Leveraging Union of Subspace Structure to Improve Constrained Clustering

John Lipor [1] **Laura Balzano** [1]

Abstract

Many clustering problems in computer vision and other contexts are also classification problems, where each cluster shares a meaningful label. Subspace clustering algorithms in particular are often applied to problems that fit this description, for example with face images or handwritten digits. While it is straightforward to request human input on these datasets, our goal is to reduce this input as much as possible. We present a pairwise-constrained clustering algorithm that actively selects queries based on the union-of-subspaces model. The central step of the algorithm is in querying *points of minimum margin* between estimated subspaces; analogous to classifier margin, these lie near the decision boundary. We prove that points lying near the intersection of subspaces are points with low margin. Our procedure can be used after any subspace clustering algorithm that outputs an affinity matrix. We demonstrate on several datasets that our algorithm drives the clustering error down considerably faster than the state-of-the-art active query algorithms on datasets with subspace structure and is competitive on other datasets.

1. Introduction

The union of subspaces (UoS) model, in which data vectors lie near one of several subspaces, has been used actively in the computer vision community on datasets ranging from images of objects under various lighting conditions (Basri & Jacobs, 2003) to visual surveillance tasks (Oliver et al., 2000). The recent textbook (Vidal et al., 2016) includes a number of useful applications for this model, including lossy image compression, clustering of face images under different lighting conditions, and video segmentation. Subspace clustering algorithms utilize the UoS model to cluster data vectors and estimate the underlying subspaces, achieving excellent performance on a variety of real datasets. However, as we will show in Section 4, even oracle UoS classifiers do not achieve perfect clustering on these datasets. While current algorithms for subspace clustering are unsupervised, in many cases a human could provide relevant information in the form of pairwise constraints between points, *e.g.,* answering whether two images are of the same person or whether two objects are the same.

The incorporation of pairwise constraints into clustering algorithms is known as pairwise-constrained clustering (PCC). PCC algorithms use supervision in the form of *must-link* and *cannot-link* constraints by ensuring that points with must-link constraints are clustered together and points with cannot-link constraints are clustered apart. In (Davidson et al., 2006), the authors investigate the phenomenon that incorporating poorly-chosen constraints can lead to an increase in clustering error, rather than a decrease as one would expect from additional label information. This is because points constrained to be in the same cluster that are otherwise dissimilar can confound the constrained clustering algorithm. For this reason, researchers have turned to *active* query selection methods, in which constraints are intelligently selected based on a number of heuristics. These algorithms perform well across a number of datasets but do not take advantage of any known structure in the data. In the case where data lie on a union of subspaces, one would hope that knowledge of the underlying geometry could give hints as to which points are likely to be clustered incorrectly.

Let $\mathcal{X} = \left\{x_i \in \mathbb{R}^D\right\}_{i=1}^N$ be a set of data points lying near a union of K linear subspaces of the ambient space. We denote the subspaces by $\{\mathcal{S}_k\}_{k=1}^K$, each having dimension d_k. An example union of subspaces is shown in Fig. 1, where $d_1 = 2$, $d_2 = d_3 = 1$. The goal of subspace clustering algorithms has traditionally been to cluster the points in $\mathcal{X}$ according to their nearest subspace without any supervised input. We turn this around and ask whether this model is useful for active clustering, where we request a very small number of intelligently selected labels. A key observation when considering data well-modeled by a union of subspaces is that uncertain points will be ones lying equally distant to multiple subspaces. Using a novel definition of margin tailored for the union of subspaces model, we incorporate this observation into an active subspace clustering

[1]Department of Electrical and Computer Engineering, University Michigan, Ann Arbor, MI, USA. Correspondence to: John Lipor <lipor@umich.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

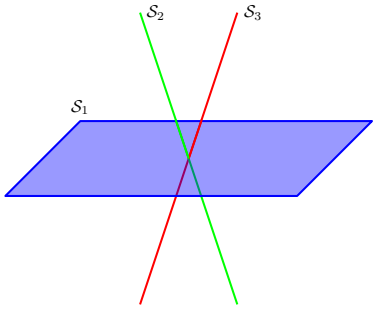

Figure 1. Example union of $K = 3$ subspaces of dimensions $d_1 = 2$, $d_2 = 1$, and $d_3 = 1$.

algorithm.

Our contributions are as follows. We introduce a novel algorithm for pairwise constrained clustering that leverages UoS structure in the data. A key step in our algorithm is choosing points of *minimum margin*, *i.e.,* those lying near a decision boundary between subspaces. We define a notion of margin for the UoS model and provide theoretical insight as to why points of minimum margin are likely to be misclustered by unsupervised algorithms. We show through extensive experimental results that when the data lie near a union of subspaces, our method drastically outperforms existing PCC algorithms, requiring far fewer queries to achieve perfect clustering. Our datasets range in dimension from 256-2016, number of data points from 320-9298, and number of subspaces from 5-100. On ten MNIST digits with a modest number of queries, we get 5% classification error with only 500 pairwise queries compared to about 20% error for current state-of-the-art PCC algorithms and 35% for unsupervised algorithms. We also achieve 0% classification error on the full Yale, COIL, and USPS datasets with a small fraction of the number of queries needed by competing algorithms. In datasets where we do not expect subspace structure, our algorithm still achieves competitive performance. Further, our algorithm is agnostic to the input subspace clustering algorithm and can therefore take advantage of any future algorithmic advances for subspace clustering.

