Margin-Based Active Subspace Clustering

John Lipor and Laura Balzano
Department of Electrical and Computer Engineering
University of Michigan, Ann Arbor
{lipor,girasole}@umich.edu

***Abstract*—Subspace clustering has typically been approached as an unsupervised machine learning problem. However in several applications where the union of subspaces model is useful, it is also reasonable to assume you have access to a small number of labels. In this paper we investigate the benefit labeled data brings to the subspace clustering problem. We focus on incorporating labels into the k-subspaces algorithm, a simple and computationally efficient alternating estimation algorithm. We find that even a very small number of randomly selected labels can greatly improve accuracy over the unsupervised approach. We demonstrate that with enough labels, we get a significant improvement by using actively selected labels chosen for points that are nearly equidistant to more than one estimated subspace. We show this improvement on simulated data and face images.**

I. Introduction

The union of subspaces model is a generalization of the subspace model wherein data vectors lie near one of several subspaces. This model is used actively in computer vision to model images of different faces or objects in a variety of lighting conditions. If the face or object pose stays fairly constant, each collection of images for a given object can be represented by a subspace [1], [2]. The model also finds applications in network topology identification [3] and gene expression analysis [4].

Subspace clustering algorithms take a collection of data vectors that lie near a union of subspaces and attempt to simultaneously cluster the vectors and identify the underlying subspaces. As with other clustering methods, it is usually an unsupervised technique, working only with unlabeled data and estimating the best clusters, subspace dimensions, and subspace spans. However in all the applications mentioned above, we do have access to label information – we could ask a human to label two images as being the same face or different, or we could ask domain experts for likely labels in the network and genetics examples. This paper investigates the benefit of using a small amount of labeled data in a classical alternating minimization algorithm for subspace clustering, k-subspaces [5].

In this paper, we study different techniques for selecting which labels to query and compare to both unsupervised clustering and random label selection. We find that the most useful measure of informativeness in a label is given by points that are nearly equidistant to more than one estimated subspace; however, randomly selected labels outperform even this measure when there are few labels. In all cases, even a very small number of labels improves clustering accuracy significantly over the unsupervised approach, making labeling worthwhile. With enough labels, k-subspaces is competitive in terms of accuracy with other algorithms on the extended Yale Face Database B [6], even with random initializations.

II. Problem Formulation

Suppose we have a $D \times N$ data matrix X whose columns are denoted $x_i \in \mathbb{R}^D$, $i = 1, \ldots, N$. These columns form K disjoint clusters, and each cluster of points lies near a low-dimensional subspace. We use C_k, $\mathcal{S}_k$, and d_k, $k = 1, \ldots, K$, to represent the clusters, subspaces, and subspace dimensions. Our goal is to recover the true clusters C_k by leveraging the union of subspaces model. We use the distance of a point to different subspaces when deciding which points to label. Define $P_{\mathcal{S}_k}$ to be the Euclidean projection matrix onto the subspace $\mathcal{S}_k$. Then the distance from a point x_i to a subspace $\mathcal{S}_k$ is defined as

$$\operatorname{dist}(x_i, \mathcal{S}_k) = \|x_i - P_{\mathcal{S}_k} x_i\|_2 \ .$$

To incorporate actively labeled points into the k-subspaces algorithm, at each iteration of the algorithm we select points to be labeled based on criteria that can be computed from the data and current subspace estimates. We will use the following two criteria. Suppose at iteration t we have estimated clusters $C_k^{(t)}$ and subspaces $\mathcal{S}_k^{(t)}$. Let x_i be currently assigned to the cluster k_i^*. We choose either the *min margin* labels for points that are most equidistant to their two closest subspaces:

$$x_{mm} = \underset{i=1,\ldots,N}{\arg\max} \quad \max_{j \neq k_i^*} \frac{\operatorname{dist}(x_i, \mathcal{S}_{k_i^*}^{(t)})}{\operatorname{dist}(x_i, \mathcal{S}_j^{(t)})} \tag{1}$$

or the *max residual* labels for points that are furthest from their assigned subspace:

$$x_{mr} = \underset{i=1,\ldots,N}{\arg\max} \operatorname{dist}(x_i, \mathcal{S}_{k_i^*}^{(t)}) \ . \tag{2}$$

A. Related Work

A comprehensive survey of subspace clustering algorithms can be found in [7]. There is a wide variety of algorithms that work well under different assumptions on the noise and relationship among the subspaces; *e.g.,* [8] requires that the subspaces are linearly independent and is sensitive to noise, while [3] gives results with respect to the probability that neighboring points are in different subspaces; others have various requirements on the collection of principal angles. The algorithms also each provide different tradeoffs between the clustering accuracy, computational speed, and theoretical guarantees.

