relatively similarly, with the nonparametric having a modest lead at most, but not all, constraint counts. More importantly, our methods consistently (and in many cases dramatically) outperform the random baseline, particularly as the number of constraints increases. Our methods always show notable improvement as more constraints are provided—in contrast to the random baseline, which, *at best*, yields minor improvement. Even on the relatively simple wine dataset, it is clear that randomly selected constraints yield little new information.

Finally, we note that our "complete" methods consistently meet or exceed the performance of the corresponding partial methods. Neither the step-scale-only methods nor the gradient-only method consistently yield better results, but in every case the combined method performs at least on-par with the better of the two, and in some cases significantly better than either (see the sonar results in particular). These results validate the theoretical conception of our method, showing that the combination of gradient and step-scale is indeed the correct way to represent the active selection problem, and that our method's performance is being driven by the combined information of both terms.

7.2 Comparison to state-of-the-art active learning methods

We next compare our methods to two active learning methods, as representatives of other pair-based techniques (Figure 5). Here we test on three binary UCI datasets in order to provide a reasonable evaluation of the QUIRE method, which is binary-only.

At least one (and usually both) of our methods outperforms both QUIRE and pKNN+AL in most cases, only definitively losing out at the very low constraint level on the sonar dataset. As with the random baseline before, the gap between our methods and the competition generally increases with the number of constraints. These results suggests that simply plugging active learning methods into a clustering setting is suboptimal—we can achieve better results by formulating a clustering-specific uncertainty reduction objective.

Also notable is the fact that, between the two active learning methods, QUIRE is clearly the superior (at least on problems where it is applicable). This is significant because, like our method, QUIRE seeks to measure the global impact of a given constraint, while pKNN+AL only models local uncertainty reduction. This lends further support to the idea that the effect of a given query should be considered within the context of the entire clustering problem, not just in terms of local statistics.

7.3 Comparison to state-of-the-art active clustering methods

Finally, we test our methods against existing active clustering techniques (as well as the random baseline) and represent the results visually in Figure 6. Not all methods appear in all charts because ASC

Jaccard Coefficient							
Dataset	#pairwise constraints	Random	URASC+N	URASC+P	URASC-NO	URASC-PO	URASC-GO
Dog dataset	1000	0.0289	**0.075**	0.0484	0.0627	0.0438	0.0626
	2000	0.032	0.2294	**0.2557**	0.2006	0.2079	0.1632
	3000	0.034	0.7199	**0.7409**	0.6783	0.7158	0.694
Face-1 Dataset	1000	0.1256	**0.1623**	0.147	0.1573	0.1383	0.1376
	2000	0.1318	**0.224**	0.2155	0.1947	0.1823	0.1627
	3000	0.1392	0.4017	**0.4775**	0.3798	0.4041	0.3764
Leaf dataset	500	0.3287	**0.4249**	0.3301	0.4092	0.3281	0.3266
	1500	0.3374	**0.4557**	0.3483	0.4532	0.3324	0.3358
	2500	0.3409	**0.6862**	0.439	0.6259	0.4184	0.428
	3000	0.3435	**0.7754**	0.6775	0.7026	0.6379	0.6028
Wine	5	0.8252	0.837	**0.8565**	0.837	**0.8544**	0.837
	10	0.836	0.8565	**0.9123**	0.8929	**0.9122**	0.8726
	15	0.8371	**0.9342**	0.9123	0.9122	0.9124	0.901
Sonar	50	0.3463	**0.3707**	0.352	0.3594	0.352	0.3483
	150	0.3464	**0.8182**	0.7103	0.6908	0.671	0.6717
	180	0.3448	**0.9124**	**0.8939**	0.7891	0.8191	0.7758
V-Measure							
Dataset	#pairwise constraints	Random	URASC+N	URASC+P	URASC-NO	URASC-PO	URASC-GO
Dog dataset	1000	0.1867	**0.3361**	0.2601	0.3188	0.2548	**0.3371**
	2000	0.212	**0.633**	0.5937	0.5673	0.5499	0.581
	3000	0.229	0.9252	**0.9326**	0.9124	0.8974	0.8965
Face-1 Dataset	1000	0.6191	**0.6634**	0.6401	**0.6635**	0.6217	0.5868
	2000	0.6319	**0.7317**	0.7156	0.709	0.7044	0.7262
	3000	0.6404	0.8567	**0.8632**	0.8482	0.825	0.8608
Leaf dataset	500	0.8021	**0.8385**	0.806	0.8291	0.8019	0.8096
	1500	0.8065	**0.8579**	0.8116	0.8567	0.8252	0.8161
	2500	0.8108	**0.9432**	0.8538	0.9313	0.8483	0.8572
	3000	0.8064	**0.9613**	0.9402	0.9475	0.9281	0.9187
Wine	5	0.7969	0.8389	**0.8579**	0.8387	0.8518	0.8387
	10	0.8213	0.8579	**0.9016**	0.8925	**0.9087**	0.8656
	15	0.8387	**0.9281**	0.9016	0.9088	0.909	0.9013
Sonar	50	0.0018	**0.0641**	0.0001	0.0479	0.0001	0.0017
	150	0.0018	**0.7152**	0.6248	0.5396	0.5783	0.5695
	180	0.0033	**0.8593**	0.8386	0.6621	0.7497	0.7073