2. Related Work

A survey of recently developed subspace clustering algorithms can be found in (Vidal, 2011) and the textbook (Vidal et al., 2016). In these and more recent work, clustering algorithms that employ spectral methods achieve the best performance on most datasets. Notable examples of such algorithms include Sparse Subspace Clustering (SSC) (Elhamifar & Vidal, 2013) and its extensions (You et al., 2016b;a), Low-Rank Representation (LRR) (Liu et al., 2010), Thresholded Subspace Clustering (TSC) (Heckel & Bölcskei, 2015), and Greedy Subspace Clustering (GSC) (Park et al., 2014). Many recent algorithms exist with both strong theoretical guarantees and empirical performance, and a full review of all approaches is beyond the scope of this work. However, the core element of all recent algorithms lies in the formation of the affinity matrix, after which spectral clustering is performed to obtain label estimates. In SSC, the affinity matrix is formed via a series of ℓ_1-penalized regressions. LRR uses a similar cost function but penalizes the nuclear norm instead of the ℓ_1. TSC thresholds the spherical distance between points, and GSC works by successively (greedily) building subspaces from points likely to lie in the same subspace. Of these methods, variants of SSC achieve the best overall performance on benchmark datasets and has the strongest theoretical guarantees, which were introduced in (Elhamifar & Vidal, 2013) and strengthened in numerous recent works (Soltanolkotabi & Candes, 2012; 2014; Wang & Xu, 2013; Wang et al., 2016). While the development of efficient algorithms with stronger guarantees has received a great deal of attention, very little attention has been paid to the question of what to do about data that cannot be correctly clustered. Thus, when reducing clustering error to zero (or near zero) is a priority, users must look beyond unsupervised subspace clustering algorithms to alternative methods. One such method is to request some supervised input in the form of pairwise constraints, leading to the study of pairwise-constrained clustering (PCC).

PCC algorithms work by incorporating *must-link* and *cannot-link* constraints between points, where points with must-link constraints are forced (or encouraged in the case of spectral clustering) to be clustered together, and points with cannot-link constraints are forced to be in separate clusters. In many cases, these constraints can be provided by a human labeler. For example, in (Biswas & Jacobs, 2014), the authors perform experiments where comparisons between human faces are provided by users of Amazon Mechanical Turk with an error rate of 1.2%. Similarly, for subspace clustering datasets such as Yale B and MNIST, a human could easily answer questions such as, "Are these two faces the same person?" and "Are these two images the same number?" An early example of PCC is found in (Wagstaff et al., 2001), where the authors modify the K-means cost function to incorporate such constraints. In (Basu et al., 2004), the authors utilize active methods to initialize K-means in an intelligent "EXPLORE" phase, during which neighborhoods of must-linked points are built up. After this phase, new points are queried against representatives from each neighborhood until a must-link is obtained. A similar explore phase is used in (Mallapragada et al., 2008), after which a min-max approach is used to select the most uncertain sample. Early work on constrained spectral clustering appears in (Xu et al., 2005; Wang & Davidson, 2010), in which spectral clustering is improved by examining the

eigenvectors of the affinity matrix in order to determine the most informative points. However, these methods are limited to the case of two clusters and therefore impractical in many cases.

More recently, the authors in (Xiong et al., 2016; Biswas & Jacobs, 2014) improve constrained clustering by modeling which points will be most informative given the current clustering, with state-of-the-art results achieved on numerous datasets by the algorithm in (Xiong et al., 2016), referred to as Uncertainty Reducing Active Spectral Clustering (URASC). URASC works by maintaining a set of *certain sets*, whereby points in the same certain set are must-linked and points in different certain sets are cannot-linked. A test point x_T is selected via an uncertainty-reduction model motivated by matrix perturbation theory, after which queries are presented in an intelligent manner until x_T is either matched with an existing certain set or placed in its own new certain set. In practice (Xiong, 2016), the certain sets are initialized using the EXPLORE algorithm of (Basu et al., 2004).

While we are certainly not the first to consider actively selecting labels to improve clustering performance, to the best of our knowledge we are the first to do so with structured clusters. Structure within and between data clusters is often leveraged for unsupervised clustering (Wright et al., 2009), and that structure is also leveraged for adaptive sampling of the structured signals themselves (*e.g.,* see previous work on sparse (Haupt et al., 2011; Indyk et al., 2011), structured sparse (Soni & Haupt, 2014), and low rank signals (Krishnamurthy & Singh, 2013)). This paper emphasizes the power of that structure for reducing the number of required labels in an active learning algorithm as opposed to reducing the number of samples of the signal itself, and points to exciting open questions regarding the tradeoff between signal measurements and query requirements in semi-supervised clustering.

3. UoS-Based Pairwise-Constrained Clustering

Recall that $\mathcal{X} = \left\{x_i \in \mathbb{R}^D\right\}_{i=1}^N$ is a set of data points lying on a union of K subspaces $\{\mathcal{S}_k\}_{k=1}^K$, each having dimension d. In this work, we assume all subspaces have the same dimension, but it is possible to extend our algorithm to deal with non-uniform dimensions. The goal is to cluster the data points according to this generative model, *i.e.,* assigning each data point to its (unknown) subspace. In this section we describe our algorithm, which actively selects pairwise constraints in order to improve clustering accuracy. The key step is choosing an informative query test point, which we do using a novel notion of *minimum subspace margin*.

Denote the true clustering of a point $x \in \mathcal{X}$ by $C(x)$. Let the output of a clustering algorithm (such as SSC) be an affinity/similarity matrix A and a set of label estimates $\left\{\hat{C}(x_i)\right\}_{i=1}^N$. These are the inputs to our algorithm. The high-level operation of our algorithm is as follows. To initialize, we build a set of certain sets $\mathcal{Z}$ using an EXPLORE-like algorithm similar to that of (Basu et al., 2004). Certain sets are in some sense equivalent to labels in that points within a certain set belong to the same cluster and points across certain sets belong to different clusters. Following this, the following steps are repeated until a maximum number of queries has been made:

1. **Spectral Clustering:** Obtain label estimates via spectral clustering.
2. **PCA on each cluster:** Obtain a low-dimensional subspace estimate from points currently sharing the same estimated cluster label.
3. **Select Test Point:** Obtain a test point x_T using subspace margin with respect to the just estimated subspaces.
4. **Assign x_T to Certain Set:** Query the human to compare the test point with representatives from certain sets until a must-link is found or all certain sets have been queried, in which case the test point becomes its own certain set.
5. **Impute Label Information:** Certain sets are used to impute must-link and cannot-link values in the affinity matrix.

