Subspace clustering without knowing the number of clusters: A parameter free approach

Vishnu Menon, Gokularam M, Sheetal Kalyani
Department of Electrical Engineering, Indian Institute of Technology Madras
Chennai, India - 600036
Email: ee16s301@ee.iitm.ac.in, ee17d400@smail.iitm.ac.in, skalyani@ee.iitm.ac.in,

***Abstract*—Subspace clustering, the task of clustering high dimensional data when the data points come from a union of subspaces, is one of the fundamental tasks in unsupervised machine learning. Most of the existing algorithms for this task require prior knowledge of the number of clusters along with few additional parameters which need to be set or tuned apriori according to the type of data to be clustered. In this work, a parameter free method for subspace clustering is proposed, where the data points are clustered on the basis of the difference in the statistical distributions of the angles subtended by the data points within a subspace and those by points belonging to different subspaces. Given an initial fine clustering, the proposed algorithm merges the clusters until a final clustering is obtained. This, unlike many existing methods, does not require the number of clusters apriori. Also, the proposed algorithm does not involve the use of an unknown parameter or tuning for one. A parameter free method for producing a fine initial clustering is also discussed, making the whole process of subspace clustering parameter free. The comparison of the proposed algorithm's performance with that of the existing state-of-the-art techniques in synthetic and real data sets shows the significance of the proposed method.**

I. Introduction

Data Clustering is the problem of categorizing entities in the given dataset into groups called *clusters* so that the entities in the same cluster are more 'similar' than those from different clusters. A comprehensive study of clustering algorithms is provided in [1]. Very often, the dataset comprises points from a Euclidean space, and the clustering problem reduces to finding the groups which are hidden among those vectors. In most techniques, distance measures are used as a similarity metric for clustering [1]. However, the conventional distance measures become unreliable in high dimensions. In a high dimensional space, the data points are sparsely located. It is shown in [2] that the distance between any two high dimensional points becomes equal as the dimension $n \to \infty$. Thus, most clustering algorithms which perform reasonably well in lower dimensions, fail in high dimensions.

Over the years, several algorithms [3] were developed for clustering data of large dimensions. In many practical scenarios, the high dimensional data points are not uniformly distributed throughout the space but lie approximately in low dimensional structure [4]. For example, the images of a face under different lighting conditions approximately lie in 9-dimensional subspace, even though they have a very large number of pixels [5]. Principal component analysis (PCA) [6] is a popular technique to retrieve a low dimensional linear subspace in which the high dimensional data points are concentrated. However, when there are multiple categories in the dataset, it is not appropriate to assume that the points lie in a single low dimensional subspace. For instance, if we have images of several faces under varying illumination conditions, then data will be lying in a union of multiple 9-dimensional subspaces. *Subspace Clustering* addresses this problem by grouping data points such that each group shall contain points from a single subspace of a lower dimension [7].

Subspace clustering is used extensively for image representation and compression [8] and computer vision problems like motion segmentation [9], face clustering [10], image segmentation [11] and video segmentation [12]. It also finds applications in other fields, including hybrid system identification [13], gene expression analysis [14], metabolic screening of new-borns [15], recommendation systems [16] and web text mining [17]. Subspace clustering algorithms can be classified into four main types [7]: (i) algebraic, (ii) iterative, (iii) statistical, (iv) spectral clustering-based. Algebraic techniques (like Generalized PCA [12]) assume that the data is noise-free and lie perfectly in the union of subspaces [7]. Sometimes they can be extended to handle moderate amounts of noise [18]. Iterative methods (like Median K-Flats (MKF) [19]) alternate between assigning points to subspaces and recovering subspaces from each cluster. Statistical methods (like Agglomerative Lossy Compression (ALC) [20]) make assumptions about the generative model for the data.

Spectral clustering-based techniques have gathered a lot of attention in recent years. These methods take a two-stage approach: finding the 'affinity matrix' and then performing spectral clustering [21] on it. Each entry in the affinity matrix (sometimes referred to as graph) denotes similarity between the corresponding pair of points. The difference between different spectral clustering-based techniques is how the affinity matrix is obtained. In recent years, affinity matrix is obtained using the 'self-representation' of each data point with respect to all the other data points [22], [23]. If the data points $\mathbf{m}_i$ are arranged as columns of the matrix $\mathbf{M}$, then the self-representation is given by $\mathbf{M} = \mathbf{MZ}$ such that $\mathbf{Z}_{ii} = 0$. After obtaining such $\mathbf{Z}$, $abs(\mathbf{Z}) + abs(\mathbf{Z}^T)$ is used as the affinity matrix (where $abs(\cdot)$ takes the absolute value of each entry in the matrix). Several techniques have been developed based on this idea. Sparse self-representation enforces the columns of $\mathbf{Z}$ to be sparse. ℓ_1-minimization (as in Sparse Subspace Clustering (SSC) [22]) or Orthogonal Matching