Fig. 4: Comparison of variants of our methods against the random baseline.

	Jaccard Coefficient					V-Measure			
Dataset	#pairwise constraints	URASC+N	URASC+P	QUIRE	pKNN+AL	URASC+N	URASC+P	QUIRE	pKNN+AL
Sonar	50	0.3707	0.352	**0.4684**	0.4237	0.0641	0.0001	**0.1268**	0.073
	150	**0.8182**	0.7103	0.6551	0.6415	**0.7154**	0.6248	0.4925	0.5018
	180	**0.9124**	0.8939	0.8174	0.8315	**0.8593**	0.8386	0.703	0.7233
Bupa	100	**0.509**	0.435	0.4258	0.4383	**0.2079**	0.1619	0.1877	0.1085
	200	**0.6001**	0.5796	0.4592	0.5639	**0.4039**	0.3868	0.2639	0.3202
	300	**0.8952**	0.8863	0.7135	0.7623	**0.8088**	**0.7972**	0.6019	0.6169
Diabetes	150	**0.5661**	0.4855	0.4777	0.383	**0.2113**	0.0846	0.1781	0.0621
	300	**0.6173**	0.493	0.5915	0.4176	**0.378**	0.1676	0.3479	0.1146
	450	0.6303	0.6067	**0.6414**	0.5102	**0.4606**	0.3705	0.4291	0.2681
Face-1 Dataset	1000	**0.1623**	0.147		0.1367	**0.6634**	0.6401		0.6367
	2000	**0.224**	0.2155		0.2032	**0.7317**	0.7156		0.6984
	3000	0.4017	**0.4775**		0.4554	0.8567	**0.8632**		0.8431
Dog dataset	1000	**0.075**	0.0484		0.0419	**0.3361**	0.2601		0.243
	2000	0.2294	**0.2557**		0.2262	**0.633**	0.5937		0.5601
	3000	0.7199	**0.7409**		0.6378	0.9252	**0.9326**		0.8574
Leaf dataset	500	**0.4249**	0.3301		0.3487	**0.8385**	0.806		0.8142
	1500	**0.4557**	0.3483		0.3599	**0.8579**	0.8116		0.8236
	2500	**0.6862**	0.439		0.488	**0.9432**	0.8538		0.8649

Fig. 5: Comparison of our methods against sample-based active learning methods. **Since QUIRE is a binary-only method, there is no result for QUIRE [34] on the multi-cluster datasets.**

[24] is applicable only to binary data. Once again, our methods present a clear overall advantage over competing algorithms, and in many cases both our parametric and nonparametric methods far exceed the performance of any others (most dramatically on the Dog dataset).

The only method that comes near to matching our general performance is Active-HACC, which also seeks to estimate the expected change in the clustering as a result of each potential query. However, this method is much more expensive than ours (due to running a large number of simulated clustering operations for every constraint selection) and fails on the Dog dataset. ASC is also somewhat competitive with our methods, but its binary nature greatly limits its usefulness for solving real-world semi-supervised clustering problems.