We refer to our algorithm as SUPERPAC (SUbsPace clustERing with Pairwise Active Constraints). A diagram of the algorithm is given in Fig. 2, and we outline each of these steps below and provide pseudocode in Algorithm 1.

3.1. Sample Selection via Margin

Min-margin points have been studied extensively in active learning; intuitively, these are points that lie near the decision boundary of the current classifier. In (Settles, 2012), the author notes that actively querying points of minimum margin (as opposed to maximum entropy or minimum confidence) is an appropriate choice for reducing classification error. In (Wang & Singh, 2016), the authors present a margin-based binary classification algorithm that achieves an optimal rate of convergence (within a logarithmic factor).

In this section, we define a novel notion of margin for the UoS model and provide theoretical insight as to why points of minimum margin are likely to be misclustered. For a subspace $\mathcal{S}_k$ with orthonormal basis U_k, let the distance of a point to that subspace be $\text{dist}(x, \mathcal{S}_k) = \min_{y \in \mathcal{S}_k} \|x - y\|_2 = \left\|x - U_k U_k^T x\right\|_2$. Let $k^* = \arg\min_{k \in [K]} \text{dist}(x, \mathcal{S}_k)$ be the index of the closest subspace, where $[K] = \{1, 2, \cdots, K\}$. Then the subspace

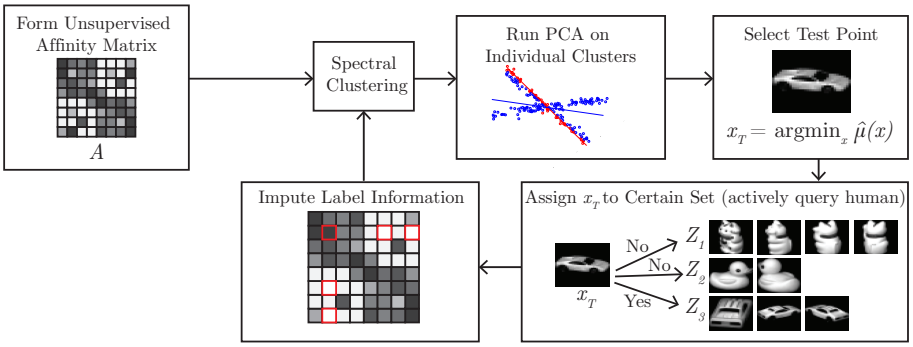

Figure 2. Diagram of SUPERPAC algorithm for pairwise constrained clustering.

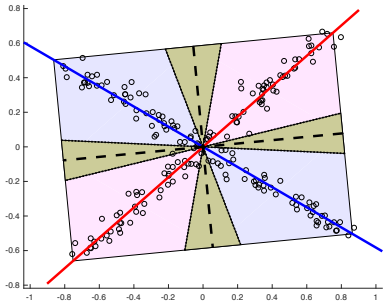

Figure 3. Illustration of subspace margin. The blue and red lines are the generative subspaces, with corresponding disjoint decision regions. The yellow-green color shows the region within some margin of the decision boundary, given by the dotted lines.

margin of a point $x \in \mathcal{X}$ is the ratio of closest and second closest subspaces, defined as

$$\hat{\mu}(x) = 1 - \max_{j \neq k^*, j \in [K]} \frac{\text{dist}(x, S_{k^*})}{\text{dist}(x, S_j)}. \tag{1}$$

The point of minimum margin is then defined as $\arg\min_{x \in \mathcal{X}} \hat{\mu}(x)$. Note that the fraction is a value in $[0, 1]$, where the a value of 0 implies that the point x is equidistant to its two closest subspaces. This notion is illustrated in Figure 3, where the yellow-green color shows the region within some margin of the decision boundary.

In the following theorem, we show that points lying near the intersection of subspaces are included among those of minimum margin with high probability. This method of point selection is then motivated by the fact that the difficult points to cluster are those lying near the intersection of subspaces [12]. Further, theory for SSC ([11],[15]) shows that problematic points are those having large inner product with some or all directions in other subspaces. Subspace margin captures exactly this phenomenon.

Theorem 1. *Consider two d-dimensional subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. Let $y = x + n$, where $x \in \mathcal{S}_1$ and $n \sim \mathcal{N}(0, \sigma^2 I_D)$. Define*

$$\mu(y) = 1 - \frac{\text{dist}(y, \mathcal{S}_1)}{\text{dist}(y, \mathcal{S}_2)} .$$

Then

$$1 - \frac{(1+\varepsilon)\sqrt{\sigma^2(D-d)}}{(1-\varepsilon)\sqrt{\sigma^2(D-d) + \text{dist}(x, \mathcal{S}_2)^2}} \leq \mu(y)$$

and

$$\mu(y) \leq 1 - \frac{(1-\varepsilon)\sqrt{\sigma^2(D-d)}}{(1+\varepsilon)\sqrt{\sigma^2(D-d) + \text{dist}(x, \mathcal{S}_2)^2}},$$

with probability at least $1 - 4e^{-c\varepsilon^2(D-d)}$, where c is an absolute constant.