State-of-the-art clustering accuracy is achieved by the sparse subspace clustering (SSC) [9] and spectral local best-fit flat (SLBF) [10] algorithms. These algorithms work well for several general cases but are known to fail in common situations. For example, the SSC algorithm is known to fail when the smallest principal angle between the subspaces is below a certain data-dependent value [9], even when the data are noise-free. The SLBF algorithm works well even in the case where the principal angles are small, but on real data its performance lags behind that of SSC, as seen in [10]. In this work, we show how a semisupervised approach can be used to counteract these issues.

The algorithms of both SSC [9] and SLBF [10], while achieving great clustering accuracy, are highly computationally burdensome. SSC solves a sparse coding problem for every data vector to be clustered, and then it runs spectral clustering. SLBF is faster, but it starts by initializing with locally best-fit planes to every point in the dataset. In contrast, the k-subspaces algorithm achieves nearly the fastest run time among all commonly tested algorithms [10]. This makes k-subspaces more tractable for the active learning setting, as users could provide labels and interact with the algorithm in real time.

Active sample labeling has been used previously in standard clustering algorithms [11] as well as spectral clustering [12]–[14]. One widely known approach is that of hierarchical sampling [15]. The idea behind this method is to first perform an initial clustering, then iteratively refine the clusters by requesting samples. Our approach is similar in that it alternates between clustering and requesting labels. However, in [15], while the clusters may be chosen actively, points within clusters are still chosen at random. Spectral methods [12]–[14] enforce hard or soft constraints on the similarity matrix, increasing or decreasing the connections between nodes based on the labeled data. In [12], the authors actively select points along the inner and outer boundaries of two clusters. Labels are then incorporated as *must-link* or *cannot-link* constraints in the spectral clustering problem. In [13], the authors implement both a constrained spectral clustering method as well as active query selection based on maximum expected error reduction and allow for both hard and soft constraints. The work of [14] extends these ideas to allow for missing data in the measurements. While these methods could be incorporated directly into spectral subspace clustering algorithms such as SSC and SLBF, our initial empirical results showed only a mild performance improvement, and so we leave careful investigation for future work.

In the context of classification, active learning has been shown to provide significantly improved rates of convergence under many settings [16]–[22]. In contrast to passive learning, where labels are obtained at random, active learning algorithms request labels for maximally informative points, based on a variety of metrics. One important line of work [21], [22], focuses on querying points likely to be misclassified in a margin-based sense. In particular, labels are first obtained at random, and a linear classifier is trained based on the data. In subsequent rounds, the algorithms request the labels of points lying near the decision boundary of the classifier, *i.e.,* points having *minimum margin*. The results of [22] show this type of learner is optimal under a common boundary noise condition, and this work serves as the motivation for our margin-based subspace clustering presented here.

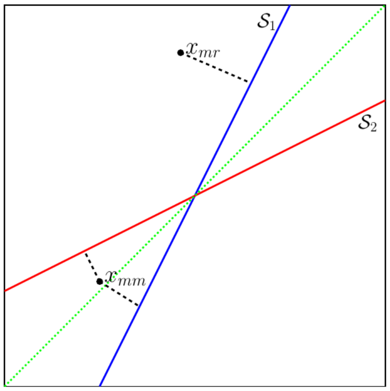


Fig. 1: Two one-dimensional subspaces with *max residual* and *min margin* points. Dashed lines from points denote $\text{dist}(x, \mathcal{S}_k)$, and the center dotted line denotes the decision boundary for points between subspaces.

