

Probabilistic Sparse Subspace Clustering Using Delayed Association

Maryam Jaberi
Department of Computer Science
University of Central Florida
Orlando, FL, USA

Marianna Pensky
Department of Mathematics
University of Central Florida
Orlando, FL, USA

Hassan Foroosh
Department of Computer Science
University of Central Florida
Orlando, FL, USA

Abstract—Discovering and clustering subspaces in high-dimensional data is a fundamental problem of machine learning with a wide range of applications in data mining, computer vision, and pattern recognition. Earlier methods divided the problem into two separate stages of finding the similarity matrix and finding clusters. Similar to some recent works, we integrate these two steps using a joint optimization approach. We make the following contributions: (i) we estimate the reliability of the cluster assignment for each point before assigning a point to a subspace. We group the data points into two groups of “certain” and “uncertain”, with the assignment of latter group delayed until their subspace association certainty improves. (ii) We demonstrate that delayed association is better suited for clustering subspaces that have ambiguities, i.e. when subspaces intersect or data are contaminated with outliers/noise. (iii) We demonstrate experimentally that such delayed probabilistic association leads to a more accurate self-representation and final clusters. The proposed method has higher accuracy both for points that exclusively lie in one subspace, and those that are on the intersection of subspaces. (iv) We show that delayed association leads to huge reduction of computational cost, since it allows for incremental spectral clustering.

I. INTRODUCTION

The problem of clustering high dimensional data when it is formed from a union of multiple subspaces is studied in research areas such as machine learning, computer vision, and pattern recognition. In a large variety of applications, data naturally form clusters of low dimensional subspaces. In video processing, for instance, motion trajectories are usually represented by high-dimensional vectors. Yet, they can span low-dimensional linear manifolds [1]. Also, in face/image classification, under some conditions, images lie on low-dimensional linear subspaces [2]. Subspace clustering algorithms are designed to discover clusters in a mixture of high-dimensional vectors drawn from multiple probability distributions. The idea is that, when a subset of high dimensional data belongs to a cluster, then the points in the cluster lie in a low dimensional subspace. Several methods are proposed in this area based on algebraic [3], iterative [4], statistical [5] and spectral clustering [6], [7]. Spectral clustering methods form a similarity matrix that describes the similarity between data points, in order to cluster them. In these methods, points in subspaces are self-representative. In other words, when

subspaces are independent and noiseless, by having sufficient number of points in each subspace, any point in a subspace can be represented as a linear combination of other points in that subspace. Given a matrix $X \in \mathbb{R}^{n \times N}$, with columns drawn from a union of C independent linear subspaces of \mathbb{R}^n , $\{S_k\}_{k=1}^C$ with dimensions $\{d_k \ll n\}_{k=1}^C$, any data point x_i can be represented as $x_i = X_{s_k} z_i$, where $x_i \in S_k$, X_{s_k} are all the data points in S_k except for x_i , and z_i is a coefficient column vector. Column z_i can be recovered as a sparse solution of an optimization problem. The optimal solution would include non-zero coefficients corresponding to the columns of X_{s_k} that are in the same subspace as x_i . In a general form, and in the presence of sparse norm and bounded noise or sparse outlier entries, the optimization problem can be written as follows

$$\min_Z (\|Z\|_\ell + \|E\|_{\ell'}) \quad \text{S.T.} \quad E = X - XZ, \quad z_{ii} = 0, \quad (1)$$

where Z is a coefficient matrix, Z_i is its i -th column and z_{ii} are the diagonal elements. E represents a bounded noise or sparse outliers. In the literature, the different choices of $\|\cdot\|_\ell$ and $\|\cdot\|_{\ell'}$ are studied [8], [7]. Using Z , a similarity matrix is defined as: $\tilde{Z} = \frac{1}{2}(|Z| + |Z^T|)$. A clustering algorithm such as normalized cuts [9] can then be applied to the similarity matrix to find the clusters. The authors of [10], [11] developed unified iterative frameworks for updating the low-rank matrix Z using clustering results and subsequently finding the clusters using this new version of Z . The idea is that both the sparse similarity matrix and the clusters depend on each other. Thus, an alternating method can be used in the spectral clustering step to remove noise from the similarity matrix, resulting in a more accurate estimator of the similarity matrix. This leads to more accurate clusters in the spectral clustering step. This method uses an approach similar to (1) and defines an objective function as follows:

$$\min_{Z, Q} (\lambda \|Z\|_{1, Q} + \|E\|_{\ell'}) \quad \text{S.T.} \quad z_{ii} = 0, \quad (2)$$

where $\|Z\|_{1, Q}$ depends on the clustering matrix $Q \in \{0, 1\}^{N \times C}$ obtained in the previous step. An extended version of this approach is proposed in [11], where Q includes continuous real values obtained by keeping the eigenvectors associated with C smallest eigenvalues of the computed Laplacian matrix. This has the advantage of including continuous real values for re-weighting the representation matrix in the next iteration.

