Online Active Constraint Selection For Semi-Supervised Clustering

Caiming Xiong and David Johnson and Jason J. Corso [1]

Abstract.

Due to strong demand for the ability to enforce top-down structure on clustering results, semi-supervised clustering methods using pairwise constraints as *side information* have received increasing attention in recent years. However, most current methods are *passive* in the sense that the side information is provided beforehand and selected randomly. This may lead to the use of constraints that are redundant, unnecessary, or even harmful to the clustering results. To overcome this, we present an active clustering framework which selects pairwise constraints online as clustering proceeds, and propose an online constraint selection method that actively selects pairwise constraints by identifying uncertain nodes in the data. We also propose two novel methods for computing node uncertainty: one global and parametric and the other one local and nonparametric. We evaluate our active constraint selection method with two different semi-supervised clustering algorithms on UCI, digits, gene and image datasets, and achieve results superior to current state of the art active techniques.

1 Introduction

Many semi-supervised clustering methods have been proposed to enforce top-down structure while clustering [3, 5, 11, 15, 16, 23]. These methods allow the user to incorporate pairwise constraints, which may be either must-link (the two points/nodes belong in the same cluster) or cannot-link (the two points/nodes belong in different clusters), on the data as *side information*. These papers have shown that the use of pairwise constraints can significantly improve the correspondence between clusters and semantic labels *when the constraints are selected well*. [7] demonstrates that poorly chosen constraints can lead to worse performance than no constraints at all. Moreover, in real world problems each added constraint represents an additional real world cost, so maximizing the effectiveness of each constraint in order to minimize the total number of constraints needed is an important goal.

Currently, most work in semi-supervised clustering ignores this problem and simply selects a random constraint set (see above cited), but some work has now been done on *active* constraint selection methods [2, 10, 17, 21, 24], which allow semi-supervised clustering algorithms to intelligently select constraints based on the structure of the data and/or intermediate clustering results.

Active selection methods can be stratified according to whether nodes or node-pairs are the primary element on which the process is based. Node-based methods first select nodes of interest, and then query constraints based on those nodes [2,9,17], while those methods that directly seek pair constraints [10, 21, 24], define an uncertainty measure on pairs and iteratively seek the most uncertain pairs during constraint selection.

Both of these current approaches have drawbacks, however. Current node-based methods function by selecting all of their constraints in one long selection phase before clustering. Because of this, they cannot incorporate information from actual clustering results into their decisions, and may thus choose many unnecessary constraints (for instance, constraints regarding points that the algorithm is able to cluster correctly even without side information). In contrast, the pair-based methods choose constraints online based on intermediate clustering results, but due to the nature of the pair selection problem (n^2 possible constraints to rank and select from) have thus far been limited to either binary or small-scale clustering problems.

In this paper, we overcome these limitations and present a node-based constraint selection framework (illustrated in Figure 1) that combines a preprocessing stage with an online, iterative process that selects uncertain nodes based on intermediate clustering results. Our framework is general to any pair-based semi-supervised clustering algorithm.

In the preprocessing stage, we adopt the farthest-first strategy to identify representative nodes in each cluster (similar to [2] and [17]). Subsequently, we repeat an online cluster-select-query loop to allow data and clustering-dependent constraints to be actively added to the constraint set. In each iteration, our algorithm seeks the most uncertain node in the dataset. The algorithm then queries constraints between this node and previously identified nodes representing each cluster. An online oracle then provides the must-link or cannot-link value for each constraint.

We define two notions of node uncertainty, from a local nonparametric and one from a global parametric view. The local measure is based on cluster contradiction between a point and its nearest neighbors, while the global measure is computed from cluster confusion in a global mixture model of the data.

Why uncertainty? If enough information has been extracted to unambiguously identify the correct relationship between each true cluster and each node, then the clustering algorithm should achieve perfect accuracy [2, 17]. Intuitively, then, uncertainty in the cluster identity of nodes leads to clustering errors. Furthermore, in this paper we show that, given some relaxation, the most uncertain node is the same as the node that will maximally reduce the uncertainty of the whole dataset if queried. Thus, by issuing queries that will unambiguously identify the cluster of the most uncertain node at each iteration we are able to make optimal progress toward a completely certain, and thus trivially solvable, clustering problem.