The proof is given in the supplementary material. Note that if $\text{dist}(y, \mathcal{S}_1) \leq \text{dist}(y, \mathcal{S}_2)$, then $\mu(y) = \hat{\mu}(y)$. In this case, Thm. 1 states that under the given noise model, points with small residual to the incorrect subspace (*i.e.,* points near the intersection of subspaces) will have small margin. These are exactly the points for which supervised label information will be most beneficial.

The statement of Thm. 1 allows us to quantify exactly how near a point must be to the intersection of two subspaces to be considered a point of minimum margin. Let $\phi_1 \leq \phi_2 \leq \cdots \leq \phi_d$ be the d principal angles[1] between $\mathcal{S}_1$ and $\mathcal{S}_2$. If the subspaces are very far apart, $\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$ is near 1, and if they are very close $\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$ is near zero. Note that, for any $x \in \mathcal{S}_1$,

$$\sin^2(\phi_1) \leq \text{dist}(x, \mathcal{S}_2)^2 \leq \sin^2(\phi_d) \ ;$$

that is, there are bounds on $\text{dist}(x, \mathcal{S}_2)$ depending on the relationship of the two subspaces. We also know that if x is drawn using isotropic Gaussian weights from a basis for $\mathcal{S}_1$, then

$$\mathbb{E}\left[\text{dist}(x, \mathcal{S}_2)^2\right] = \frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i) \ .$$

Given this, we might imagine that margin of the noisy points is a useful indicator of points near the intersection in a scenario where $\sin^2(\phi_1)$ is small but $\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$ is not,

[1] See (Golub & Loan, 2012) for a definition of principal angles.

e.g., when the subspaces have an intersection but are distant in other directions. With this in mind we state the following corollary, whose proof can be found in the supplementary material.

Corollary 1. *Suppose $x_1 \in \mathcal{S}_1$ is such that*

$$\operatorname{dist}(x_1, \mathcal{S}_2)^2 = \sin^2(\phi_1) + \delta \left(\frac{1}{d} \sum_{i=1}^{d} \sin^2(\phi_i) \right) \quad (2)$$

for some small $\delta \geq 0$; that is, x_1 is close to the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. Let x_2 be a random point in $\mathcal{S}_1$ generated as $x_2 = U_1 w$ where U_1 is a basis for $\mathcal{S}_1$ and $w \sim \mathcal{N}(0, \frac{1}{d} I_d)$. We observe $y_i = x_i + n_i$, where $n_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, 2$. If there exists $\tau > 1$ such that

$$\delta < \frac{5}{7} - \frac{1}{\tau}$$

and

$$\tau \left(\sin^2(\phi_1) + \frac{1}{6}\sigma^2 (D - d) \right) < \frac{1}{d} \sum_{i=1}^{d} \sin^2(\phi_i) \,, \quad (3)$$

that is, the average angle is sufficiently larger than the smallest angle, then

$$\mathbb{P}\{\mu(y_1) < \mu(y_2)\} \geq 1 - e^{-c\left(\frac{7}{100}\right)^2 ds} - 4e^{-c\left(\frac{1}{50}\right)^2 (D-d)}$$

where $\mu(y)$ is defined as in Thm. 1, c is an absolute constant, and $s = \frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$.

We make some remarks first to connect our results to other subspace distances that are often used. Perhaps the most intuitive form of subspace distance between that spanned by U_1 and U_2 is $\frac{1}{d}\|(I - U_1 U_1)^T U_2\|_F^2$; if the two subspaces are the same, the projection onto the orthogonal complement is zero; if they are orthogonal, we get the norm of U_2 alone, giving a distance of 1. This is equal to the more visually symmetric $1 - \frac{1}{d}\|U_1^T U_2\|_F^2$, another common distance. Further we note that, by the definition of principal angles (Golub & Loan, 2012),

$$1 - \frac{1}{d}\|U_1^T U_2\|_F^2 = 1 - \frac{1}{d}\sum_{i=1}^{d} \cos^2(\phi_i) = \frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i) \,.$$

From Equation (2), we see that the size of δ determines how close $x_1 \in \mathcal{S}_1$ is to $\mathcal{S}_2$; if $\delta = 0$, x_1 is as close to $\mathcal{S}_2$ as possible. For example, if $\phi_1 = 0$, the two subspaces intersect, and $\delta = 0$ implies that $x_1 \in \mathcal{S}_1 \cap \mathcal{S}_2$. Equation (3) captures the gap between average principal angle and the smallest principal angle. We conclude that if this gap is large enough and δ is small enough so that x_1 is close to $\mathcal{S}_2$, then the observed y_1 will have smaller margin than the average point in $\mathcal{S}_1$, even when observed with noise.

Algorithm 1 SUPERPAC

Input: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$: data, K: number of clusters, d: subspace dimension, A: affinity matrix, maxQueries: maximum number of pairwise comparisons
Estimate Labels: $\hat{C} \leftarrow$ SPECTRALCLUSTERING(A,K)
Initialize Certain Sets: Initialize $\mathcal{Z} = \{Z_1, \cdots, Z_{n_c}\}$ and numQueries via UOS-EXPLORE in supplementary material.
while numQueries $<$ maxQueries **do**
  PCA on Each Cluster: Solve

$$\mathcal{S}_k = \min_{U \in \mathbb{R}^{D \times d}} \sum_{i:\hat{C}(x_i)=k} \|x_i - UU' x_i\|^2 \,.$$

  Obtain Test Point: select $x_T \leftarrow \arg\min_{x \in \mathcal{X}} \hat{\mu}(x)$
  Assign x_T to Certain Set:

    Sort $\{Z_1, \cdots, Z_{n_c}\}$ in order of most likely must-link (via subspace residual for x_T), query x_T against representatives from Z_k until must-link constraint is found or $k = n_c$. If no must-link constraint is found, set $\mathcal{Z} \leftarrow \{Z_1, \cdots, Z_{n_c}, \{x_T\}\}$ and increment n_c.