Pursuit (as in SSC-OMP [24] [25]) can be used to obtain such sparse representation. Least square regression (LSR) [26] uses least-squares representations. Elastic net Subspace Clustering (EnSC) [27] provides a mixture of ℓ_1 and ℓ_2 regularizations to obtain self-representations. Few other techniques utilize low-rank self-representation like Low-Rank Recovery (LRR) [23] and Low-Rank Subspace Clustering (LRSC) [28]. Low-Rank Sparse Subspace Clustering (LRSSC) [29], [30] imposes low-rank constraint as well as sparsity constraint on the self-representation matrix. Another work [31] uses block diagonal self-representation (BDR) and performs better than several existing approaches.

There also exist several agglomerative hierarchical algorithms [20], [32], [33] for subspace clustering. Agglomerative (or bottom-up) hierarchical methods start with a large number of fine clusters and merge them progressively until a stopping criterion is reached. Agglomerative Lossy Compression (ALC) [20], which is also a statistical subspace clustering method, finds the clustering that minimizes coding length needed to fit the data points with a Gaussian mixture. A recent work, [34] provides a new approach called Innovation Pursuit, which is an iterative method but can be integrated with spectral clustering to provide a new class of spectral clustering-based techniques. Currently, neural network-based clustering approaches are gaining popularity [35]. Especially, auto-encoder architecture [36] is used to obtain sparse [37] and low rank [38] representation for subspace clustering. These techniques can recover non-linear low dimensional structures underlying the data.

Recently, the distributions of angles between data points [39] have been used in [40] to develop a parameter free technique for outlier detection in high dimensions. While some previous works in subspace clustering [41]–[44] rely on the statistical distribution of angles, they do so only through the use of mean and all these works involve use of prior knowledge of the number of clusters (like in [43]) and/or involve the prior setting of one or more parameters (like in [44]). In contrast, the proposed work utilizes the entire distribution of angles, providing improved performance while also avoiding the need to tune any parameters. The proposed algorithm exploits the difference in the statistical distributions of angles subtended by the points within a subspace and of angles subtended by the points from different subspaces for achieving parameter free clustering.

A. Motivation

Many clustering algorithms require the user to supply the number of clusters to be formed beforehand. In many situations, fixing the number of clusters apriori is not a good choice, especially when the knowledge about the dataset is limited. For example, in gene expression datasets, the number of clusters to be prefixed is not so clear [45]. There are some clustering algorithms which require one or several parameters, if not the number of clusters. Self-representation based techniques require regularization parameters [22] to be set along with the number the subspaces. Even neural network-based clustering methods require the setting of several hyperparameters. There are also methods which tune for unknown parameters in the model - for instance, λ-means clustering [46] tunes for the parameter λ in DP-means algorithm [47], a general clustering algorithm used to cluster data generated by Dirichlet Process. Similarly, ALC [20] doesn't need the number of clusters apriori but requires the user to provide the distortion level ϵ. Different ϵ will result in a different number of clusters in the output clustering, and hence we need to tune for ϵ for the data in hand. When an algorithm requires one or more free parameters, the user has to set them using either cross-validation or prior knowledge about the dataset. However, parameter tuning [48] is a difficult task, and any incorrect tuning of parameters would result in huge performance degradation.

There are several techniques in the literature to determine the number of clusters for conventional distance-based clustering of low-dimensional data [49]–[52]. Several approaches were proposed for estimating the number of subspaces, and these estimates can then be used as an input for the subspace clustering algorithm. In [23], the authors suggested a way to obtain the number of subspaces by soft thresholding the singular values of the Laplacian matrix of the affinity matrix. But this approach needs to set a threshold τ. In [41], it is suggested to estimate the number of subspaces by looking for the maximum separation between successive eigenvalues of the Laplacian matrix. Though this eigen-gap heuristic approach doesn't involve any threshold, the technique is still dependent on the parameters which were used to obtain the affinity matrix. Also, this heuristic can fail when data points are noisy, and subspaces are closer [21].

Recently, parameter free approaches have been developed in the areas of high dimensional outlier detection [40], sparse signal recovery [53] [54], robust regression [55], and these were shown to have results comparable with those which use the explicit knowledge about the parameters. Hence, we look for a parameter free method for subspace clustering.