Neither of our two variant methods has a clear advantage over the other, though the nonparametric approach appears to be more reliable given the relative failure of the parametric algorithm on the Leaf and Diabetes sets. Also noteworthy is the tendency on a number of datasets for URASC+P to have relatively poor performance early on and improve (again

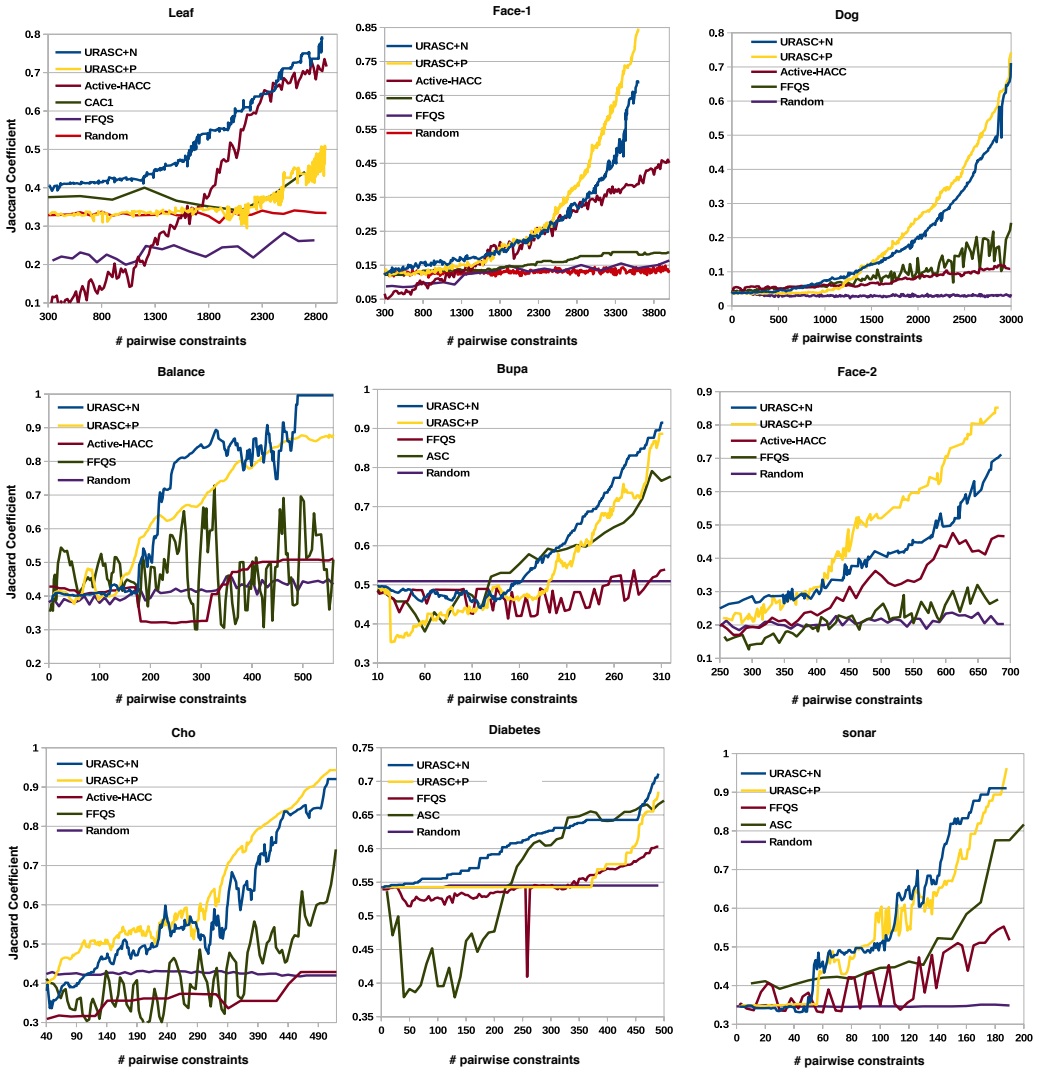


Fig. 6: Comparison to state-of-the-art active clustering methods. y-axis is Jaccard Coefficient score. *Best viewed in color.*