Why *node* uncertainty? We adopt an approach based on node rather than pair uncertainty for two reasons. First, an uncertain pair may be uncertain either because it contains one uncertain node or

[1] Department of Computer Science and Engineering, SUNY at Buffalo

because it contains *two* uncertain nodes. In the latter case, a constraint between these nodes yields limited information because the constraint will not extrapolate well beyond these two nodes. Second, pair selection has an inherently higher complexity due to the presence of n^2 constraints for every n nodes, limiting the scalability of a pair-based approach.

Main contributions. Our paper makes four contributions:

- We present a new active selection framework, **O**nline **C**onstraint **S**election via **N**ode **U**ncertainty (**OCSNU**), with general applicability to pair-based semi-supervised clustering methods.
- We propose and justify two novel definitions of node uncertainty.
- We show theoretically that selecting the most uncertain node at each iteration yields the maximum reduction in total uncertainty for the dataset.
- We test our framework and constraint selection methods on two different semi-supervised clustering algorithms and conclusively obtain strong results.

We compare OCSNU with baseline and state of the art active clustering and active learning techniques on UCI machine learning datasets, two digits datasets [1], one gene dataset [6] and part of the Caltech 101 image dataset [8]. The results show that given the same number of pairs queried, OCSNU using our node uncertainty measures can obtain better accuracy than existing methods.

Relation to active learning. Active query selection has previously seen extensive use in the field of active learning. [20] and [4], for example, both offer methods similar to ours in that they select and query uncertain nodes. However, in active learning algorithms, the oracle needs to know the class label of the queried data point. This approach is not applicable to many clustering problems, where the oracle can only give feedback about the relation between pairs of nodes. Though we implicitly label queried nodes by comparing them to a set of exemplar nodes representing each cluster, we do so strictly via pairwise queries.

Additionally, though for the sake of comparison we begin our experiments with the cluster structure fully explored (i.e. at least one example of each class identified), in real data this may not be the case—the total number of clusters may not be known during the initial exploration phase. Our method can dynamically add new clusters as needed to adapt to such situations, while these (and most other) active learning methods cannot.

2 Active constraint selection for semi-supervised clustering

In this section, we first present the framework for our active semi-supervised clustering, then describe details of each framework-component and two novel node uncertainty models/definitions for use in active constraint selection. Throughout the rest of the paper, let X be a data set with n nodes, $X = \{x_1, x_2, \cdots, x_n\}$ and $x_i \in R^d$ with k total clusters.

2.1 Active semi-supervised clustering framework

We now present our framework for active clustering—recall the basic flow of the algorithm depicted earlier in Figure 1. We divide the whole framework into two parts: the exploration preprocess and the online active selection/clustering process. We begin by running the exploration preprocess once, before computing any clustering results. The goal is to obtain a set of exemplar nodes Q^0, with (hopefully but not necessarily) at least one representing each true cluster.

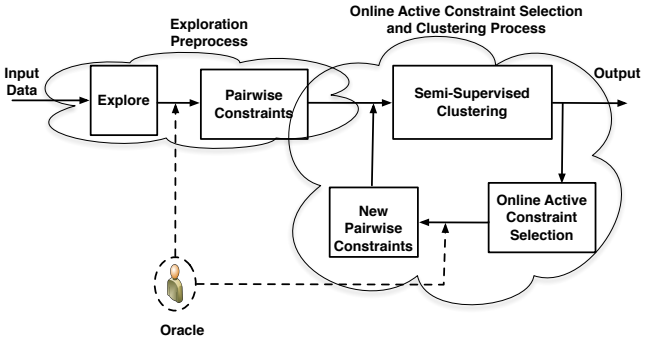

Figure 1. Flowchart of OCSNU for active clustering. First run the exploration preprocess to obtain a set of exemplar nodes and associated pair constraints, then input constraints to the online active constraint selection and clustering process. Second, enter into the online cluster-select-query loop, in which we repeatedly do semi-supervised clustering based on the current constraints, then use the results of the clustering to actively select new constraints and query the oracle. This process continues until either the oracle is satisfied or a fixed number of queries have been made.

After this, the data and the initial Q^0 serve as inputs to the online active clustering process. In each iteration of this process, our approach will automatically select and query new pair constraints based on the current clustering results and add the newly obtained constraints into the current constraint set, then recluster using the new constraint set. We iterate the online active clustering process until the oracle is satisfied or a fixed number of queries have been made.

2.2 Exploration preprocess

In the exploration preprocess phase, we adopt the exploration process from [2], which uses a farthest-first traversal strategy to query and identify at least one node from each of the k clusters in a reasonably small number of attempts. The identified nodes from the same cluster are collected into a single node set, so there are at most k non-empty node sets. [2] prove that this strategy yields a significant $\ln k$ factor improvement in the number of queries required to identify the needed exemplars, as compared to random selection.