However, this method has the disadvantage of removing less noise from the similarity matrix compared to $Q \in \{0, 1\}^{N \times C}$.

In this paper, we introduce a joint optimization approach that converges to an optimal solution for self-representation of clusters in high dimensional data. The novelty of our approach is that we delay the association of a point to a cluster at a given iteration, when such association is *uncertain*. Remaining points are considered *certain*, and clustered right away. This helps improve the accuracy of updating the elements of the similarity matrix Z in the next iteration. At each iteration, *certain* points, leading to get a better representation of subspaces. Also, it allows *uncertain* points to be drawn closer to the correct subspace before the final assignments is made. Two main advantages are: (i) We effectively combine the advantages of both hard and soft clustering, leading to more accurate representation of the points in subspaces, and hence more accurate final results, since certain points are hard-clustered, whereas for uncertain points continuous values are used for re-weighting the representation in the next iteration. (ii) This process lends itself to the possibility of using an incremental spectral clustering, which in turn leads to a huge reduction in complexity and computational time.

II. PROPOSED METHOD

In the proposed joint optimization method, we alternate between finding the coefficient matrix Z and the final clustering assignments. Assuming the initial clustering if points are given, we define assignments of points to C clusters by a soft clustering matrix $\Phi \in [0, 1]^{N \times C}$ where elements ϕ_{ij} represent the probability of point i belonging to subspace j , so that $\sum_{j=1}^C \phi_{ij} = 1$, $i = 1, \dots, N$. In particular, $\phi_{ij} = 1$ when point i is confidently assigned to cluster j , and $\phi_{ij} = 0$ when point i is confidently excluded from cluster j . Thus, we can define the *association matrix* A as $A = \Phi\Phi^T$. Elements a_{ij} of matrix $A \in [0, 1]^{N \times N}$ indicate the strength of the connection between points i and j in the dataset. If Φ were a clustering matrix with entries of zero or one only, then one would have $a_{ij} = 1$ if points i and j lie in the same class and $a_{ij} = 0$ otherwise.

The sparse similarity matrix \bar{Z} indicates the connection between each point and all other points in the dataset. On the other hand, the association matrix A represents the relationship between clustered points. Hence, these two matrices are related to each other and present similar information about points in the dataset. The association matrix A can be used to denoise and sparsify the coefficient matrix Z , while taking into account the coefficient matrix Z can lead to more accurate recovery of the association matrix A . The following equation formulates the connection between Z and the newly introduced matrix A :

$$\min_{Z, E, A} \left(\lambda_0 \|Z\|_1 + \frac{1}{2} \|E\|_F^2 + \lambda_1 \|(\mathbf{1} - A) * Z\|_F^2 \right) \quad (3)$$

subject to $E = X - XZ$, $z_{ii} = 0$, $\text{rank}(\Phi) = C$.

where $A = \Phi\Phi^T$, $\mathbf{1}^{N \times N}$ is the matrix with all unit elements, and $*$ is the Hadamard product. The first and second term

in (3), similar to equation (1), enforce the sparsity and small errors between points and their linear representations, respectively. The last term $\|\cdot\|_F^2$ in this equation imposes connectivity between points in the same subspace and removes connectivity of points in different subspaces. Indeed, since the entries a_{ij} of matrix A represent the probabilities of points i and j being in the same cluster, the elements $1 - a_{ij}$ of matrix $\mathbf{1} - A$ are smaller and do not force the respective entries z_{ij} and z_{ji} of matrix Z to be small.

The proposed method jointly searches for a sparse self-representation matrix Z that satisfies $E = X - XZ$ and the soft subspace segmentation matrix Φ that satisfies $\text{rank}(\Phi) = C$. In order to identify the subspace clusters and mitigate/eliminate noise and outliers from the coefficient matrix Z and the association matrix A , we alternate between finding the association matrix A and the coefficient matrix Z . For a given matrix A , the objective function (3) is convex in $\{Z, E\}$. Given $\{Z, E\}$, we estimate A using spectral clustering. A main novelty of our approach is in the second step, where given the matrix $\{Z, E\}$, we generate the association matrix $A \in [0, 1]^{N \times N}$.