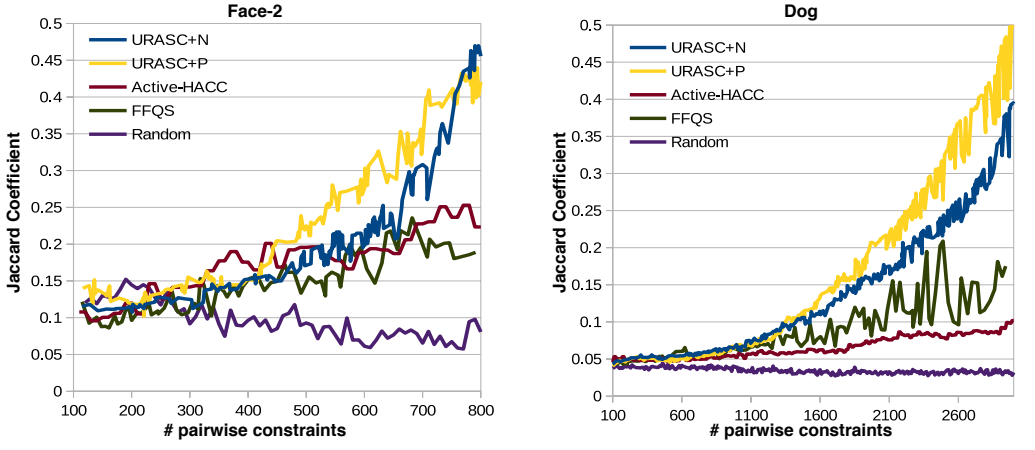


Fig. 7: Comparison of our methods against other active clustering methods on two image datasets with a 2% simulated error rate on the oracle queries. *Best viewed in color.*

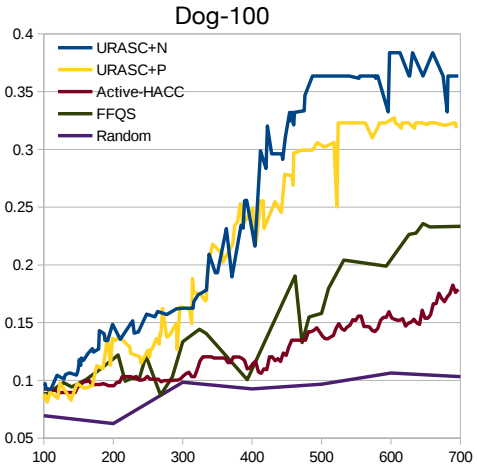


Fig. 8: Comparison of our methods against other active clustering methods on the dog-100 image dataset when using real human input to acquire the pairwise constraints. *Best viewed in color.*

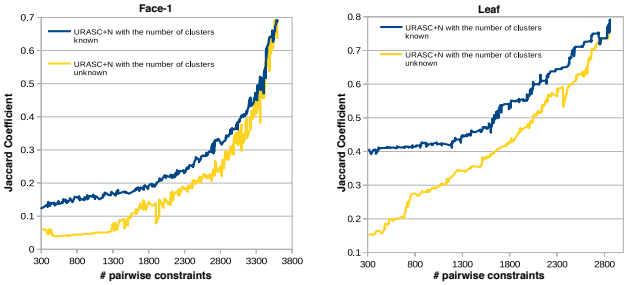


Fig. 9: Comparison of URASC+N clustering results with known and (initially) unknown numbers of clusters. *Best viewed in color.*

relatively) as the number of queries increases. We suspect this is because the underlying GMM used by this method is also improving as more constraints are obtained, thus allowing the query selection method *itself* to improve over time.

7.4 Comparison with synthesized noisy input

Our previous experiments are all based on the assumption that the oracle reliably returns a correct ground-truth response every time it is queried. Previous works in active clustering have also relied on this assumption [16], [17], [18], [22], [24], [25], [26]. Obviously, this is not, as a general rule, realistic—human oracles may make errors, and in some problems the ground-truth itself may be ambiguous and subjective. Specifically, for the face and leaf datasets used here, Amazon Mechanical Turk experiments [9], [12] have shown that human error is about 1.2% on face queries and 1.9% on leaf queries.

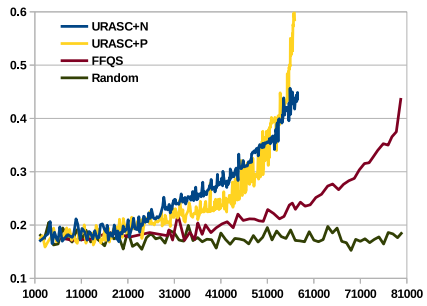
Thus, we performed a set of experiments with a simulated uniform 2% query error rate on the Face-2 and Dog datasets. We plot the results of our experiment in Figure 7, and find that, while improvement is noticeably slower and noisier than in the previous experiments, our algorithms still demonstrate a significant overall advantage over other active or passive clustering techniques.