B. Contributions

Given the high-dimensional data points coming from the union of several low-dimensional subspaces, we propose an algorithm to achieve a clustering without the knowledge of the true number of clusters, L. This essentially consists of two steps. First, we start with an initial clustering with a large number of clusters such that each cluster is likely to contain points from one subspace. In the second stage, the clusters are merged to arrive at a final clustering. Given an initial clustering, we propose a method based on the statistical distance between the distributions of the angles subtended by the data points to find the final clustering without having to prefix the number of clusters. This makes the proposed algorithm an agglomerative hierarchical method. We also suggest a parameter free approach for initial clustering.

The performance of the proposed algorithm is compared with state-of-the-art subspace clustering algorithms, namely, SSC-ADMM [22], SSC-OMP [25], EnSC [27], TSC [41], ALC [20] and LRR [56] and another recent algorithm namely, BDR-Z [31]. We compare the algorithms in terms of Clustering Error (CE) and Normalized Mutual Information (NMI)

on synthetic as well as real datasets like Gene Expression Cancer RNA-Seq [57] [58], Novartis multi-tissue [59], Wireless Indoor Localization [60] [58], Phoneme [61], MNIST [62], Extended Yale-B [63], [64] and Hopkins-155 [65]. It is observed that the proposed algorithm performs on par with other methods even without the need for the number of clusters or any other parameters and has lower computational complexity.

C. Organization of the paper

The rest of the paper is organized as follows. In Section II, we set up the problem and provide the definitions and notations used in this paper, along with a brief overview of the algorithm. The proposed parameter free algorithm for subspace clustering is introduced in Section III. In Section IV, we provide the analysis of our algorithm under certain assumptions on the data model. Section V provides numerical results on synthetic and real datasets and compares the performance of our algorithm with other existing algorithms. In Section VI, we discuss the utility of the proposed algorithm and its pros and cons in light of all the conducted experiments.

II. Overview and Essential Definitions

The problem that we are addressing in this work is to find the clustering of a dataset comprising of high dimensional points coming from a union of subspaces. Suppose we have N data points $\mathbf{m}_i \in \mathbb{R}^n, \forall i \in \{1,2,\ldots,N\}$ and each $\mathbf{m}_i \in \mathcal{U}_1 \cup \mathcal{U}_2 \cup \ldots \cup \mathcal{U}_L$, where $\mathcal{U}_k$'s, $k \in \{1,2,\ldots,L\}$, $L \ll N$ are subspaces in $\mathbb{R}^n$ with dimensions r_k's respectively. We assume that there are N_k points from the subspace $\mathcal{U}_k$.

Definition 1. *A clustering of the dataset with $K \geq 1$ clusters is defined as $\mathcal{C}_K = \{I_1, I_2, \ldots, I_K\}$, where I_j's are mutually disjoint index sets such that $\forall j = 1,2,\ldots,K$, $I_j \subseteq \{1,2,\ldots,N\}$ and $I_j \neq \emptyset$ with $\bigcup_{j=1}^{K} I_j = \{1,2,\ldots,N\}$. We will call I_j's as constituent clusters. If $i \in I_j$, we say that j is the cluster label of the point $\mathbf{m}_i$.*

Definition 2. *The true clustering of the dataset is defined as the clustering $\mathcal{C}_L^* = \{I_1, I_2, \ldots, I_L\}$, where $\forall j = 1,2,\ldots,L$, $I_j = \{i \mid \mathbf{m}_i \in \mathcal{U}_k$ for some $k \in 1,2,\ldots,L\}$ and $|I_j| = N_k$, where $|\cdot|$ denotes the cardinality of the set. i.e., each constituent cluster contains indices of all the points from a subspace and only the points from that subspace.*

Here, we will be working with angles subtended by high dimensional data points and their distributions. We will be using the normalized data points $\mathbf{x}_i = \frac{\mathbf{m}_i}{\|\mathbf{m}_i\|_2}$, where $\|\cdot\|_2$ denotes the ℓ_2 norm. These points $\mathbf{x}_i \in \mathbb{R}^n, \forall i \in \{1,2,\ldots,N\}$ will lie in the high dimensional hypersphere $\mathbb{S}^{n-1}$. Let θ_{ij} denote the angle between two data points $\mathbf{m}_i$ and $\mathbf{m}_j$, i.e.,

$$\theta_{ij} = \cos^{-1}(\mathbf{x}_i^T \mathbf{x}_j) \qquad \theta_{ij} \in [0, \pi]. \tag{1}$$

In this work, we hypothesize that the angles subtended by the points within the subspace and the angles subtended by the points between subspaces come from different statistical distributions and these distributions can be well approximated in many cases by distinct Gaussians with a different location and scale parameters. This is true for a random subspace model[1] and also holds for many real datasets. The proposed algorithm looks at the statistical distances between the distributions of within-cluster and between-cluster angles, where the distributions are estimated through the available angle samples. The next few definitions correspond to these samples and the related estimates, with $S^{(j)}$, $j = 1,2,\ldots,|S|$ denoting the j^{th} element in a set S.