Updating Φ and A : Given the similarity matrix \bar{Z} and the error matrix E , our objective is to find the soft segmentation matrix Φ and the association matrix A . The solution for determining Φ as the probability of assigning points to C clusters can be defined as a pairwise data clustering problem [9], [12]. One can find hard clusters by applying a spectral clustering algorithm such as normalized cuts. Given the initial hard clusters obtained from the spectral clustering algorithm, we need to determine the likelihood of a point x_i belonging to each subspace $\{S_k\}_{k=1}^C$, for the purpose of computing Φ . We define $\|\delta_{s_k}(\bar{z}_i)\|_1$ as the *degree of association* of each point x_i with the subspace $\{S_k\}_{k=1}^C$, where \bar{z}_i is the i^{th} column of similarity matrix \bar{Z} corresponding to point x_i , and $\delta_{s_k}(\bar{z}_i)$ is found by keeping all the elements of the vector \bar{z}_i that are associated with subspace S_k , and setting the remaining elements to zero. A point x_i , $i = 1, \dots, N$, is assumed to be more likely to be associated with a subspace S_k if it has a higher *degree of association* to the subspace S_k defined as:

$$p_{ik} = \frac{\|\delta_{s_k}(\bar{z}_i)\|_1}{\|(\bar{z}_i)\|_1} \quad k = 1, \dots, C \quad (4)$$

We build the matrix $P \in [0, 1]^{N \times C}$ with elements p_{ik} being the probabilities of assigning point x_i , $i = 1, \dots, N$, to the subspace S_k , $k = 1, \dots, C$, where $\sum_k p_{ik} = 1$. The soft subspace segmentation matrix Φ is determined as described below. For each i , we denote $k_i = \arg \max_{1 \leq k \leq C} \{p_{ik}\}$ and divide points into *certain* and *uncertain* using the delayed association parameter Ω . The soft clustering matrix Φ is determined by the computed probability matrix P as described below. For each i , we denote $k_i = \arg \max_{1 \leq k \leq C} \{p_{ik}\}$ and divide points into *certain* and *uncertain* by applying a threshold to the probability matrix P as follows:

$$\phi_{ij} = \begin{cases} 1, & \text{if } j = k_i \text{ and } p_{ik_i} \geq \Omega \\ 0, & \text{if } j \neq k_i \text{ and } p_{ik_i} \geq \Omega \\ p_{ij}, & \text{if } p_{ik_i} < \Omega \end{cases} \quad (5)$$

The delayed association parameter Ω is calculated by finding the average affinity between points of a subspace using the following equation:

$$\Omega = 1 - \frac{\sum_{i \neq j} M_{ij}}{(C-1) \sum_{i=j} M_{ij}} \quad \text{where } M = P^T P \quad (6)$$

Note that matrix $M \in \mathbb{R}^{C \times C}$ demonstrates the affinity between points in subspaces. The main diagonal of this matrix shows the correlation between points of a subspace and off-diagonal entries showing the similarity of points in different subspaces. Thus, when the similarity matrix \bar{Z} turns into a block diagonal matrix, the probabilities p_{ik} will be pushed through zero or one and the defined matrix M turns into the identity matrix with low affinity between points of different subspaces and strong connectivity among points of a subspace. We use matrix M to find the delayed association parameter Ω in each iteration. Based on the definition of Ω , when there is a high ambiguity between clustered points, matrix M turns into a matrix containing all similar entries and $\Omega \approx 0$. This allows more points to be grouped as *uncertain* and give them the chance to find a better representation before being assigned to a subspace. On the other hand, when there is a low affinity between points of different subspaces, M turns into the identity matrix and $\Omega \approx 1$, which allows more points to be grouped as *certain*.

Given an assignment matrix Φ , we form the association matrix $A \in [0, 1]^{N \times N}$ which is a symmetric matrix. For each point x_i marked as *certain*, $a_{ij} \in \{0, 1\}$ shows if point x_j is assigned to the same class as x_i . For *uncertain* points $a_{ij} \in [0, 1]$ represents the probability of assigning x_i and x_j to the same cluster. The rationale behind the method is that, in the sparse similarity matrix \bar{Z} , each column i includes the coefficients associated with other points used to represent the point x_i . These coefficients indicate the connection between a point x_i and all other points. When a point is marked as *certain* in the association matrix A , we discard coefficients from other clusters by setting $a_{ij} = 0$, even when the values are large. By setting $a_{ij} = 1$ for all $i, j \in S_k$, we also improve the connection between points within the same subspace. For an *uncertain* point, however, there is an ambiguity regarding the correct cluster. Using the definition in formula (5), we preserve all strong connections, regardless of the cluster to which it is assigned in the spectral clustering step. We include all the strongly connected points while updating the coefficient matrix Z in the next iteration. This approach helps us improve the connections between points and reduce noise in the next iteration. This process is summarized in algorithm 1.