We denote the preprocessing as the 0^{th} iteration in the online process, and when it is finished we will have obtained k_0 node sets for use, such that the set of node sets $NS^0 = \{X_1, X_2, \cdots, X_{k_0}\}$, where each $X_i \in NS^0$ is a node set , and $k_0 \leq k$ since preprocessing could be stopped before k non-empty node sets are found. The initial identified nodes set $Q^0 = \bigcup_{X_i \in NS^0} X_i$. [2]

Using these node sets, we then begin the active clustering process, using the set of constraints represented in NS^0 (i.e. must-link constraints among all nodes in the same set, and cannot-link constraints among nodes in different sets) as input to an initial semi-supervised clustering operation.

2.3 Online active constraint selection

After obtaining a set of output clusters from a semi-supervised clustering algorithm, our method will then perform an active online

[2] Note the set of node sets NS^0 will be updated after each iteration in the online active clustering process. After the t^{th} iteration, the set of node sets NS^{t-1} will be updated to NS^t and $Q^t = \bigcup_{X_i \in NS^t} X_i$.

search for new constraints to further improve the clustering. As we described earlier, at each iteration we identify the most uncertain node in the dataset and query the relationship between this node and some of the previously identified exemplars. We show here that the most uncertain node is (given some relaxation), also the node that will yield the maximum reduction in total node uncertainty for the dataset when queried.

In the t^{th} iteration, Q^{t-1} is the set of queried nodes, which we consider to be "certain" (i.e. if we were clustering only these nodes we could do so trivially and unambiguously) and $X = Q^{t-1} \bigcup \neg Q^{t-1}$, where $\neg Q^{t-1}$ is the set of unqueried nodes. We define the uncertainty $\mathbf{U}$ of the dataset in the t^{th} iteration to be conditioned on the original data distribution and previous semi-supervised clustering results, which are in turn based on the queried node set Q^{t-1}. Thus the uncertainty can be expressed as $\mathbf{U}(\cdot|M(X, C^{t-1}))$, where C^{t-1} is clustering result at iteration $t-1$ and $M(\cdot,\cdot)$ is a model for calculating uncertainty (we provide two such models here).

For a set of nodes X', denote $\mathbf{U}(\{X'\}; M(X, C^{t-1})) = \sum_{x_i \in X'} \mathbf{U}(\{x_i\}; M(X, C^{t-1}))$ if all nodes in X' are conditionally independent under $M(X, C^{t-1})$. Since queried nodes are considered to be "certain" , we assume $\mathbf{U}(X; M(X, C^{t-1})) = \mathbf{U}(\neg Q^{t-1}; M(X, C^{t-1}))$, therefore our objective function for node selection is as follows:

$$\mathbf{x}_j^* = \underset{\mathbf{x}_j \in \neg Q^{t-1}}{\operatorname{argmax}} \mathbf{U}(\neg Q^{t-1}; M(X, C^{t-1})) - \mathbf{U}(\neg (Q')^{t-1}; M(X, (C')^{t-1})) \ , \quad (1)$$

where $(Q')^{t-1} = Q^{t-1} \bigcup \{\mathbf{x}_j\}$ and $(C')^{t-1}$ is the clustering result based on $(Q')^{t-1}$.

In order to solve Equation 1 and obtain node x_j, we make two assumptions: first, $(C')^{t-1}$ and C^{t-1} are sufficiently similar that,

$$\mathbf{U}(\neg (Q')^{t-1}; M(X(C')^{t-1})) \approx \mathbf{U}(\neg (Q')^{t-1}; M(X, C^{t-1})) \quad (2)$$

This assumption is reasonable because each iteration yields only a small number of localized constraints, so we expect that generally only a small subset of the clustering results is altered. The objective function thus becomes:

$$\begin{aligned} & \mathbf{U}(\neg Q^{t-1}; M(X, C^{t-1})) - \mathbf{U}(\neg (Q')^{t-1}; M(X, (C')^{t-1})) \\ \approx \quad & \mathbf{U}(\neg Q^{t-1}; M(X, C^{t-1})) - \mathbf{U}(\neg (Q')^{t-1}; M(X, C^{t-1})) \quad (3) \end{aligned}$$