III. Active k-subspaces

In this section, we modify the k-subspaces algorithm to incorporate labels obtained by querying an oracle and show a method of actively selecting which labels to query in order to improve performance. As mentioned in the previous section, many active learning algorithms choose queries based on some metric of uncertainty, as these points are likely to be misclassified. In the case of subspace clustering, a natural metric is that of maximum residual, as defined in (2). However, as we will see shortly, requesting labels for points according to this metric does not provide a significant benefit over randomly selecting queries. Instead, we turn to a closer analog of the margin-based approach. Consider, for example, two one-dimensional subspaces lying in $\mathbb{R}^2$, as in Fig. 1. The nearest subspace labeling approach can be viewed as classification, where the decision boundary is a function of $\text{dist}(x_i, \mathcal{S}_k)$. As seen in the figure, points of maximum residual (here x_{mr}) may lie far from the boundary, and hence not be in danger of misclassification. In contrast, points whose distance to multiple subspaces is roughly equal (x_{mm}) tend to lie near the decision boundary and correspond to points of minimum margin in the classification sense. Further, it has been noted in [9] that many subspace clustering methods fail when the underlying subspaces are close in some sense. In fact, the authors of [9] show that for two common benchmark datasets, a significant number of points have neighbors lying in different subspaces, indicating the points themselves lie near multiple subspaces. This motivates our *min margin* approach to active label selection.

We now describe our approach to active subspace clustering. To incorporate the obtained labels, we make two changes

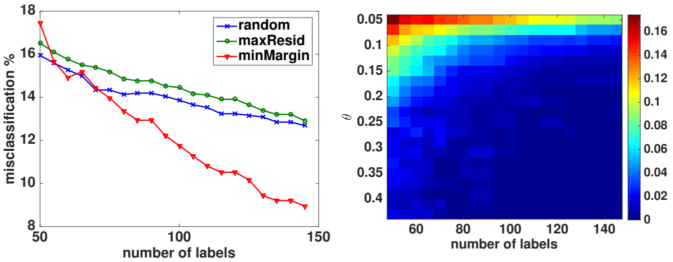
Algorithm 1 Active k-subspaces

```
Input: $X = \{x_1, x_2, \ldots, x_N\}$: data, $K$: number of subspaces, $d$: dimension of subspaces, maxIter: maximum number of iterations, maxLabels: maximum number of labels, method: method of label selection (random, maxResid, or minMargin)
Initialize Subspaces and Clustering: Initialize subspace estimates $\mathcal{S}_1^{(0)}, \ldots, \mathcal{S}_K^{(0)}$ at random. Assign labels to points by nearest subspace estimate.
nLabels ← 0
for t = 1,...,maxIter do
    Update Subspaces Using Labeled and Unlabeled Data: Estimate subspace $\mathcal{S}_k^{(t)}$ by performing SVD on points in $C_k^{(t-1)}$ and obtaining best rank-$d$ estimate.
    Update Labels: Assign labels to points by nearest subspace estimate, $k_i^* = \arg\min_k \mathrm{dist}(x_i, \mathcal{S}_k^{(t)})$.
    if $C_k^{(t)} = C_k^{(t-1)}$ and nLabels $<$ maxLabels then
        $m \leftarrow \min(\lfloor \log(\text{maxLabels}) \rfloor, \text{maxLabels} - \text{nLabels})$
        Request $m$ labels according to method specified and update $C_k^{(t)}$ for the queried points.
        nLabels ← nLabels $+ m$
    end if
end for
```

to the k-subspaces algorithm from [5]. First, for all queried points, we set the estimated cluster to the true cluster. While this step alone improves the algorithm's performance, a more significant improvement is obtained by making one further change. In the subspace estimation step, when the number of labeled points for a given subspace $\mathcal{S}_k$ is greater than d_k, those points alone are used to estimate the subspace. This guarantees our subspace estimate stems only from points lying in or near the true subspace. However, we will see shortly that this step can have a negative effect when just more than d_k labels are obtained from a subspace, making the subspace estimate noisy. An outline of the algorithm is given in Algorithm 1. The algorithm first performs clustering using k-subspaces until the cluster assignments are stable. At this point, $\lfloor \log(\text{maxLabels}) \rfloor$ labels are requested according to the desired metric (random selection, *max residual*, or *min margin*). The algorithm then follows standard k-subspaces again and the process is repeated until maxLabels labels have been obtained and the algorithm runs a terminal number of iterations. Note that the choice of $\lfloor \log(\text{maxLabels}) \rfloor$ is based on empirical results. However, the algorithm is not sensitive to this number in practice. Determining an optimal number of labels per algorithm iteration is a topic for our future study.

IV. Simulations

In this section, we compare the performance of the proposed *min margin* label selection with unsupervised k-subspaces, random label selection, and *max residual* label selection on both real and simulated data. We set maxIter to 128, which is sufficient to request maxLabels labels in all cases. For all methods, ten restarts are performed and the misclassification rate of the one with smallest ℓ_2 error is recorded. Further, we seed the random number generator so that the subspace initializations, random data generated, and random face choices are consistent across trials.