Updating Z and E: Given the probability matrix Φ and association matrix A , we update the coefficient matrix Z and error matrix E in the next step by solving the optimization equation (3) with respect to $\{Z, E\}$

$$\min_{Z, E} \left(\lambda_0 \|Z\|_1 + \frac{1}{2} \|E\|_F^2 + \lambda_1 \|(\mathbf{1} - A) * Z\|_F^2 \right) \quad (7)$$

subject to $E = X - XZ, \quad z_{ii} = 0$

Alternating between updating matrices $\{Z, E\}$, and matrices $\{\Phi, A\}$, as explained, helps us remove small values in the sparse coefficient matrix Z and obtain a better pairwise representation of points with less noise/outliers, and hence a more accurate clustering result.

Algorithm 1 Finding clustering matrix Φ

Require: Cluster assignment, similarity matrix $\bar{Z}^{(t)}$

- 1: Compute matrix $P^{(t)}$ using equation (4)
 - 2: Compute threshold $\Omega^{(t)}$ using equation (6)
 - 3: **for** $i \in \{1, \dots, N\}$ **do**
 - 4: $k_i = \arg \max_{1 \leq k \leq C} \{p_{ik}^{(t)}\}$.
 - 5: **if** $P_{ik_i}^{(t)} \geq \Omega^{(t)}$ **then** {Mark i as “certain”}
 - 6: **for** $j \in \{1, \dots, C\}$ **do**
 - 7: **if** $j = k_i$ **then**
 - 8: $\phi_{ij}^{(t)} = 1$
 - 9: **else**
 - 10: $\phi_{ij}^{(t)} = 0$
 - 11: **else** {Mark i as “uncertain”}
 - 12: $\phi_{ij}^{(t)} = p_{ij}^{(t)} \quad \forall j \in \{1, \dots, C\}$
 - 13: **return** $\Phi^{(t)}$
-

In the initial step of the proposed method, we set $A = \mathbf{1}^{N \times N}$. This is equivalent to removing the third term in the equation (3), which converts it to equation (1). We compute the coefficient matrix Z and error E using (1). Then, using spectral clustering [12], we divide the input data X into C clusters. By defining association degrees (4), we form the probability matrix P , the soft clustering matrix Φ and the association matrix A . After this initial step, we alternate between optimizing with respect to $\{\Phi, A\}$ and $\{Z, E\}$. After updating the sparse similarity matrix \bar{Z} , we need to find the clusters in each iteration. Previous methods (e.g. [11]) use normalized cuts and recompute the solution from scratch, with a time complexity of $O(N^{3/2})$ in the best case [13]. Our delayed association of points allows us to resort to an incremental spectral clustering [14] at a substantially lower computational cost, since the computed eigenvectors are updated when there are changes in the similarity matrix \bar{Z} . The running time of this clustering approach is close to $O(N)$ when every column of the coefficient matrix Z is sparse. In our approach, when *certain* points do not have any coefficient that connects them to *uncertain* points, they do not need to be updated in spectral clustering. Incremental clustering is applied to *uncertain* points and all other points that are connected to *uncertain* in \bar{Z} . As a result of updating only *uncertain* points in the clustering matrix, the time complexity is drastically reduced. In section III, we further show that the number of *uncertain* points, denoted as $\kappa(\Phi^{(t)})$, generally decreases, which implies that the cost of incremental updating itself is reducing at each iteration. The proposed method is summarized in algorithm 2.