Second, similar to the strong naive Bayes assumption (which is oversimplified but works quite well in many complex real-world situations) we assume node x_i is conditionally independent of every other node x_k for $i \neq k$. Thus we can infer that:

$$\begin{aligned} & \mathbf{U}(\neg Q^{t-1}; M(X, C^{t-1})) - \mathbf{U}(\neg (Q')^{t-1}; M(X, C^{t-1})) \\ = & \sum_{x_i \in \neg Q^{t-1}} \mathbf{U}(\{x_i\}; M(X, C^{t-1})) - \sum_{x_k \in \neg Q'^{t-1}} \mathbf{U}(\{x_k\}; M(X, C^{t-1})) \\ = \ & \mathbf{U}(\{x_j\}; M(X, C^{t-1})) \quad (4) \end{aligned}$$

Therefore, the objective function can be approximated as:

$$\mathbf{x}_j^* = \underset{\mathbf{x}_j \in \neg Q^{t-1}}{\operatorname{argmax}} U(\{x_j\}; M(X, C^{t-1})) \quad (5)$$

So the problem is transferred into finding the most uncertain node in the unqueried node set based on the current uncertainty model $M(X, C^{t-1})$. Next, we propose two definitions/models for node uncertainty, which we can then use to actively select new nodes and associated pair constraints. These two models represent, respectively, a local nonparametric and a global parametric view of uncertainty.

Local nonparametric structure model for node uncertainty. Sparse graphs are a widely used and effective way to describe the structure of data in many machine learning algorithms. Here we adopt a $\mathcal{K}$NN graph to represent the entire dataset, with the local structure of each node described by $\mathcal{K}$ edges.

Now define a "good" edge in the $\mathcal{K}$NN graph as an edge between two points in the same cluster. If the $\mathcal{K}$NN graph consists of all "good" edges then the output of a clustering algorithm run on the graph should match the $\mathcal{K}$NN graph connectivity. Alternately, in a graph with many "bad" edges (such as most graphs generated from real data), the clustering result usually does *not* obey the $\mathcal{K}$NN graph connectivity, and the graph will thus contain nodes whose neighbors are assigned to different clusters. Thus, uncertain nodes should be identifiable by the presence of "bad" edges linked to them.

Based on this observation, we define a local uncertainty model based on the degree of cluster assignment disagreement between a node and its neighbors. Thus, $M_L(X, C^{t-1})$ is based on the local structure of the data in the feature space and the clustering result in $t-1^{th}$ iteration. We calculate the uncertainty of a node $\mathbf{x}_j$ using:

$$\mathbf{U}(\{\mathbf{x}_j\}; M_L(X, C^{t-1})) = 1 - \frac{\#\{c_z = c_j, z \in N_j\}}{\#\{N_j\}} \ , \quad (6)$$

where $\frac{\#\{c_z = c_j, z \in N_j\}}{\#\{N_j\}}$ is the ratio of neighbors of node j in the graph that are assigned to same cluster as node $\mathbf{x}_j$ during clustering (notation c_z denotes the cluster index of neighbor z). If $\mathbf{U}(\{\mathbf{x}_j\}; M_L(X, C^{t-1}))$ is high, the cluster relationship between this node and its neighbors disagrees strongly with the local structure, so this node is highly uncertain.

We note that the distance measure and $\mathcal{K}$ value used to compute the $\mathcal{K}$NN graph do effect this uncertainty formulation, and serve as inputs to the algorithm. In our experiments, we select the commonly used Mahalanobis distance whose metric matrix is equal to the inverse covariance matrix of the data points, and $\mathcal{K} = 10$.

Global parametric model for node uncertainty. Most semi-supervised clustering methods learn a modified similarity kernel matrix as part of the clustering algorithm. Here, we define a new representation for the data by computing the k largest eigenvectors of the modified kernel matrix and using them as the features in a new data space. This functions similarly to the spectral eigenmap from input dimensionality d to dimensionality $k \leq d$:

$$\phi : \mathbf{x}_i \in \mathbb{R}^d \rightarrow \mathbb{R}^\infty \rightarrow \mathbb{R}^k \ . \quad (7)$$

$\mathbf{x}_i \in \mathbb{R}^d \rightarrow \mathbb{R}^\infty$ is accomplished using the kernel trick (with the kernel learned via semi-supervised clustering) to project the data from the original feature space to an infinitely high dimensional space. In the high dimensional space, we assume data nodes lie on a low dimensional manifold and each cluster is represented by a Gaussian.