  Impute Constraints: Set $A_{ij} = A_{ji} = 1$ for (x_i, x_j) in the same certain set and $A_{ij} = A_{ji} = 0$ for (x_i, x_j) in different certain sets (do not impute for points absent from certain sets).
  Estimate Labels: $\hat{C} \leftarrow$ SPECTRALCLUSTERING(A,K)
end while

For another perspective, consider that in the noiseless case, for $x_1, x_2 \in \mathcal{S}_1$, the condition $\operatorname{dist}(x_1, \mathcal{S}_2) < \operatorname{dist}(x_2, \mathcal{S}_2)$ is enough to guarantee that x_1 lies nearer to $\mathcal{S}_2$. Under the given additive noise model ($y_i = x_i + n_i$ for $i = 1, 2$) the gap between $\operatorname{dist}(x_1, \mathcal{S}_2)$ and $\operatorname{dist}(x_2, \mathcal{S}_2)$ must be larger by some factor depending on the noise level. After two applications of Thm. 1 and rearranging terms, we have that $\mu(y_1) < \mu(y_2)$ with high probability if

$$\beta \operatorname{dist}(x_2, \mathcal{S}_2)^2 - \operatorname{dist}(x_1, \mathcal{S}_2)^2 > (1 - \beta)\sigma^2(D - d). \quad (4)$$

where $\beta = ((1 - \varepsilon)/(1 + \varepsilon))^4$, a value near 1 for small ε. Equation (4) shows that the gap $\operatorname{dist}(x_2, \mathcal{S}_2)^2 - \operatorname{dist}(x_1, \mathcal{S}_2)^2$ must grow (approximately linearly) with the noise level σ^2. The relationship of this gap to the subspace distances is quantified by Corollary 1; plugging $\sin^2(\phi_1)$ from Equation (2) into Equation (3) and rearranging yields a statement of the form in Equation (4).

3.2. Pairwise Constrained Clustering with SUPERPAC

We now describe SUPERPAC in more detail, our algorithm for PCC when data lie near a union of subspaces, given in Algorithm 1. The algorithm begins by initializing a set of disjoint certain sets, an optional process described in the

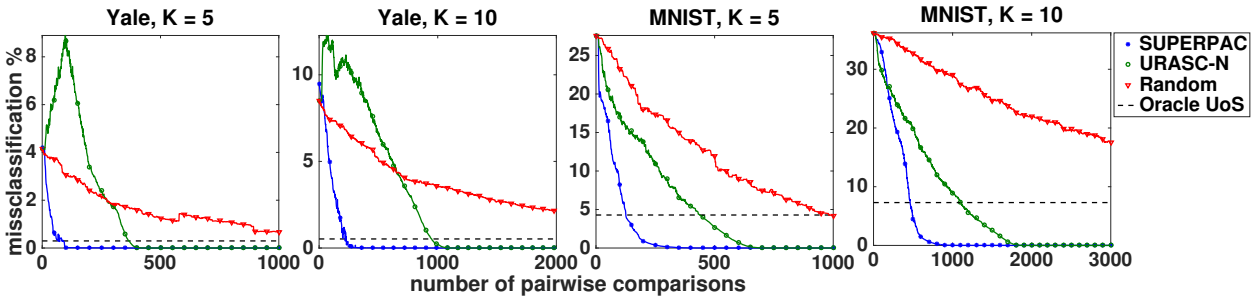

Figure 4. Misclassification rate for Yale B and MNIST datasets with many pairwise comparisons. Left-to-right: Yale B $K = 5$ (input from SSC), Yale B $K = 10$ (input from SSC), MNIST $K = 5$ (input from TSC), MNIST $K = 10$ (input from TSC).

supplementary material due to space constraints. Next our algorithm assigns the points most likely to be misclassified to certain sets by presenting a series of pairwise comparisons. Finally, we impute values onto the affinity matrix for all points in the certain sets and perform spectral clustering. The process is then repeated until the maximum number of pairwise comparisons has been reached.

Let x_T be the test point chosen as the min-margin point. Our goal is to assign x_T to a certain set using as the fewest number of queries possible. For each certain set Z_k, the representative x_k is chosen as the maximum-margin point within the set. Next, for each k, we let U_k be the d-dimensional PCA estimate of the matrix whose columns are the points $\left\{x \in \mathcal{X} : \hat{C}(x) = \hat{C}(x_k)\right\}$. We then query our test point x_T against the representatives x_k in order of residual $\left\|x_T - U_k U_k^T x_T\right\|_2$ (smallest first). If a must-link constraint is found, we place x_T in the corresponding certain set. Otherwise, we place x_T in its own certain set and update the number of certain sets. Pseudocode for the complete algorithm is given in Algorithm 1. As a technical note, we first normalize the input affinity matrix A so that the maximum value is 2. For must-link constraints, we impute a value of 1 in the affinity matrix, while for cannot-link constraints we impute a 0. The approach of imputing values in the affinity matrix is common in the literature but does not strictly enforce the constraints. Further, we found in our experiments that imputing the maximum value in the affinity matrix resulted in unstable results. Thus, users must be careful to not only choose the correct constraints as noted in (Basu et al., 2004), but to incorporate these constraints in a way that allows for robust clustering.