7.5 Comparison with noisy input from real world human input

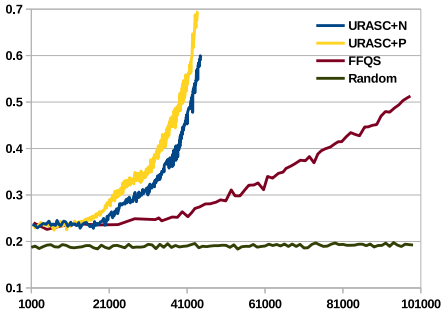
In order to better verify our method, besides the experiment on the perfect labels of pairwise constraints and synthesized noise labels, we design a new, more realistic experiment that obtains noisy label input directly from real humans. We extract 100 dog images with 10 classes from Stanford Dogs dataset [10] and call it the Dog-100 dataset, we have hired eight people with different education backgrounds, and then exhaustively label all pairs of dog images; if they think the pair is from same class, click 'YES, otherwise click 'NO'. For each pair of images, we collect at least three responses and adopt a majority voting strategy to get the label of the pair. According to the human labels, we found that the error rate of the human input is 5.2%. This error rate is higher than the human error rate in Biswas and Jacobs [12]; we suspect this is due to the greater variability in the dog images than the leaf images, which leads to more difficult human judgements. Then we test our method and other comparison methods on this human labeled dataset. In Figure 8, we display these comparative results. First, we observe that the performance plateaus after the human-performance level has been reached, which confirms a notion introduced in Biswas and Jacobs [12]. Second, comparing with other methods, our methods still show state-of-the-art performance, which is same as discussed in the other experiments..

7.6 Comparison with unknown numbers of clusters

Since one advantage of our method is its ability to dynamically discover the number of clusters based on query results, we analyze how this approach effects performance over time. We thus run our method on the Face-1 (50 ground-truth clusters) and Leaf (62 ground-truth clusters) datasets, with the number of clusters k initially set to 2, and increasing as new certain-sample sets are discovered. Our results are shown in Figure 9. The results are promising, with the unknown-k results initially much lower (as expected), but converging over time towards the known-k results as the cluster structure is discovered. On both datasets tested, the results appear to eventually become indistinguishable.



(a) Caltech-101



(b) ImageNet-100

Fig. 10: Results of our experiments on large-scale image datasets. Note that Active-HACC could not be run on this data due to memory limitations.

7.7 Comparison on large-scale image datasets

To evaluate our proposed framework's scalability and performance on more realistic large-scale data, we run two experiments on significantly larger image datasets: Caltech-101 [28] (>8000 images) and ImageNet-100 [29](10,000 images). The results are shown in Figure 10. In this experiment, we were unable to run Active-HACC [12] due to its algorithmic and memory requirements (as implemented by the authors, the method requires upwards of 100 GB of main memory for ImageNet-100). The trends we observe for this data are similar to what we see on smaller datasets. Cluster improvement is initially slow, then accelerates greatly as our algorithm accumulates information about the problem. Our parametric method, in particular, seems to eventually reach a point of dramatically increasing returns on new queries, again likely due to the improving quality of its underlying model of the data.

The performance of the other methods is significantly weaker. FFQS is effective, but produces a slower, more linear improvement than our methods, failing to match their results even when allotted more than twice as many queries. By contrast, random query selection on this data appears to yield *no* improvement in clustering quality, further emphasizing the need for strong active methods.

8 Conclusion

In this paper, we present a novel sample-based online active spectral clustering framework that actively selects pairwise constraint queries with the goal of minimizing the uncertainty of the clustering problem. In order to estimate uncertainty reduction, according to first-order Taylor expansion, we decompose it into a gradient (estimated via matrix perturbation theory) and step-scale (based on one of two models of local label entropy). We then use pairwise queries to disambiguate the sample with the largest estimated uncertainty reduction. Our experimental results validate this decomposed model of uncertainty and support our theoretical conception of the problem, as well as demonstrating performance significantly superior to existing state-of-the-art algorithms. Moreover, our experiments show that our method is robust to noise in the query responses and functions well even if the number of clusters in the problem is initially unknown.