Eigendecomposition then gives us $\mathbb{R}^\infty \rightarrow \mathbb{R}^k$, projecting the data from the high dimensional feature space back to a low dimensional representation. We can consider this projection process a marginalization over most of the infinite number of dimensions. In the ideal case, this projection should retain the distribution of data nodes, each cluster can thus be accurately represented in the low dimensional space by a Gaussian, and the whole dataset can be represented by a global Gaussian mixture model (GMM):

$$p(\mathbf{x}_i | \{\alpha_z\}, \{\mu_z\}, \{\Sigma_z\}) = \sum_{z=1}^{k} \alpha_z \mathcal{N}(\mathbf{x}_i; \mu_z, \Sigma_z) \ , \quad (8)$$

where $\{\alpha_z\}$ are the mixing weights and $(\{\mu_z\}, \{\Sigma_z\})$ are the component Gaussian parameters. Using EM, it is easy to estimate the parameters for the GMM and obtain the probability of each data point given each cluster z: $p(z|\mathbf{x}_i) = \frac{\alpha_z \mathcal{N}(\mathbf{x}_i;\mu_z,\Sigma_z)}{\sum_j^k \alpha_j \mathcal{N}(\mathbf{x}_i;\mu_j,\Sigma_j)}$. We can then define an uncertainty model $M_G(X, C^{t-1})$ based on the entropy of the $p(z|\mathbf{x}_i)$ distribution for a point, yielding an uncertainty measure:

$$\mathbf{U}(\{\mathbf{x}_j\}; M_G(X, C^{t-1})) = \sum_{z=1}^{k} p(z|\mathbf{x}_j) \log p(z|\mathbf{x}_j) \ , \quad (9)$$

where C^{t-1} implicitly determines this node uncertainty function via the learned kernel matrix.

Querying pairs based on the selected node. After finding the most uncertain node, we obtain pairs to query using the node sets NS^{t-1} as follows.

First, for each node set X_i, choose the single node within the set which is closest to the selected node $\mathbf{x}_j$: $\mathbf{x}_l = \text{argmin}_{\mathbf{x}_l \in X_i} Dist(\mathbf{x}_j, \mathbf{x}_l)$ and record this node and distance value.

Second, since there are k_{t-1} node sets, we will have recorded k_{t-1} nodes and distance values, so sort the nodes based on their corresponding distances. Now, in order of ascending distance, query the oracle on the relation between the selected node $\mathbf{x}_j$ and $\mathbf{x}_l$ until we find a must-link connection, then add $\mathbf{x}_j$ into the node set. If all of the relations are cannot-link, we create a new node set $X_{k_{t-1}+1}$ and add $\mathbf{x}_j$ to it. This new node set $X_{k_{t-1}+1}$ is then added to NS^{t-1} to obtain NS^t, and Q^{t-1} is correspondingly updated to Q^t. Since the relation between the new node and all node sets in NS^t is known, we can generate new pairwise constraints between the selected node $\mathbf{x}_j$ and all nodes in Q^t.

2.4 Semi-supervised clustering

When new pairwise constraints are obtained, we add them to the original constraint set. We then run the semi-supervised spectral clustering method on the new constraint set. There are a number of possible candidate methods for semi-supervised clustering, here we choose the two spectral clustering-based algorithms described below, due to the power and generality of spectral methods.

Spectral Learning This method [14] is a simple, easily implemented spectral learning algorithm, which applies the constraints directly. Given the set of pairwise constraints, the algorithm directly modifies the affinity(similarity) matrix W, then performs spectral clustering on the modified affinity matrix. Specifically, the new affinity matrix is now defined as:

- For each pair of must-linked points (i, j) assign the values $W_{ij} = W_{ji} = 1$.
- For each pair of cannot-linked points (i, j) assign the value $W_{ij} = W_{ji} = 0$.
- Normalize $N = \frac{1}{d_{max}}(W + d_{max}I - D)$, where d_{max} is the maximum of summed rows of W, and I is the identity matrix.

The spectral clustering algorithm then proceeds normally.

Flexible Semi-supervised spectral clustering This method was proposed by [22]. Rather than applying the constraints directly, this method optimizes the following objective function:

$$\underset{u \in R^N}{\text{argmin}}\, u^T L u \text{ s.t. } u^T D u = \sum_i D_{ii}, u^T Q u \geq \alpha \ , \quad (10)$$

where L is the normalized graph Laplacian of the affinity matrix and D is the diagonal degree matrix. It is easy to see that the difference from spectral clustering is the new term $u^T Q u \geq \alpha$. Q is the constraint matrix such that $Q_{ij} = 1$ for must-link, $Q_{ij} = -1$ for cannot-link and $Q_{ij} = 0$ for unknown pairs. u is the cluster assignment vector. $u^T Q u$ can thus be considered as a measure of how well the pairwise constraints conform to the cluster assignment u. Since $u \in R^N$ is a vector, the method is limited into two-cluster cases.