(a) (b)

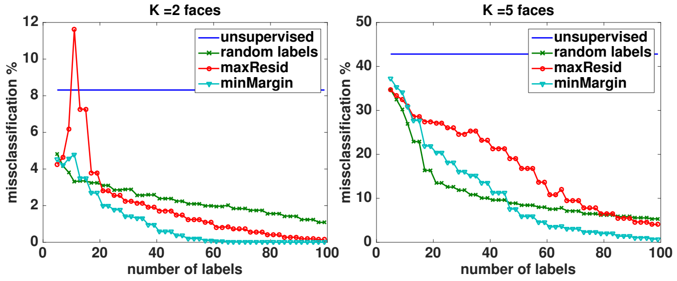
Fig. 2: Misclassification results for simulated data. (a) Misclassification rate as a function of number of supervised labels for all label selection methods. Unsupervised error is 39.74%. (b) Misclassification rate as a function of number of labels and subspace angle for *min-margin* label selection.

For the simulated data, we use an ambient dimension $D = 50$ and then generate data in a manner similar to [9]. We use $K = 3$ subspaces of dimension $d = 4$, and then generate the subspaces such that all angles between subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ and subspaces $\mathcal{S}_2$ and $\mathcal{S}_3$ are equal to θ. From each subspace, 250 points are drawn randomly and corrupted with zero-mean additive Gaussian noise with variance 0.01. We then record the average misclassification rate over 100 trials as a function of θ and the number of labels requested. Fig. 2 shows the resulting error as a function of the number of labels for $\theta = 0.05$, as well as a heat map displaying the misclassification rate as a function of θ and the number of labels requested. Although not shown, with fewer than 50 labels, random selection outperforms the *min-margin* method, and for fewer than 20 labels, unsupervised learning yields the best performance. This is due to the fact that the subspace estimates are performed using only labeled data, which is noisy enough to provide a poor estimate when very few points are used. However, as the number of labels increases, we see that incorporating labels significantly improves the clustering performance. Further, as the number of labels increases, the *min margin* approach dominates other label selection methods.

Next, we test our algorithm on the extended Yale Face Database B in a manner following [10]. This dataset consists of face images for $K = 38$ individuals under a variety of lighting conditions, where each individual corresponds to a subspace to be identified. For each person, there are $N_k = 64$ images of size 192×168 pixels. Since the intrinsic dimension of each subspace is known to be close to 5, we first project the data onto its first $5K$ principal components, and then perform clustering on the dimension-reduced data for 100 randomly selected subsets of size $K = 2, 3, \ldots, 8$ persons, following the methodology of [10]. Due to the distribution of the data in this set (analyzed in [9]), algorithms relying on a nearest neighbor approach tend to suffer in performance, as seen in

K	2	3	4	5	6	7	8
unsupervised	8.63	24.64	39.01	45.04	49.91	50.88	54.79
random labels	2.88	4.51	5.86	6.78	7.31	7.39	8.51
max residual	2.52	4.59	6.75	8.61	9.86	10.02	14.37
min margin	1.27	2.20	2.25	2.40	2.88	2.45	3.06
state of the art (unsupervised)	3.46	6.08	10.04	10.32	11.02	11.85	12.47

TABLE I: Misclassification rates for extended Yale Face Database B with $3Kd = 15K$ labels. The final row denotes the minimum unsupervised error as reported in [10], which includes the results of algorithms in [9], [10], [23].



(a)

(b)

Fig. 3: Misclassification rates vs. number of supervised labels for extended Yale B dataset for (a) $K = 2$ and (b) $K = 5$.

the simulations performed in [10]. In particular, as the number of subspaces increases, the LBF-based algorithms struggle significantly, achieving classification errors of 27-36% (for the various forms) in the case of 8 subspaces. In the absence of significant preprocessing (*i.e.,* applying robust principal component analysis as in [9]), this data set presents challenges to all existing algorithms. Fig. 3 shows the misclassification rate as a function of the number of labels requested for $K = 2$ and $K = 5$ faces. In this case, all semisupervised methods significantly outperform unsupervised clustering as expected. Further, for sufficiently many labels, we see that the *min-margin* approach again dominates both random and *max residual* queries. Note that for very few labels, random label selection achieves the best performance, since the requested labels are spread evenly through all subspaces. Our simulations suggest that approximately $2Kd = 10K$ labels are sufficient for active methods to match random labeling. Table I shows the classification errors for $K = 2, 3, \ldots, 8$ individuals with $3Kd$ labels queried. Here we see that the *min-margin* approach yields a strong improvement over other methods. Further, our algorithm achieves a much lower classification error than the best performers from [10].