SUPERPAC can be thought of as an extension of ideas from PCC literature (Basu et al., 2004; Biswas & Jacobs, 2014; Xiong et al., 2016) to leverage prior knowledge about the underlying geometry of the data. For datasets such as Yale B and MNIST, the strong subspace structure makes Euclidean distance a poor proxy for similarity between points in the same cluster, leading to the superior performance of our algorithm demonstrated in the following sections. This structure does not exist in all datasets, in which case we do not expect our algorithm to outperform current PCC algorithms. The reader will note we made a choice to order the certain sets according to the UoS model; this is similar to the choice in (Xiong et al., 2016) to query according to similarity, where our notion of similarity here is based on subspace distances. We found this resulted in significant performance benefits, matching our intuition that points are clustered based on their nearest subspace. In contrast to (Biswas & Jacobs, 2014; Xiong et al., 2016), where the test point is chosen according to a global improvement metric, we choose test points according to their classification margin. In our experiments, we found subspace margin to be a strong indicator of which points are misclassified, meaning that our algorithm rapidly corrects the errors that occur as a result of unsupervised subspace clustering.

Finally, note that the use of certain sets relies on the assumption that the pairwise queries are answered correctly—an assumption that is common in the literature (Basu et al., 2004; Mallapragada et al., 2008; Xiong et al., 2016). However, in (Xiong et al., 2016), the authors demonstrate that an algorithm based on certain sets still yields significant improvements under a small error rate. The study of robustly incorporating noisy pairwise comparisons is an interesting topic for further study.

4. Experimental Results

We compare the performance of our method and the non-parametric version of the URASC algorithm (URASC-N)[2] over a variety of datasets. Note that while numerous PCC algorithms exist, URASC achieves both the best empirical results and computational complexity on a variety of datasets. We also compared with the methods from (Basu et al., 2004) and (Biswas & Jacobs, 2014) but found both to perform significanly worse than URASC on all datasets considered, with a far greater computational cost in the case

[2]In our experiments, the parametric version of URASC was found to be numerically unstable and did not have significantly different performance from URASC-N in the best cases.

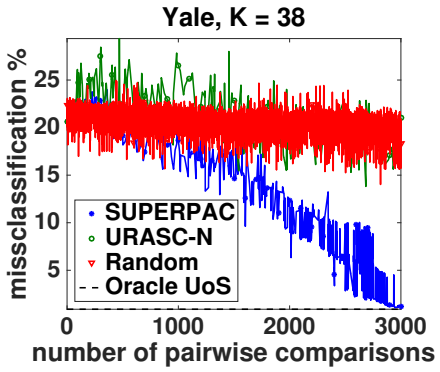

Figure 5. Misclassification rate versus number of pairwise comparisons for extended Yale face database B with $K = 38$ subjects. Input affinity matrix is taken from SSC-OMP.

of (Biswas & Jacobs, 2014). We use a maximum query budget of $2K$ for UoS-EXPLORE and EXPLORE. For completeness, we also compare to random constraints, in which queries are chosen uniformly at random from the set of unqueried pairs.

Finally, we compare against the oracle PCA classifier, which we now define. Let U_k be the d-dimensional PCA estimate of the points whose true label $C(x) = k$. Then the oracle label is $\hat{C}_o(x) = \arg\min_{k\in[K]} \left\|x - U_k U_k^T x\right\|_2$. This allows us to quantitatively capture the idea that, because the true classes are not perfectly low-rank, some points would not be clustered with the low-rank approximation of their own true cluster. In our experiments, we also compared with oracle robust PCA (Candes et al., 2011) implemented via the augmented Lagrange multiplier method (Lin et al., 2011) but did not find any improvement in classification error.

Datasets We consider five datasets commonly used as benchmarks in the subspace clustering literature[3], with a summary of the datasets and their relevant parameters are given in Table 1. The Yale B dataset consists of 64 images of size 192×168 of each of 38 different subjects under a variety of lighting conditions. For values of K less than 38, we follow the methodology of (Zhang et al., 2012) and perform clustering on 100 randomly selected subsets of size K. We choose $d = 9$ as is common in the literature (Elhamifar & Vidal, 2013; Heckel & Bölcskei, 2015). The MNIST handwritten digit database test dataset consists of 10,000 centered 28×28 pixel images of handwritten digits 0-9. We follow a similar methodology to the previous section and select 100 random subsets of size K, using subspace dimension $d = 3$ as in (Heckel & Bölcskei, 2015). The COIL-20 dataset (Nene et al., 1996b) consists of 72 images

[3]The validity of the UoS assumption for two of these datasets is investigated in (Elhamifar & Vidal, 2013; Heckel & Bölcskei, 2015).

Dataset	N	K	D	d
Yale	320-2432	5,10,38	2016	9
MNIST	500-1000	5,10	784	3
COIL-20	1440	20	1024	9
COIL-100	7200	100	1024	9
USPS	9298	10	256	15

Table 1. Datasets used for experiments with relevant parameters; N: total number of samples, K: number of clusters, D: ambient dimension, d: estimated subspace dimension.

of size 32×32 of each of 20 objects. The COIL-100 dataset (Nene et al., 1996a) contains 100 objects (distinct from the COIL-20 objects) of the same size and with the same number of images of each object. For both datasets, we use subspace dimension $d = 9$. Finally, we apply our algorithm to the USPS dataset provided by (Cai et al., 2011), which contains 9,298 *total* images of handwritten digits 0-9 of size 16×16 with roughly even label distribution. We again use subspace dimension $d = 9$.

Input Subspace Clustering Algorithms A major strength of our algorithm is that it is agnostic to the initial subspace clustering algorithm used to generate the input affinity matrix. To demonstrate this fact, we apply our algorithm with an input affinity matrix obtained from a variety of subspace clustering methods, summarized in Table 1. Note that some recent algorithms are not included in the simulations here. However, the simulations show that our algorithm works well with *any* initial clustering, and hence we expect similar results as new algorithms are developed.