In our experiments, we apply OCSNU in conjunction with our local nonparametric and global parametric uncertainty measures to the above two semi-supervised clustering algorithms, and the results show the superiority and generality of our methods.

3 Experiments

3.1 Dataset

We evaluate our proposed framework and uncertainty measures on UCI machine learning, digits [1], gene (Cho's [6] and image datasets. The image data is a subset of the Caltech-101 [8], with images represented by image gist features (reduced to 100 dimensions via PCA). More details in Table 1. All results are averaged over 30 runs.

Table 1. UCI Datasets, DIGITS, GENE and IMAGE Datasets

Name	$\#Classes$	$\#Instances$	$\#Features$
Balance	3	625	4
Bupa	2	345	6
Diab	2	768	8
Sonar	2	208	60
Wine	3	178	13
Semeion digits	10	1593	256
Multiple features digits	10	2000	649
Cho	5	386	16
Caltech 5-Class	5	307	100

3.2 Evaluation protocols

We evaluate all cluster solutions via two commonly used cluster evaluation metrics: Rand Index [18] and V-Measure [19].

Rand Index. To measure the performance, we first adopt the well-known Rand Index, defined by:

$$\text{Accuracy} = \sum_{i>j} \frac{1\{1\{c_i = c_j\} = 1\{\hat{c}_i = \hat{c}_j\}\}}{n(n-1)/2}$$

where $1\{\cdot\}$ is the indicator function that outputs 1 when the input is true and 0 otherwise. c_i and $\hat{c}_i$ are the true cluster membership and the predicted cluster membership of the ith data point, respectively. n is the number of data points in the dataset.

V-Measure. We also employ the well-known V-Measure metric, which defines entropy-based measures for the completeness and homogeneity of the clustering results, and computes the harmonic mean of the two (in our case, we weight both measures equally). It is defined as follow:

$$V_\beta = \frac{(1+\beta) * h * c}{(\beta * h) + c} \quad (11)$$

where h is homogeneity and c is completeness.

3.3 Baseline and state of the art methods

To evaluate our active clustering framework and the two different selection strategies, we compare our method with the following set of methods, including a baseline and multiple the state of the art[3]:

[3] All selection algorithm code was downloaded from authors' websites except FFQS, which we reimplemented.

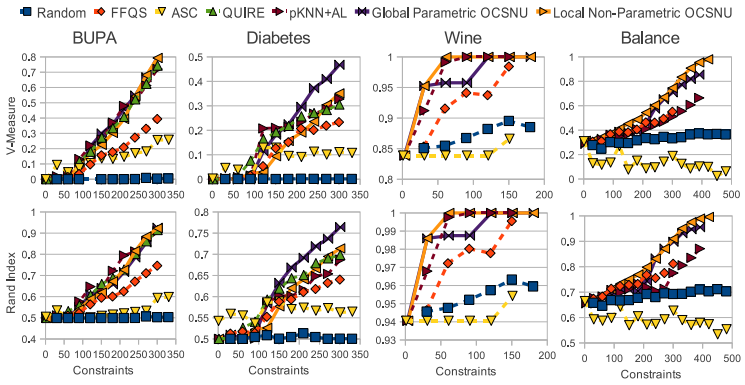

Figure 2. Rand Index and V-measure accuracy (vertical axes) with increasing pair relation queries on UCI datasets using spectral learning. *Best viewed in color.*

- **Random**: a baseline in which constraints are randomly sampled from the available pool.
- **FFQS** [2]: this method uses the farthest-first strategy to explore the data, then queries points randomly against the discovered node sets.
- **ASC** [21]: a pair-based method that queries pairs that will yield the maximum reduction in expected pair value error. In the original paper it is used in conjunction with flexible spectral clustering, and thus only applicable to two-class problems, but the active selection method itself can be applied to multiclass cases as well.
- **QUIRE** [12]: this is binary-only active learning method that computes node uncertainty based on the informativeness and representativeness of each node. We use our OCSNU framework to generate the requested node labels from pairwise queries.
- **pKNN+AL** [13]: this is is a minmax-based multi-class active learning method. Again, we use our OCSNU framework to translate node label requests into pairwise constraint queries.
- **Global Parametric OCSNU**: this is our proposed framework, using our global parametric node uncertainty measure to select nodes.
- **Local Nonparametric OCSNU**: this is our proposed framework, using our local nonparametric node uncertainty measure to select nodes.