V. Conclusion and Future Work

We have shown that an active approach to label selection can provide significant performance improvements to the efficient k-subspaces algorithm. The presented algorithm is a natural extension of optimal algorithms from active classification. With few labels, the k-subspaces algorithm is competitive with state of the art algorithms such as SSC and SLBF. Several open problems remain. First, we wish to incorporate active labeling into other subspace clustering algorithms to give a fair comparison for clustering accuracy, labels needed, and computational speed. Labels could be directly incorporated into SLBF and spectral methods, and we could introduce them as constraints in the ℓ_1 minimization problem of SSC. Further, we have seen that for very few labels, random queries outperform active selection. In this regime, a mixture of active and random labels would likely improve performance.

References

[1] R. Basri and D. Jacobs, “Lambertian reflectance and linear subspaces,” *IEEE TPAMI*, vol. 25, no. 2, pp. 218–233, February 2003.

[2] N. Oliver, B. Rosario, and A. Pentland, “A bayesian computer vision system for modeling human interactions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[3] L. Balzano, B. Eriksson, and R. Nowak, “High rank matrix completion and subspace clustering with missing data,” in *Proceedings of the conference on Artificial Intelligence and Statistics (AIStats)*, 2012.

[4] D. Jiang, C. Tang, and A. Zhang, “Cluster analysis for gene expression data: A survey,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 16, no. 11, pp. 1370–1386, 2004.

[5] P. S. Bradley and O. L. Mangasarian, “k-Plane clustering,” *Journal of Global Optimization*, vol. 16, pp. 23–32, 2000.

[6] A. Georghiades, P. Belhumeur, and D. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[7] R. Vidal, “Subspace clustering,” *IEEE Signal Processing Magazine*, vol. 28, pp. 52–68, Mar. 2011.

[8] J. P. Costeira and T. Kanade, “A multibody factorization method for independently moving objects,” *International Journal of Computer Vision*, vol. 29, no. 3, 1998.

[9] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2765–2781, Nov. 2013.

[10] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, “Hybrid linear modeling via local best-fit flats,” *International Journal of Computer Vision*, vol. 100, pp. 217–240, 2012.

[11] B. Settles, *Active Learning*. Morgan & Claypool, 2012.

[12] Q. Xu, M. desJardins, and K. L. Wagstaff, “Active constrained clustering by examining spectral eigenvectors,” in *Proc. 8th Int. Conf. on Discovery Science*, 2005.

[13] X. Wang and I. Davidson, “Active spectral clustering,” in *Proc. 10th Int. Conf. on Data Mining*, 2010.

[14] F. L. Wauthier, J. Nebojsa, and M. I. Jordan, “Active spectral clustering via iterative uncertainty reduction,” in *Proc. 18th ACM Int. Conf. on Knowledge Discovery and Data Mining*, 2010.

[15] S. Dasgupta and D. Hsu, “Hierarchical sampling for active learning,” in *Proc. International Conference on Machine Learning*, 2008.

[16] S. Dasgupta, “Coarse sample complexity bounds for active learning,” in *Proc. Advances in Neural Information Processing Systems*, 2005.

[17] ——, “Analysis of a greedy active learning strategy,” in *Proc. Advances in Neural Information Processing Systems*, 2005.

[18] R. Castro and R. Nowak, “Minimax bounds for active learning,” *IEEE Trans. Inf. Theory*, vol. 54, pp. 2339–2353, May 2008.

[19] D. Cohn, L. Atlas, and R. Ladner, “Improving generalization with active learning,” *Machine Learning*, vol. 15(2):201-221, 1994.

[20] M.-F. Balcan, A. Beygelzimer, and J. Langford, “Agnostic active learning,” in *Proc. International Conference on Machine Learning*, 2006.

[21] M.-F. Balcan, A. Broder, and T. Zhang, “Margin based active learning,” *Learning Theory*, p. 350, 2007.

[22] Y. Wang and A. Singh, “Noise-adaptive margin-based active learning for multi-dimensional data,” *arXiv Preprint*, vol. 1406.5383v1, 2014.

[23] Y. Ma, H. Derksen, W. Hong, and J. Wright, “Segmentation of multivariate mixed data via lossy coding and compression,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 9, pp. 1546–1562, 2007.