One avenue of future research involves reducing the computational burden of the active selection process by adjusting the algorithm to select multiple query samples at each iteration, so that this active spectral clustering method could become a powerful tool for use in large-scale online problems, particularly in the increasingly popular crowdsourcing domain.

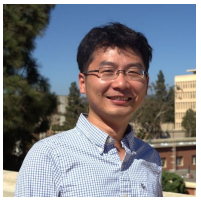
Acknowledgements

We are grateful for the support in part provided through the following grants: NSF CAREER IIS-0845282, ARO YIP W911NF-11-1-0090, DARPA Minds Eye W911NF-10-2-0062, DARPA CSSG D11AP00245, and NPS N00244-11-1-0022. Findings are those of the authors and do not reflect the views of the funding agencies.

References

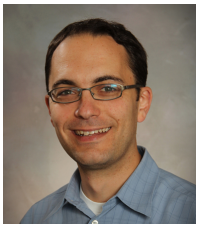
[1] S. Basu, M. Bilenko, and R. Mooney, "A probabilistic framework for semi-supervised clustering," in *SIGKDD*. ACM, 2004, pp. 59–68.
[2] Z. Li and J. Liu, "Constrained clustering by spectral kernel learning," in *ICCV*. IEEE, 2009, pp. 421–427.
[3] Z. Lu and M. Carreira-Perpinán, "Constrained spectral clustering through affinity propagation," in *CVPR*. IEEE, 2008, pp. 1–8.
[4] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *NIPS*, pp. 521–528, 2003.
[5] S. Hoi, R. Jin, and M. Lyu, "Learning nonparametric kernel matrices from pairwise constraints," in *ICML*. ACM, 2007, pp. 361–368.
[6] L. Chen and C. Zhang, "Semi-supervised variable weighting for clustering," in *SDM*, 2011, pp. 862–871.
[7] I. Davidson, K. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," in *ECML PKDD*. Springer, 2006.
[8] "http://leafsnap.com/", 2012.
[9] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *ICCV*. IEEE, 2009, pp. 365–372.
[10] A. Khosla, N. Jayadevaprakash, B. Yao, and F. Li, "Novel dataset for fine-grained image categorization," in *CVPR Workshops*, 2011.