All the above methods select constraints and feed the queried pair constraints to the two semi-supervised clustering methods we introduced in Section 2.4.

3.4 Results

Results on UCI dataset. We compared on 5 different UCI datasets, 3 binary and 2 multiclass.

Figure 2 shows the accuracy of different active selection methods with varying numbers of constraints when using spectral learning as the semi-supervised clustering algorithm. The first row of the figure shows V-measure values and the second Rand Index. Both metrics yield the same conclusion: our two methods are competitive in the two-cluster case and clearly superior for multicluster (wine and balance) problems. In particular, our local nonparametric uncertainty method is the best or tied for best on all but the Diabetes dataset, where it is beaten out by our global parametric method.

Figure 3 presents the results for various active selection methods in conjunction with the flexible semi-supervised spectral clustering method. As this is a binary clustering method, only 2-class datasets are used. Local nonparametric OCSNU still achieves competitive results on the BUPA and diabetes sets, and is clearly the best on sonar, though the global parametric uncertainty measure performs noticeably worse here. Interestingly, the different active selection methods are all much closer to each other in performance with this clustering method. For BUPA in particular, the choice of active method seems to have little effect on the results.

Results on Gene, Digits, images datasets. Figure 4 shows Rand Index and V-measure results for active clustering on the gene, image, digits datasets. Local nonparametric OCSNU performs particularly well here, most notably on the Caltech and Semeion datasets. We note that the robustness of the local method among all data tested may be partially attributable to the natural synergy between it and $\mathcal{K}$NN graph-based clustering methods, of which spectral clustering is a prime example. By comparison, the global parametric uncertainty measurement appears to be less robust, performing well on Caltech and Cho, but very poorly on the digits data. This is likely due to that fact that the parametric model underlying our global method may itself be a poor fit for some datasets.

Also notable is that the multiple feature digits dataset is the only case where the V-measure and Rand index metrics differ significantly in their relative assessment of the different selection algorithms, with V-measure scoring local OCSNU highest, and Rand index granting the best result to FFQS.

4 Conclusion

In this paper, we have considered the problem of active constraint selection for semi-supervised spectral clustering. Our paper makes two primary contributions: first, we describe a powerful general framework for online active semi-supervised clustering based on node uncertainty; second, we propose two methods for actively sampling constraints by transforming the pair-uncertainty problem into a node-uncertainty problem. We test our online active framework and selection criteria with two different semi-supervised clustering algorithms, against a number of existing active selection methods (including active clustering and active learning techniques), and find our method to be the most effective and robust of those surveyed.

In the future we hope to explore new node selection criteria. In particular, we wish to examine the possibility of a nonparametric global uncertainty measure, and of a compound uncertainty measure that considers both local and global structure.

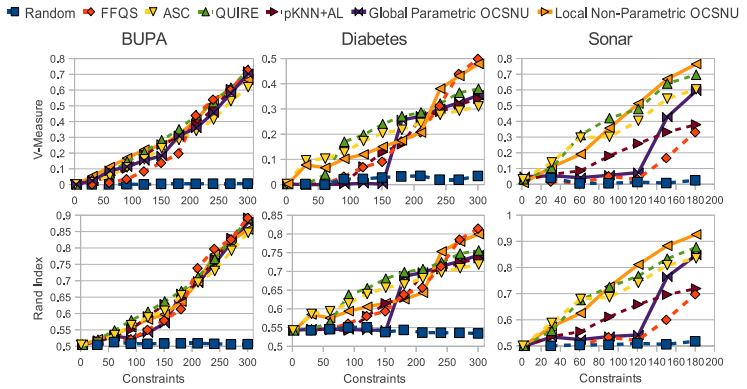

Figure 3. Rand Index and V-measure accuracy (vertical axes) with increasing pair relation queries on UCI datasets using flexible semi-supervised clustering. Since the method is limited to the two-class case, only two-class datsets were tested. *Best viewed in color.*