As pointed out earlier, we expect that a delayed probabilistic association in subspace clustering leads us to a better self-representation matrix and better clustering assignment. Figure 1 illustrates an example of updates in an association matrix A over 3 sequential steps. The data is from the Extended Yale Database B [15] with $C = 5$ subspaces (subjects). As shown

Algorithm 2 Probabilistic sparse subspace clustering

Require: $X \in \mathbb{R}^{n \times N}$

- 1: **Initialization:** set $A^{(0)} = \mathbf{1}^{N \times N}$
 - 2: **repeat**
 - 3: **Update** $Z^{(t)}$: $\min_{Z, E} (\lambda_0 \|Z\|_1 + \frac{1}{2} \|E\|_F^2 + \lambda_1 \|(1 - A) * Z\|_F^2)$
 - 4: **Set** $\bar{Z}^{(t)} = \frac{1}{2} [Z^{(t)} + (Z^T)^{(t)}]$
 - 5: **Find** clusters by incremental spectral clustering[14] or re-initialize by spectral clustering. [12]
 - 6: **Update** $\Phi^{(t)}$ using Algorithm 1
 - 7: **Set** $A^{(t)} = (\Phi^{(t)})(\Phi^{(t)})^T$
 - 8: **until** $\Phi^{(t)} = \Phi^{(t-1)}$ or $\kappa(\Phi^{(t)}) \geq \kappa(\Phi^{(t-1)})$ or $t \geq t_{\max}$
-

in this figure, the percentage of *uncertain* points is decreased from $\% \kappa_1 = 30\%$ in the first iteration to $\% \kappa_3 = 1\%$ in the third iterations. Also, the misclassification errors of subspace clustering are decreased from 7.19% to 0.1%.

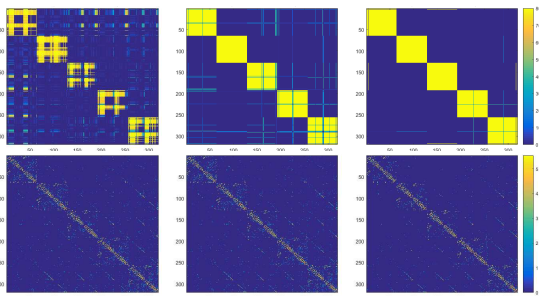


Fig. 1: Updates in A (top) and Z (bottom) over three consecutive iterations. Data is from the Yale B Face Dataset with 5 subjects. The misclassification errors are 7.19%, 1.25% and 1.0% in iterations $t = \{1, 2, 3\}$ respectively.

III. EXPERIMENTAL RESULTS

To evaluate the accuracy of the proposed subspace clustering method, we used different metrics. The first metric we used is a direct measure of the misclassification error of sparse subspace clustering results. This is defined as:

$$Err = \frac{\#\text{Misclassified Points}}{N} \quad (8)$$

This is the total number of incorrectly clustered points over the total number of points in the population. Another metric used is the subspace sparse recovery error [16]. This metric computes the error of representing each point in its final cluster according to the coefficient matrix Z . The columns of z_i determine all the coefficients to self-represent the point x_i using all the other points in the dataset. Using the result of the clustering algorithm, each column z_i gets divided into C classes $\delta_{s_1}(z_i), \delta_{s_2}(z_i), \dots, \delta_{s_C}(z_i)$, where each of the $\delta_{s_k}(z_i)$ are the coefficients of representing x_i in cluster k . Assuming m is the correct cluster for point x_i , and that $\delta_{s_m}(z_i)$ are the corresponding coefficients to reconstruct the point, the average subspace sparse recovery error can be defined as:

$$SSR = 1 - \frac{1}{N} \sum_{i=1}^N \frac{\|\delta_{s_m}(Z_i)\|_1}{\|Z_i\|_1} \quad (9)$$

In the following sections, we study the accuracy of the proposed method for both synthetic and real datasets, and compare the results with state-of-the-art algorithms. In our experiments, the value of $t_{\max} = 10$ and $\frac{\lambda_0}{\lambda_1} = 100$.

A. Synthetic Data: Subspace Intersection

As pointed out earlier, we expect that a delayed probabilistic association in subspace clustering is better suited to handle subspaces with large intersections and overlaps, i.e. those with a large number of points that belong to more than one subspace. This is of practical interest, since in most real data there is either ambiguity due to noise or significant similarity of points in different clusters, due to nested subspaces. In this section, we experimentally study this important problem. To generate our dataset, we used the method described in [16]. We examine the effect of intersection between subspaces when each subspace $\{S_j\}_{j=1}^C$ has a true dimension of $d = 10$ in \mathbb{R}^n with an ambient dimension $n = 200$, and an intersection dimension of s (i.e. sharing s basis vectors). The first subspace S_1 of dimension d is generated uniformly at random. To generate each of the remaining subspaces for each subspace $\{S_j\}_{j=2}^C$, we generated two sets of basis: (i) the intersection basis $S_j^{(1)}$ of dimension s where $s < d$. (ii) the disjoint basis $S_j^{(2)}$ with dimension $d - s$. Then, the basis for each of the remaining subspaces ($S_j, j = 1..C$) are formed as $S_j = S_j^{(1)} \cup S_j^{(2)}$. We generated three different models by varying the number of subspaces C from two to four and samples $N_j = 100$ points uniformly at random from each subspace. We generated 20 instances from each of these models and changed the ratio of the intersection between subspaces in the range $\frac{s}{d} = \{40\%, \dots, 90\%\}$.