Experimental Results Fig. 4 shows the clustering error versus the number of pairwise comparisons for the Yale and MNIST datasets. The input affinity matrix is obtained by running SSC for the Yale datset and by running TSC for the MNIST dataset. The figure clearly demonstrates the benefits of leveraging UoS structure in constrained clustering—in all cases, SUPERPAC requires roughly *half* the number of queries needed by URASC to achieve perfect clustering. For the Yale dataset with $K = 5$, roughly $2Kd$ queries are required to surpass oracle performance, and for $K = 10$ roughly $3Kd$ queries are required. Note that for the Yale dataset, the clustering error *increases* using URASC. This is due to the previously mentioned fact that imputing the wrong constraints can lead to worse clustering performance. For sufficiently many queries, the error decreases as expected. Fig. 5 shows the misclassification rate versus number of points for all $K = 38$ subjects of the Yale databse, with the input affinity matrix taken from SSC-OMP (You et al., 2016b). We space out the markers for clearer plots. In this case, URASC performs roughly the same as random query selection, while SUPERPAC performs significantly

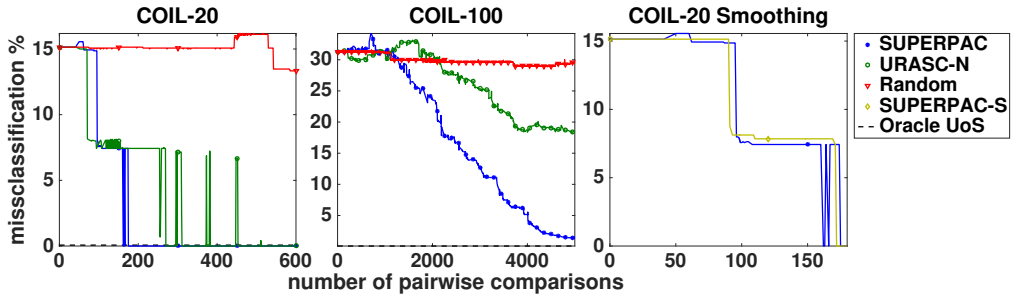

Figure 6. Misclassification rate versus number of pairwise comparisons for COIL-20 ($K = 20$) and COIL-100 ($K = 100$)) databases. Input affinity matrix is taken from EnSC. Rightmost plot shows proposed smoothing heuristic.

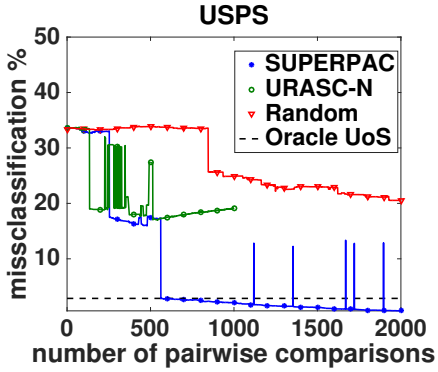

Figure 7. Misclassification rate versus number of pairwise comparisons for USPS dataset with $K = 10$ digits, 9,298 total samples. Input affinity matrix is taken from EnSC. URASC did not complete after 48 hours of run time.

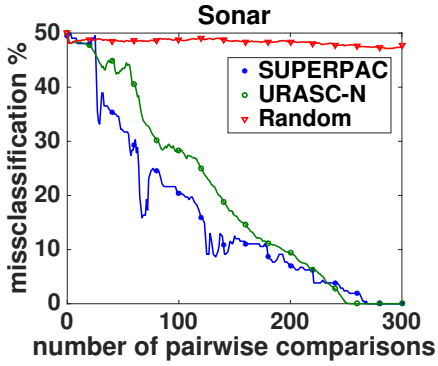

Figure 8. Misclassification rate for Sonar dataset from (Xiong et al., 2016), where there is not reason to believe the clusters have subspace structure. We are still very competitive with state-of-the-art.

better.

Fig. 6 demonstrates the continued superiority of our algorithm in the case where UoS structure exists. In the case of COIL-20, the clustering is sometimes unstable, alternating between roughly 0% and 7% clustering error for both active algorithms. This further demonstrates the observed phenomenon that spectral clustering is sensitive to small perturbations. To avoid this issue, we kept track of the K-subspaces cost function (see (Bradley & Mangasarian, 2000)) and ensured the cost decreased at every iteration. We refer to this added heuristic as SUPERPAC-S in the figure. The incorporation of this heuristic into our algorithm is a topic for further study.

Fig. 7 shows the resulting error on the USPS dataset, again indicating the superiority of our method. Note that N is large for this dataset, making spectral clustering computationally burdensome. Further, the computational complexity of URASC is dependent on N. As a result, URASC did not complete 2000 queries in 48 hours of run time when using 10 cores, so we compare to the result after completing only 1000 queries. Finally, in Fig. 8, we demonstrate that even on data without natural subspace structure, SUPERPAC performs competitively with URASC.

5. Conclusion

We have presented a method of selecting and incorporating pairwise constraints into subspace clustering that considers the underlying geometric structure of the problem. The union of subspaces model is often used in computer vision applications where it is possible to request input from human labelers in the form of pairwise constraints. We showed that labeling is often necessary for subspace classifiers to achieve a clustering error near zero; additionally, these constraints can be chosen intelligently to improve the clustering procedure overall and allow for perfect clustering with a modest number of requests for human input.

Developing techniques for handling noisy query responses will allow extension to undersampled or compressed data. One may assume that compressed data would be harder to distinguish, leading to noisier query responses. Finally, we saw that for datasets with different types of cluster structure, the structure assumptions of each algorithm had direct impact on performance; in the future we plan to additionally develop techniques for learning from unlabeled data whether the union of subspace model or a standard clustering approach is more appropriate.