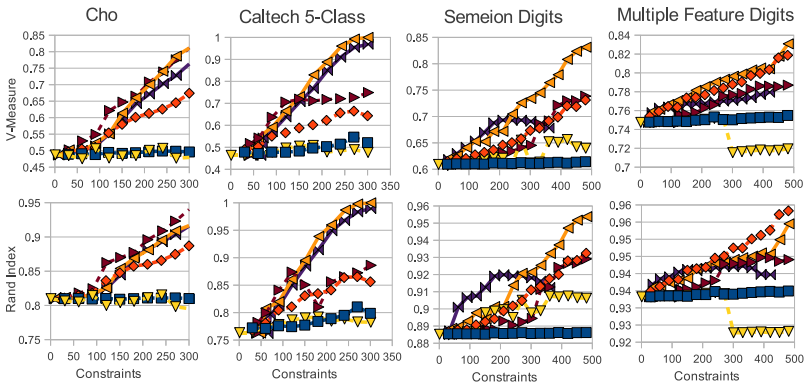

Figure 4. Rand Index and V-measure accuracy (vertical axes) with increasing pair relation queries on the gene, image and digits datasets using spectral learning. *Best viewed in color.*

REFERENCES

[1] A. Asuncion and D.J. Newman, 'UCI machine learning repository', (2007).

[2] S. Basu, A. Banerjee, and R.J. Mooney, 'Active semi-supervision for pairwise constrained clustering', in *ICDM*, pp. 333–344, (2004).

[3] S. Basu, M. Bilenko, and R.J. Mooney, 'A probabilistic framework for semi-supervised clustering', in *SIGKDD*, pp. 59–68. ACM, (2004).

[4] A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang, 'Agnostic active learning without constraints', *Arxiv preprint arXiv:1006.2588*, (2010).

[5] Ling Chen and Chengqi Zhang, 'Semi-supervised variable weighting for clustering', in *SDM*, pp. 862–871, (2011).

[6] R.J. Cho, M.J. Campbell, E.A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T.G. Wolfsberg, A.E. Gabrielian, D. Landsman, D.J. Lockhart, et al., 'A genome-wide transcriptional analysis of the mitotic cell cycle', *Molecular cell*, **2**(1), 65–73, (1998).

[7] I. Davidson, K. Wagstaff, and S. Basu, 'Measuring constraint-set utility for partitional clustering algorithms', *PKDD*, 115–126, (2006).

[8] L. Fei-Fei, R. Fergus, and P. Perona, 'One-shot learning of object categories', *PAMI*, 594–611, (2006).

[9] 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*, pp. 2161–2164. ACM, (2011).

[10] S.C.H. Hoi and R. Jin, 'Active kernel learning', in *ICML*, pp. 400–407. ACM, (2008).

[11] S.C.H. Hoi, R. Jin, and M.R. Lyu, 'Learning nonparametric kernel matrices from pairwise constraints', in *ICML*, pp. 361–368. ACM, (2007).

[12] S.J. Huang, R. Jin, and Z.H. Zhou, 'Active learning by querying informative and representative examples'. NIPS, (2010).

[13] P. Jain and A. Kapoor, 'Active learning for large multi-class problems', in *CVPR*, pp. 762–769. IEEE, (2009).

[14] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher, 'Spectral learning', in *IJCAI*. Stanford InfoLab, (2003).

[15] Z. Li and J. Liu, 'Constrained clustering by spectral kernel learning', in *ICCV*, pp. 421–427. IEEE, (2009).

[16] Z. Lu and M.A. Carreira-Perpinán, 'Constrained spectral clustering through affinity propagation', in *CVPR*, pp. 1–8. IEEE, (2008).

[17] P.K. Mallapragada, R. Jin, and A.K. Jain, 'Active query selection for semi-supervised clustering', in *ICPR*, pp. 1–4. IEEE, (2008).

[18] W.M. Rand, 'Objective criteria for the evaluation of clustering methods', *JASA*, 846–850, (1971).

[19] A. Rosenberg and J. Hirschberg, 'V-measure: A conditional entropy-based external cluster evaluation measure.', in *EMNLP-CoNLL'07*, pp. 410–420, (2007).

[20] B. Settles, 'Active learning literature survey', Technical report, (2010). (unpublished report).

[21] X. Wang and I. Davidson, 'Active Spectral Clustering', in *ICDM*, (2010).

[22] X. Wang and I. Davidson, 'Flexible constrained spectral clustering', in *SIGKDD*, pp. 563–572. ACM, (2010).

[23] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell, 'Distance metric learning with application to clustering with side-information', *NIPS*, 521–528, (2003).

[24] Q. Xu, M. Desjardins, and K. Wagstaff, 'Active constrained clustering by examining spectral eigenvectors', in *Discovery Science*, pp. 294–307. Springer, (2005).