Clustering Accuracy: In the first experiment, we examined the convergence of the algorithm, i.e. the clustering misclassification versus the number of iterations. We show the result for 50% and 90% intersections and for two to four clusters. The average number of iterations t before reaching the condition in algorithm 2 is $t = 5$. The complete summary of average misclassification error over 20 independent trials is shown in table I. As shown in this table, we compare the misclassification error of the proposed method with SSC [17] as a baseline and S^3C [11] as the state-of-the-art. The proposed delayed association method consistently outperforms both the baseline and the state-of-the-art methods. Additionally, table II shows a complete summary of average representation errors (equation (9)) on the same experiment. As seen in this table, our method computes a sparser matrix Z compared with SSC. However, in some cases, it may compute a less sparse matrix Z compared with S^3C due to delayed association. When there is more ambiguity in the clusters, our method keeps more points in the *uncertain* group, making the coefficient matrix Z temporarily less sparse compared with S^3C . However, this helps keep the classification error lower (see Table I). Figure 2 shows the average misclassification error (equation (8)) for 20 independent trials. We compared the results of the proposed method with SSC [17] and S^3C [11]. As illustrated,

the proposed method is more accurate compared with state-of-the-art algorithms. Graphs in figure 3 illustrate the average changes in accuracy of subspace clustering by alternately updating $\{Z, E\}$ and A . These graphs compare the accuracy of the proposed algorithm with the approaches in S^3C [11] and in SSC.

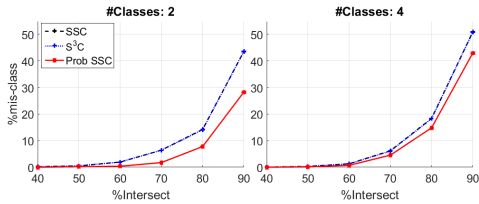


Fig. 2: Average % misclassification errors (8) for Prob-SSC, SSC and S^3C methods over 20 independent trials.

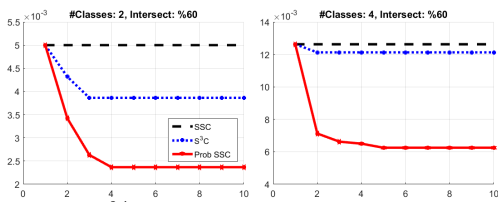


Fig. 3: Average % misclassification errors over 10 iterations. Results are shown for Prob-SSC, SSC and S^3C .

To show that the percentage of uncertain points ($\kappa(\Phi)$) decreases and leads to converge the proposed method, we removed constraints of $\Phi^{(t)} = \Phi^{(t-1)}$ or $\kappa(\Phi^{(t)}) \geq \kappa(\Phi^{(t-1)})$ in algorithm 2 and repeated the process for 20 iterations. Figure 4 shows the average percentage of uncertain points decreases monotonically. The experiment is on 50% intersection between subspaces over 20 independent trials.

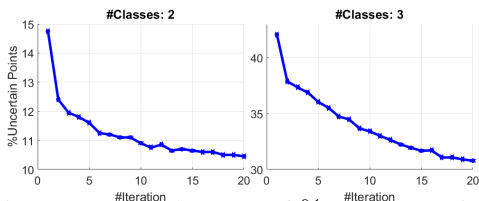


Fig. 4: Average decrease of % uncertain points.

B. Real Data: Face Clustering

Face classification is one of the many applications of subspace clustering. Face image classification techniques try to cluster images of the same subject under varying illumination or imaging conditions in one group. Study in [2] shows that a set of images of an object under varying illumination lies in a low-dimensional linear subspace of the image space of up to nine dimensions. Thus, face images in this condition can be clustered using subspace clustering techniques. The Extended Yale B Database [15] is a facial dataset widely used in subspace clustering literature [8], [18] and contains 2,414 frontal face images of 38 human subjects taken under approximately 64 different illumination conditions. This dataset is considered as a challenging one for clustering techniques