Definition 3. *The within-cluster angle set for constituent cluster I_k is defined as*

$$W_k = \{\theta_{ij} \mid i,j \in I_k,\ i < j\}. \tag{2}$$

If $|I_k| = t_k$, then $|W_k| = \binom{t_k}{2}$ is the number of unique angles in the set.

Definition 4 (Within-cluster estimates)**.** *Given a within-cluster angle set W_k for I_k, suppose $W_k^t \subseteq W_k$ with $|W_k^t| = t$, then the estimates corresponding to this subset are given as*

$$\begin{aligned} \mu_{w_k t} &= \frac{1}{t}\sum_j W_k^{t,(j)} \qquad \textit{and} \\ \sigma^2_{w_k t} &= \frac{1}{t-1}\sum_j \left(W_k^{t,(j)} - \mu_{w_k t}\right)^2. \end{aligned} \tag{3}$$

Here, $\mu_{w_k t}$ and $\sigma^2_{w_k t}$ are respectively sample mean and sample variance of elements of the set W_k^t and $W_k^{t,(j)}$ is the j^{th} element of the set W_k^t.

Definition 5. *The between-cluster angle set for two constituent clusters I_k and I_l is defined as*

$$B_{kl} = \{\theta_{ij} \mid i \in I_k,\ j \in I_l\}. \tag{4}$$

Clearly, $|B_{kl}| = t_k t_l$, the number of possible cross angles.

Definition 6 (Between-cluster estimates)**.** *Given a between-cluster angle set B_{kl} for I_k and I_l, suppose $B_{kl}^t \subseteq B_{kl}$ with $|B_{kl}^t| = t$, then the estimates corresponding to this subset are given as*

$$\begin{aligned} \mu_{b_{kl} t} &= \frac{1}{t}\sum_j B_{kl}^{t,(j)} \qquad \textit{and} \\ \sigma^2_{b_{kl} t} &= \frac{1}{t-1}\sum_j \left(B_{kl}^{t,(j)} - \mu_{b_{kl} t}\right)^2. \end{aligned} \tag{5}$$

Here, $\mu_{b_{kl} t}$ and $\sigma^2_{b_{kl} t}$ are respectively sample mean and sample variance of elements of the set B_{kl}^t and $B_{kl}^{t,(j)}$ is the j^{th} element of the set B_{kl}^t.

Bhattacharyya distance is a very popular measure used for measuring distances between probability distributions [66]. We use its empirical version [67] as our key metric.

Definition 7. *Given two constituent clusters I_k and I_l, the distance D_{kl} between them is defined as the Bhattacharyya distance [66] between the distribution of angles in W_k and B_{kl}, i.e., $D_{kl} = D_B(\theta_{W_k}, \theta_{B_{kl}})$, where θ_{W_k} is the statistical*

[1] This will be introduced in Section III

distribution of angles in W_k and $\theta_{B_{kl}}$ is the statistical distribution of angles in B_{kl} and $D_B()$ is the Bhattacharyya distance between them. The empirical version used here is defined as

$$d_{kl} = \frac{1}{4}\left[\frac{(\mu_{w_k t} - \mu_{b_{kl} t})^2}{\sigma^2_{w_k t} + \sigma^2_{b_{kl} t}} + \log_e\left(\frac{1}{4}\left[\frac{\sigma^2_{w_k t}}{\sigma^2_{b_{kl} t}} + \frac{\sigma^2_{b_{kl} t}}{\sigma^2_{w_k t}}\right] + \frac{1}{2}\right)\right]. \quad (6)$$

In the algorithm, We start from a fine clustering and merge those clusters which are closest in terms of the empirical Bhattacharyya distance until we reach a final clustering. The closeness is measured in terms of the scores, as defined below.