Acknowledgements

This work was supported by NSF F031543-071159-GRFP and US ARO Grant W911NF1410634.

References

Basri, R. and Jacobs, D. Lambertian reflectance and linear subspaces. *IEEE TPAMI*, 25(2):218–233, February 2003.

Basu, Sugato, Banerjee, Arindam, and Mooney, Raymond J. Active semi-supervision for pairwise constrained clustering. In *Proc. SIAM Int. Conf. on Data Mining*, 2004.

Biswas, Arjit and Jacobs, David. Active image clustering with pairwise constraints from humans. *International Journal on Computer Vision*, 108:133–147, 2014.

Bradley, Paul S. and Mangasarian, Olvi L. k-Plane clustering. *Journal of Global Optimization*, 16:23–32, 2000.

Cai, Deng, He, Xiaofei, Han, Jiawei, and Huang, Thomas S. Graph regularized non-negative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

Candes, Emmanuel J., Li, Xiadong, Ma, Yi, and Wright, John. Robust principal component analysis? *Journal of the ACM*, 58(3), May 2011.

Davidson, Ian, Wagstaff, Kiri L., and Basu, Sugato. Measuring constraint-set utility for partitional clustering algorithms. In *Proc. European Conf. on Machine Learning and Prinicpals and Practice of Knowledge Discovery in Databases*, 2006.

Elhamifar, Ehsan and Vidal, Renee. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35:2765–2781, November 2013.

Golub, Gene and Loan, Charles Van. *Matrix Computations*. Johns Hopkins University Press, 2012.

Haupt, Jarvis, Castro, Rui M, and Nowak, Robert. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 57 (9):6222–6235, 2011.

Heckel, Reinhard and Bölcskei, Helmut. Robust subspace clustering via thresholding. *IEEE Trans. Inf. Theory*, 24 (11):6320–6342, 2015.

Indyk, Piotr, Price, Eric, and Woodruff, David P. On the power of adaptivity in sparse recovery. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pp. 285–294. IEEE, 2011.

Krishnamurthy, Akshay and Singh, Aarti. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, pp. 836–844, 2013.

Lin, Zhouchen, Chen, Minming, Wu, Leqin, and Ma, Yi. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in Neural Information Processing Systems*, 2011.

Liu, Guangcan, Lin, Zhouchen, and Yu, Yong. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 663–670, 2010.

Mallapragada, Pavan Kumar, Jin, Rong, and Jain, Anil K. Active query selection for semi-supervised clustering. In *Proc. Int. Conf. on Pattern Recognition*, 2008.

Nene, S. A., Nayar, S. K., and Murase, H. Columbia object image library (COIL-100). Technical report, Columbia University, 1996a.

Nene, S. A., Nayar, S. K., and Murase, H. Columbia object image library (COIL-20). Technical report, Columbia University, 1996b.

Oliver, N.M., Rosario, B., and Pentland, A.P. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

Park, Dohyung, Caramanis, Constantine, and Sanghavi, Sujay. Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, pp. 2753–2761, 2014.

Settles, Burr. *Active Learning*. Morgan & Claypool, 2012.

Soltanolkotabi, Mahdi and Candes, Emmanuel J. A Geometric Analysis of Subspace Clustering with Outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

Soltanolkotabi, Mahdi and Candes, Emmanuel J. Robust Subspace Clustering. *The Annals of Statistics*, 42(2): 669–699, 2014.

Soni, Akshay and Haupt, Jarvis. On the fundamental limits of recovering tree sparse vectors from noisy linear measurements. *IEEE Transactions on Information Theory*, 60(1):133–149, 2014.

Vidal, Rene, Sastry, S. Shankar, and Ma, Yi. *Generalized Principal Component Analysis*. Springer-Verlag, 2016.

Vidal, Renee. Subspace clustering. *IEEE Signal Processing Magazine*, 28:52–68, March 2011.

Wagstaff, Kiri, Cardie, Claire, Rogers, Seth, and Schroedl, Stefan. Constrained K-means clustering with background knowledge. In *Proc. Int. Conf. on Machine Learning*, 2001.

Wang, Xiang and Davidson, Ian. Active spectral clustering. In *Proc. 10th Int. Conf. on Data Mining*, 2010.

Wang, Y. and Singh, A. Noise-adaptive margin-based active learning for multi-dimensional data and lower bounds under tsybakov noise. In *Proc. AAAI Conference on Artificial Intellgence*, 2016.

Wang, Yining, Wang, Yu-Xiang, and Singh, Aarti. Graph connectivity in noisy sparse subspace clustering. In *Proceedings of The 19th International Conference on Artificial Intelligence and Statistics*, 2016.

Wang, Yu-Xiang and Xu, Huan. Noisy sparse subspace clustering. In *Proceedings of The 30th International Conference on Machine Learning*, pp. 89–97, 2013.

Wright, John, Yang, Allen Y, Ganesh, Arvind, Sastry, S Shankar, and Ma, Yi. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

Xiong, Caiming, 2016. personal correspondence.

Xiong, Caiming, Johnson, David M., and Corso, Jason J. Active clustering with model-based uncertainty reduction. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 2016. Accepted for publication.

Xu, Qianjun, desJardins, Marie, and Wagstaff, Kiri L. Active constrained clustering by examining spectral eigenvectors. In *Proc. 8th Int. Conf. on Discovery Science*, 2005.

You, Chong, Li, Chun-Guang, Robinson, Daniel P., and Vidal, Rene. Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2016a.

You, Chong, Robinson, Daniel P., and Vidal, Rene. Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2016b.

Zhang, Teng, Szlam, Arthur, Wang, Yi, and Lerman, Gilad. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100:217–240, 2012.