due to its extreme lighting variations. In this experiment, we used the proposed algorithm to study the improvement in accuracy of face clusters compared to previous methods. We used subsets of $C = \{2, 3, 5, 8, 10\}$ different subjects (subspaces) from the dataset. Each subject includes 64 images ($N_j = 64$). Each downsampled image has a dimension of 48×42 that is vectorized to a 2016-dimensional vector. These 38 subjects are divided in 4 groups of 1-10, 11-20, 21-30 and 31-38 [11]. Similar to the reported results in [11], we kept the original size of the image vectors (2016) and reported the result on the whole dataset. We examined the accuracy of the proposed subspace clustering algorithm on this dataset. Table III shows the clustering error percentages for the proposed algorithm. We compared the error of our proposed method with S^3C [11], SSC[17], and LRR [6]. We cite the reported results in [19], [7] and [11] in this table. As it is shown in this table, our method outperforms all other state-of-the-art algorithms.

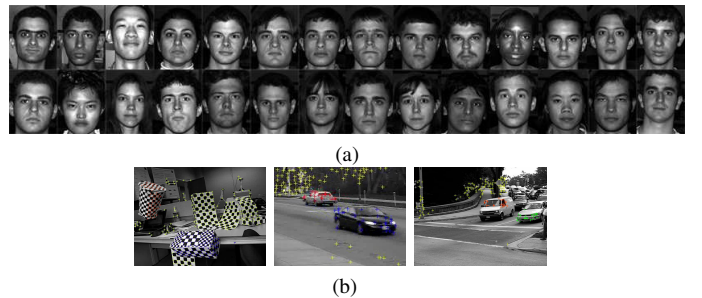


Fig. 5: Real data. (a): Faces from the Extended Yale Dataset B. (b): Images from the Hopkins 155 dataset.

C. Real Data: Motion Segmentation

Motion segmentation of trajectory data has been widely studied in computer vision applications such as reconstructing dynamic scenes [1]. In dynamic scenes with multiple moving rigid objects, the trajectories can be represented by high-dimensional vectors. Yet, they can span low-dimensional linear manifolds [1]. It is shown that, in an affine subspace, trajectories of a single motion in an ambient dimension \mathbb{R}^{2F} , where F is the number of frames, lies in a low-dimensional linear subspace of up to four dimensions. Thus, subspace clustering algorithms can be used to cluster the trajectories of different motions in separate subspaces. In this part, we used Hopkins 155 motion segmentation dataset [20] to examine the proposed subspace clustering method. We compared the misclassification errors of our proposed method on this dataset with LRR[6], SSC [17], and S^3C [11]. Table IV shows the clustering error percentages for these methods. We cite the reported results in [11] in this table. As shown, our proposed method outperforms all other state-of-the-art algorithms.

IV. CONCLUSION

We introduced a new subspace clustering method that outperforms state-of-the-art methods reported recently in the literature. This boost in accuracy is because we replace the usual clustering matrix Q with an *association matrix* A that

TABLE I: Intersection %misclassification as a function of subspaces intersection and number of subspaces.

#Subspace %intersect	2					3					4				
	50%	60%	70%	80%	90%	50%	60%	70%	80%	90%	50%	60%	70%	80%	90%
SSC	0.10	0.50	2.68	8.0	33.8	0.13	1.20	3.98	13.3	42.0	0.31	1.30	6.19	18.4	50.9
S ³ C	0.03	0.40	2.38	7.63	30.9	0.15	1.10	3.98	13.3	42.0	0.30	1.23	5.94	18.1	50.5
Prob SSC	0.03	0.25	1.58	7.25	24.9	0.08	0.65	3.23	11.7	34.8	0.11	0.66	4.41	14.4	42.4

TABLE II: Intersection %SSR error as a function of subspaces intersection and number of subspaces.

#Subspace %intersect	2					3					4				
	50%	60%	70%	80%	90%	50%	60%	70%	80%	90%	50%	60%	70%	80%	90%
SSC	8.03	19.3	22.6	28.6	34.8	11.8	17.7	28.3	36.0	53.4	12.5	25.9	28.3	48.6	60.4
S ³ C	3.64	13.3	15.9	21.1	37.8	5.17	8.68	21.8	27.3	47.9	4.93	13.2	16.1	42.0	57.4
Prob SSC	3.04	10.5	17.7	25.3	35.7	3.93	8.90	22.8	32.2	46.3	5.81	19.2	19.5	43.6	56.6

TABLE III: Average %misclassification errors on Extended Yale B Dataset [15].