[11] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *CVPR*, 2013.
[12] A. Biswas and D. Jacobs, "Active image clustering with pairwise constraints from humans," *IJCV*, 2014.
[13] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares, "Leafsnap: A computer vision system for automatic plant species identification," in *ECCV*. Springer, 2012, pp. 502–516.
[14] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk a new source of inexpensive, yet high-quality, data?" *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.
[15] R. Huang and W. Lam, "An active learning framework for semi-supervised document clustering with language modeling," *DKE*, vol. 68, no. 1, pp. 49–67, 2009.
[16] S. Basu, A. Banerjee, and R. Mooney, "Active semi-supervision for pairwise constrained clustering," in *ICDM*, 2004.
[17] Y. Fu, B. LI, X. Zhu, and C. Zhang, "Do they belong to the same class: active learning by querying pairwise label homogeneity," in *CIKM*. ACM, 2011, pp. 2161–2164.
[18] P. Mallapragada, R. Jin, and A. Jain, "Active query selection for semi-supervised clustering," in *ICPR*. IEEE, 2008, pp. 1–4.
[19] C. Xiong, D. M. Johnson, and J. J. Corso, "Online active constraint selection for semi-supervised clustering," in *ECAI*, 2012, pp. 12–17.
[20] C. Xiong, D. M. Johnson, and J. J. Corso, "Spectral active clustering via purification of the k-nearest neighbor graph," in *ECDM*, 2012, pp. 133–141.
[21] C. Xiong, D. M. Johnson, and J. J. Corso, "Uncertainty reduction for active image clustering via a hybrid global-local uncertainty model," in *AAAI (Late-Breaking Developments)*, 2013, pp. 149–151.
[22] Q. Xu, M. Desjardins, and K. Wagstaff, "Active constrained clustering by examining spectral eigenvectors," in *Discovery Science*. Springer, 2005, pp. 294–307.
[23] S. Hoi and R. Jin, "Active kernel learning," in *ICML*. ACM, 2008, pp. 400–407.
[24] X. Wang and I. Davidson, "Active Spectral Clustering," in *ICDM*, 2010.
[25] A. Biswas and D. Jacobs, "Large scale image clustering with active pairwise constraints," in *ICML Workshops*, 2011.
[26] F. Wauthier, N. Jojic, and M. Jordan, "Active spectral clustering via iterative uncertainty reduction," in *SIGKDD*. ACM, 2012, pp. 1339–1347.
[27] S. Xiong, J. Azimi, and X. Z. Fern, "Active learning of constraints for semi-supervised clustering," *TKDE*, vol. 99, no. PrePrints, p. 1, 2013.
[28] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *CVIU*, vol. 106, no. 1, pp. 59–70, 2007.
[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*. IEEE, 2009, pp. 248–255.
[30] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[31] R. Cho, M. Campbell, E. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. Wolfsberg, A. Gabrielian, D. Landsman, D. Lockhart *et al.*, "A genome-wide transcriptional analysis of the mitotic cell cycle," *Molecular Cell*, vol. 2, no. 1, pp. 65–73, 1998.
[32] B. Settles, "Active learning literature survey," *University of Wisconsin, Madison*, 2010.
[33] A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang, "Agnostic active learning without constraints," *Arxiv preprint arXiv:1006.2588*, 2010.
[34] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples." Advances in Neural Information Processing Systems, 2010.
[35] P. Jain and A. Kapoor, "Active learning for large multi-class problems," in *CVPR*. IEEE, 2009, pp. 762–769.
[36] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *NIPS*, vol. 2, pp. 849–856, 2002.
[37] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher, "Spectral learning," in *IJCAI*. Stanford InfoLab, 2003.
[38] G. Stewart and J. Sun, *Matrix perturbation theory*. Academic press New York, 1990, vol. 175.
[39] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation." in *AAAI*, 2011.
[40] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *SIGKDD*. ACM, 2009, pp. 907–916.
[41] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, vol. 2. IEEE, 2006, pp. 2169–2178.
[42] H. Agrawal, N. Chavali, M. C., A. Alfadda, , P. Banik., and D. Batra, "Cloudcv: Large-scale distributed computer vision as a cloud service," 2013. [Online]. Available: http://cloudcv.org
[43] A. Vedaldi and B. Fulkerson, "VLFeat: An Open and Portable Library of Computer Vision Algorithms," 2008. Available: http://www.vlfeat.org/
[44] T. Pang-Ning, M. Steinbach, and V. Kumar, "Introduction to data mining," *WP Co*, 2006.
[45] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure." in *EMNLP*, 2007, pp. 410–420.



Caiming Xiong received the B.S. and M.S. from Huazhong University of Science and Technology in 2005 and 2007, respectively, and the Ph.D. in computer science and engineering from SUNY Buffalo in 2014. He is a senior researcher at Metamind Inc, Palo Alto, CA. He was a Postdoctoral Researcher in the Department of Statistics at the University of California, Los Angeles from 2014-2015. His research interests include interactive learning and clustering, deep learning, video understanding, and human-robot interaction. Note that the work in this paper was completed when he was a PhD student at SUNY Buffalo.

David M. Johnson received the BS Degree in neuroscience from Brandeis University in 2009. He is currently working toward the PhD degree in the Computer Science and Engineering Department, SUNY Buffalo, and currently working as a visiting scholar at the Electrical Engineering and Computer Science Department of the University of Michigan. His research interests include active and semisupervised learning and computer vision.



Jason J. Corso received the BS(Hons.) degree from Loyola College, MD, USA, in 2000 and the MSE and PhD degrees from the Johns Hopkins University in 2002 and 2005, respectively, all in computer science. He is an associate professor of Electrical Engineering and Computer Science at the University of Michigan. He spent two years as a post-doctoral fellow at the University of California, Los Angeles. From 2007-2014 he was a member of the computer science and engineering faculty at SUNY Buffalo. He received the Google Faculty Research Award 2015, the Army Research Office Young Investigator Award 2010, NSF CAREER award 2009, SUNY Buffalo Young Investigator Award 2011, a member of the 2009 DARPA Computer Science Study Group, and received the Link Foundation Fellowship in Advanced Simulation and Training 2003. He has authored more than 100 peer-reviewed papers on topics of his research interest including computer vision, robot perception, data science, and medical imaging. He is a member of the AAAI, ACM, MAA and a senior member of the IEEE.