Definition 8. *Score of a constituent cluster I_j in a clustering $\mathcal{C}_K$ is given by*

$$\eta_j = \min_{l=1,2,\ldots,K,\ l \neq j} d_{jl}. \quad (7)$$

Also, we define the partner of a cluster as the one which produces its score. i.e., if $j' = \underset{l=1,2,\ldots,K,\ l\neq j}{\arg\min} d_{jl}$, then $I_{j'}$ is the partner of I_j.

Definition 9. *Score of a clustering $\mathcal{C}_K$ is given by*

$$\gamma_K = \min_{i=1,2,\ldots,K} \eta_i. \quad (8)$$

Let $i^* = \underset{i=1,2,\ldots,K}{\arg\min}\ \eta_i$ and also let I_{j^*} be the partner of I_{i^*}. Then, we call (I_{i^*}, I_{j^*}) as a mergeable pair of $\mathcal{C}_K$.

The algorithm is explained in detail in Sections III and IV.

Other Notations: $\mathbb{P}(\cdot)$ denotes the probability measure. $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. Let $F_{\mathcal{N}}(\cdot)$ denote the cdf of the standard normal distribution, $\Gamma(\cdot)$ denote the gamma function and χ^2_k denote the standard chi-squared distribution with k degrees of freedom. $F_{\chi^2_k}(\cdot)$ denotes the cdf of χ^2_k distribution and $F_{\chi^2(1,\lambda)}(\cdot)$ denotes the cdf of a non-central χ^2 distribution with 1 dof and the non-centrality parameter λ. $\beta'(a,b)$ denotes beta prime distribution with parameters a and b and $F_{\beta'(a,b)}(\cdot)$ is its cdf. Also, $w.p$ indicates 'with probability'. $\lfloor x \rfloor$ is the floor of x and $\lceil x \rceil$ is the ceiling of x. $O()$ denotes the Big O notation for complexity. $abs(x)$ denotes the absolute value of x.

III. Algorithm

In this section, we will explain the proposed algorithm for clustering. First, we will assume that some process gives us an initial fine clustering $\mathcal{C}_P$ and develop an algorithm for merging. Then, we will discuss a method that will select the appropriate clustering from the set of outputs after the merging process such that the final clustering estimate is close to the true clustering. We will also discuss possible initial clustering methods. Here, we give a theoretical framework which forms the basic idea of our proposal. The exact definitions of the distances we used and the derivation of the thresholds can be found in Section IV.

A. Algorithm for merging

Suppose we have a fine clustering $\mathcal{C}_P$, with $P \gg L$. The proposed algorithm runs through from P to 2, by merging clusters and then selects the appropriate clustering through methods described later. First, we will see the merging process starting with $K = P$.

1. Let the current clustering be $\mathcal{C}_K$ with constituent clusters $I_1, I_2, \ldots, I_K$. For each I_j, calculate the distances between constituent clusters as per Definition 7 and then for each I_j, find its score η_j using (7) and also get their respective partners.
2. Calculate the score of the clustering γ_K as in (8) and find the mergeable pair in the current clustering $\mathcal{C}_K$. Let the mergeable pair be (I_{k_1}, I_{k_2}).
3. Merge clusters I_{k_1} and I_{k_2} and form the new clustering $\mathcal{C}_{K-1}$ with $K-1$ constituent clusters.
4. Repeat steps $1-3$ until $K = 2$.

In this merging, we form a series of clusterings $\mathcal{C}_P, \mathcal{C}_{P-1}, \ldots, \mathcal{C}_2$ where each subsequent clustering is formed by merging the mergeable pair in the previous clustering, or in other words we simply combine the closest clusters in terms of the distance between distributions of the within-cluster angles and between-cluster angles.

The intuition behind this merging process is as follows. We hypothesize that the angles between points from the same subspace come from one statistical distribution, and the angles between points coming from different subspaces follow a different distribution. A theoretical treatment of this hypothesis and the motivation behind it is provided in Section IV. This hypothesis implies that, when there are constituent clusters with points from the same subspace in them, the statistical distance between the distributions of angles within the constituent cluster and between the constituent clusters would be close to 0, provided we have enough angle samples in each set.