# Subjects %Err	2		3		5		8		10	
	Average	Median	Average	Median	Average	Median	Average	Median	Average	Median
LRR	6.74	7.03	9.30	9.90	13.94	14.38	25.61	24.80	29.53	30.00
LRSC	3.15	2.34	4.71	4.17	13.06	8.44	26.83	28.71	35.89	34.84
LatLRR	2.54	0.78	4.21	2.60	6.90	5.63	14.34	10.06	22.92	23.59
SSC	1.87	0.00	3.35	0.78	4.32	2.81	5.99	4.49	7.29	5.47
S ³ C	1.27	0.00	2.71	0.52	3.41	1.25	4.15	2.93	5.16	4.22
Soft S ³ C	0.76	0.00	0.82	0.52	1.32	1.25	2.14	1.95	2.40	2.50
Prob SSC	0.48	0.00	0.77	0.52	1.23	0.93	2.08	1.26	2.14	2.19

TABLE IV: Average %misclassification on Hopkins dataset.

%Err	2 Motions		3 Motions		Total	
	Avg	Median	Avg	Median	Avg	Median
LRR	3.76	0.00	9.92	1.42	5.15	0.00
LRSC	2.57	0.00	6.62	1.76	3.47	0.09
SSC	1.95	0.00	4.94	0.89	2.63	0.00
S ³ C	1.73	0.00	5.50	0.81	2.58	0.00
Soft-S ³ C	1.65	0.00	4.27	0.61	2.24	0.00
Prob SSC	1.57	0.00	3.60	0.58	2.13	0.00

allows us to track the assignment of points in the same clusters, and hence delay hard assignments until later iterations, when more confidence is gained. This is possible because, at each iteration, the method splits the points into two groups of *certain* and *uncertain*, allowing the latter group’s association to be delayed until later iterations when the association probabilities become higher. A direct advantage of this delayed association is that the method performs better when subspaces are highly overlapping (i.e. high intersection of bases). The results on both synthetic and real data confirm these advantages.

REFERENCES

- [1] J. Yan and M. Pollefeys, “A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate,” in *Computer Vision—ECCV 2006*. Springer, 2006, pp. 94–106.
- [2] K.-C. Lee, J. Ho, and D. Kriegman, “Nine points of light: Acquiring subspaces for face recognition under variable lighting,” in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I–I.
- [3] K. Huang, Y. Ma, and R. Vidal, “Minimum effective dimension for mixtures of subspaces: A robust gpca algorithm and its applications,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. II–II.
- [4] P. Tseng, “Nearest q-flat to m points,” *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 249–252, 2000.
- [5] Y. Ma, H. Derksen, W. Hong, and J. Wright, “Segmentation of multi-variate mixed data via lossy data coding and compression,” *IEEE Trans. on PAMI*, vol. 29, no. 9, pp. 1546–1562, 2007.
- [6] G. Liu, Z. Lin, and Y. Yu, “Robust subspace segmentation by low-rank representation,” in *Proc. of 27th Int. Conf. on machine learning (ICML)*, 2010, pp. 663–670.
- [7] G. Liu and S. Yan, “Latent low-rank representation for subspace segmentation and feature extraction,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1615–1622.
- [8] E. Elhamifar and R. Vidal, “Sparse subspace clustering: Algorithm, theory, and applications,” *IEEE trans. on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [9] D. Verma and M. Meila, “A comparison of spectral clustering algorithms,” *University of Washington Tech Rep UWCE030501*, vol. 1, pp. 1–18, 2003.
- [10] X. Guo, “Robust subspace segmentation by simultaneously learning data representations and their affinity matrix,” in *IJCAI*, 2015, pp. 3547–3553.
- [11] C.-G. Li, C. You, and R. Vidal, “Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2988–3001, 2017.
- [12] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [13] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.
- [14] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. S. Huang, “Incremental spectral clustering by efficiently updating the eigen-system,” *Pattern Recognition*, vol. 43, no. 1, pp. 113–127, 2010.
- [15] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Trans. PAMI*, vol. 23, no. 6, pp. 643–660, 2001.
- [16] M. Soltanolkotabi, E. J. Candes *et al.*, “A geometric analysis of subspace clustering with outliers,” *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.
- [17] E. Elhamifar and R. Vidal, “Sparse subspace clustering,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2790–2797.
- [18] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. PAMI*, vol. 31, no. 2, pp. 210–227, 2009.
- [19] R. Vidal and P. Favaro, “Low rank subspace clustering (lsrc),” *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.
- [20] R. Tron and R. Vidal, “A benchmark for the comparison of 3-d motion segmentation algorithms,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.