To clarify, suppose we look at the clustering $\mathcal{C}_P$. Take the constituent cluster I_1 and suppose that I_k also contains only points from the same subspace, then the angles in W_1 and B_{1k} come from the same distribution. This means that we have a low value close to 0 for d_{1k}, which is the measure of divergence between the distributions of angles in W_1 and B_{1k}. On the other hand, if I_j is a constituent cluster with points from a different subspace to those in I_1, then the angles in W_1 and B_{1j} come from different distributions, and hence the distance d_{1j} will be high and bounded away from 0. We calculate η_1 as the minimum distance made by the constituent cluster I_1. When the clustering contains at least another constituent cluster with only points from the same subspace as is the case above, then the partner of that constituent cluster will be one of those clusters with only points from the same subspace. i.e., suppose for the above case, let I_k, I_l and I_m have only points from the same subspace as in I_1, then the partner of I_1 will be either I_k, I_l or I_m. Hence, when we look at the bigger picture of a clustering, whenever a clustering contains at least a pair of constituent clusters having only points from the same subspace, the mergeable pair shall contain only points from the same subspace and the clustering score, γ_K will be very close to 0. In each step, we merge the closest clusters in terms of d_{kl}, which ensures that points from the same subspace get clustered together as we keep merging. The merging process is summarized in Algorithm 1.

Algorithm 1 Procedure for Merging

Input: Initial clustering $\mathcal{C}_P$, normalized data matrix $\mathbf{X}$.
Initial calculation: Calculate $\theta_{ij}\ \forall i,j = 1,2,\ldots,N$ as in (1).
Initialization: $K = P$.

1: Calculate γ_K for the current clustering $\mathcal{C}_K$ as in (8).
2: Merge the mergeable pair in $\mathcal{C}_K$ and form $\mathcal{C}_{K-1}$.
3: Repeat steps $1-2$ until $K = 2$.

Output: Clusterings $\mathcal{C}_P, \mathcal{C}_{P-1}, \ldots, \mathcal{C}_2$.

B. Selecting optimal clustering

If we start with a clustering $\mathcal{C}_P$ such that each constituent cluster in $\mathcal{C}_P$ only contains indices of points from one subspace, then through the merging process described in the previous subsection, at some point, we will arrive at a clustering $\mathcal{C}_{\hat{L}}$. At this point, no two constituent clusters have points from the same subspace, which means that no cluster pair can form a distance d_{ij} that is close to 0. Hence, the cluster score $\gamma_{\hat{L}}$ will take a jump from a value close to zero to a higher magnitude. This is what we try to exploit in our algorithm to find $\hat{L}$.

Also note that here we compute the statistical distances using the angle samples we have in the within-cluster and between-cluster sets, i.e., for constituent clusters I_i and I_j, θ_{W_i} and $\theta_{B_{ij}}$ are estimated distributions and d_{ij} is the empirical Bhattacharyya distance. Suppose we have t angle samples, we can state the following on the behaviour of γ_K:

a) Suppose for a clustering $\mathcal{C}_K$, I_i and I_j contain only points from the same subspace $\mathcal{U}_a$ such that the angles between the points in $\mathcal{U}_a$ are distributed with a distribution $p_{\mathcal{U}_a}$, then as the number of angle samples $t \to \infty$, $\theta_{W_i} \to p_{\mathcal{U}_a}$ and $\theta_{B_{ij}} \to p_{\mathcal{U}_a} \Rightarrow d_{ij} \to 0$. In other words, d_{ij} will be very close to 0 if one has a large number of angle samples for estimating the distribution.
b) Hence in $\mathcal{C}_K$, for I_i, if $\exists I_j$ as described in a), then $\eta_i \to 0$ given a large number of angle samples. This is because of the definition of η_i, which is the minimum distance that a constituent cluster makes.
c) From the above, for a clustering $\mathcal{C}_K$, if there exists at least one such I_i, I_j pair as a), then $\gamma_K = \min_{l=1,2,\ldots,K} \eta_l \to 0$ at large enough number of angle samples.
d) Suppose that such a pair as a) does not exist in $\mathcal{C}_K$, i.e., for any i and j, I_i and I_j contain points from different subspaces, say $\mathcal{U}_a$ and $\mathcal{U}_b$ respectively. Then, as $t \to \infty$, $\theta_{W_i} \to p_{\mathcal{U}_a}$ and $\theta_{B_{ij}} \to q$, where q is the distribution of angles between points from different subspaces. Then, as t increases, $d_{ij} \to D_B(p_{\mathcal{U}_a}, q)$ and since these are different distributions, d_{ij} is bounded away from 0 $\forall i,j$, which in turn leads to γ_K being bounded away from 0.
e) Also note that, whenever I_i and I_j contains a mixture of points from different subspaces, we cannot make an assertion on the nature of distributions in each set and hence the distance metric d_{ij} becomes unpredictable.

Hence while merging, suppose we arrive at a true clustering at $K = \hat{L}$, then there exists no more mergeable pair which have points from the same subspace. So, we can state the following assuming that $t \to \infty$.

- For $K > \hat{L}$, $\gamma_K \to 0$.
- For $K = \hat{L}$, γ_K is bounded away from zero.
- For $K < \hat{L}$, γ_K behaviour is unknown.

Now, we will describe the method that can identify $\hat{L}$ from the calculated γ_K's. This method is essentially a thresholding scheme, which uses a threshold ζ_K on the scores γ_K and looks for the first crossing of this threshold as our final clustering. The algorithm is summarized in Algorithm 2.

Algorithm 2 Thresholding with ζ_K

Input: $\mathcal{C}_P, \mathcal{C}_{P-1}, \ldots, \mathcal{C}_2$ and associated γ_K's.

1: For each K, calculate ζ_K as in (16).
2: $\hat{L} = \max\{K \mid \gamma_K > \zeta_K\}$.
3: $\hat{\mathcal{C}}_{\hat{L}} = \mathcal{C}_{\hat{L}}$.

Output: Estimated optimal clustering $\hat{\mathcal{C}}_{\hat{L}}$.

In Section IV, we formulate a threshold ζ_K theoretically, which with a very high probability ensures that the scores $\gamma_K \leq \zeta_K$ whenever the mergeable pair in $\mathcal{C}_K$ contains only points from the same subspace, i.e., through this threshold we ensure that whenever $K > \hat{L}$, the scores $\gamma_K \leq \zeta_K$. So, $\hat{L}$ is the first instance where the score exceeds the threshold, and we find that as in step 2 in Algorithm 2. The formulation of the high probability threshold involves a statistical analysis on the distances and is discussed in Section IV-B, where we derive the threshold ζ_K. For this derivation, we use a data model described in Section IV-A, where we also provide the motivation behind the model.

C. Initial Clustering

The algorithm discussed previously requires as input an initial fine clustering $\mathcal{C}_P$ with $P \gg L$. Here, we discuss possible methods to get this clustering from data. One way is to use well-known algorithms for clustering, setting them for over-estimating the number of clusters. For instance, one could use K-Means clustering by setting a large value for K. Here we will use a method based on the closeness of points while keeping a minimum of 3 points per cluster, which is the only assumption made. The following steps describe the algorithm for initial clustering.

1. For each data point, find the two closest points in terms of the acute angles between them, i.e. $\cos^{-1}(abs(\mathbf{x}_i^T\mathbf{x}_j))$. Let us call them allies of a point.
2. Starting from any point chosen randomly, form clusters with the point and it's two allies. Run through the points forming such clusters, avoiding repetition of points in clusters. Here, a new cluster is formed only when a point and both its allies are not already allocated to a cluster. Hence, in this run, a lot of points go unallocated.
3. For all the points left out, add them to the cluster of its closest ally, if it has a cluster allocated or else add them to the cluster of its second ally. After this run, all points are added to some cluster.

This forms our initial clustering, with at least 3 points in each constituent cluster. Please note that this initialization scheme does not guarantee that the initial clustering has

constituent clusters with points only from the same subspace. This method of initial clustering is similar to [68] and could be improved upon as part of future work.

IV. Theoretical Analysis

To provide theoretical results, we require a model for data points from different subspaces. In this section, we will describe the assumptions that we make on the data model and the motivation behind this assumption. Then, under this model, we analyse the algorithm theoretically and derive the threshold ζ_K. Here we show theoretically that, under the Gaussian assumption on the nature of distributions of angles, the score $\gamma_K \leq \zeta_K$ with high probability, whenever a mergeable pair exists in the current clustering. This will result in the algorithm achieving a true clustering with high probability.

A. Assumptions used and their motivations

The distribution of angle between two high dimensional points is studied in detail in [39]. When independent points are chosen uniformly at random from $\mathbb{S}^{n-1}$, the distribution of the angles between any two of them is approximately Gaussian with mean $\frac{\pi}{2}$ and variance $\frac{1}{n-2}$ [39]. Even if all the points are independent, the angles which involve the same point are only pairwise independent [69]. Let us look at the following model:

Model 1. *The subspaces $\mathcal{U}_i$'s, $i = 1, 2, \ldots, L$ are chosen uniformly at random from the set of all r_i dimensional subspaces respectively and the normalized points in each subspace are sampled uniformly at random from the $\mathcal{U}_i \cap \mathbb{S}^{n-1}$.*

Note that Model 1 is the fully-random model as used in [41], [70]. In this model, we can use results proved in [39], [40] to state the following:

Lemma 1. *In Model 1, let $\mathcal{C}_L^*$ be a true clustering.*

a) *When $i, j \in I_k$, with I_k corresponding to the subspace $\mathcal{U}_a$, θ_{ij}'s are identically distributed with an expected value of $\frac{\pi}{2}$ and its pdf is given by $h_{r_a}(\theta) = \frac{1}{\sqrt{\pi}} \frac{\Gamma\left(\frac{r_a}{2}\right)}{\Gamma\left(\frac{r_a-1}{2}\right)} (\sin\theta)^{r_a-2}, \quad \theta \in [0, \pi]$.*

b) *When $i \in I_k$, $j \in I_l$, then θ_{ij}'s are identically distributed with an expected value of $\frac{\pi}{2}$ and its pdf is given by $h_n(\theta) = \frac{1}{\sqrt{\pi}} \frac{\Gamma\left(\frac{n}{2}\right)}{\Gamma\left(\frac{n-1}{2}\right)} (\sin\theta)^{n-2}, \quad \theta \in [0, \pi]$. Also, θ_{ij} converges in distribution to $\mathcal{N}\left(\frac{\pi}{2}, \frac{1}{n-2}\right)$ as $n \to \infty$ and the rate of convergence is of $O\left(\frac{1}{n}\right)$.*

Proof. Please refer to Appendix A. □

Also, we state the following remark.

Remark 1. *The pdf $h_p(\theta)$ from Lemma 1 can be approximated by the pdf of Gaussian distribution with mean $\frac{\pi}{2}$ and variance $\frac{1}{p-2}$, specifically for $p \geq 5$ as validated in [39].*

Through Lemma 1 and Remark 1, we can see that in Model 1, the angles between points coming from the same subspace and the angles between points of different subspaces both follow Gaussian distribution with different variances. Model 1 gives us a framework to work with. However, it is too restrictive. Hence, we generalize this model: We assume that the angles between points in the same subspace are approximately Gaussian distributed with some mean μ_w and variance σ_w^2 and those coming from points of different subspaces also approximately Gaussian distributed with some other mean μ_b and variance σ_b^2, with all of the parameters varying according to the data used and the pair of clusters considered. Essentially, the distributions of angles subtended by points in the same subspace and those from different subspaces have different distributions, both of which can be well approximated by Gaussian distributions. We can see that this assumption holds in many cases. Fig. 1 shows the histogram of within-cluster angles and between-clusters angles in Gene Expression Cancer RNA-Seq dataset [57] [58], which are approximately Gaussian distributed with different parameters[2]. This is the motivation behind making the following assumption in this work.

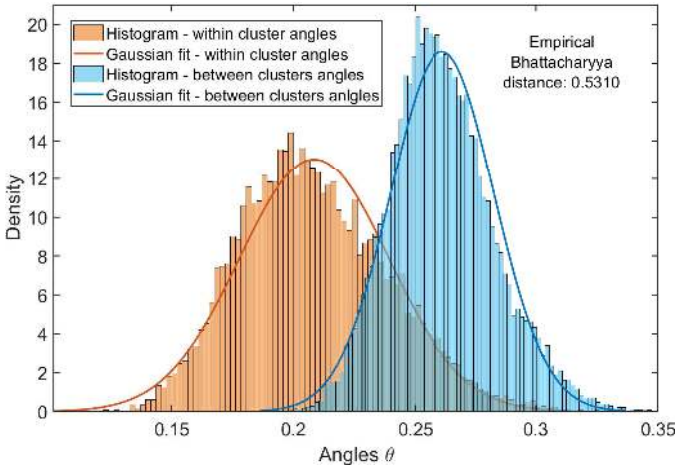


Fig. 1. Distribution of within-cluster angles and between-clusters angles in Gene Expression Cancer RNA-Seq dataset

Assumption 1. *For $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_{N_a} \in \mathcal{U}_a$, the angles between them, θ_{ij}'s are identically distributed as $\mathcal{N}(\nu_a, \rho_a^2)$. For any $\mathbf{x}_i \in \mathcal{U}_a$ and $\mathbf{x}_j \in \mathcal{U}_b$, the angle θ_{ij} is distributed as $\mathcal{N}(\nu_{ab}, \rho_{ab}^2)$. For any i, j, k and l, the angles θ_{ij}, θ_{ik} and θ_{il} are not mutually independent but are pairwise independent, i.e., θ_{ij} and θ_{ik} are independent, so are θ_{ij} and θ_{il}.*

Please note that the model only assumes that the distributions of within-cluster angles and between-cluster angles are Gaussian and does not require the subspaces to be linearly independent. Also, note that trivially the angles are mutually independent if they do not have a common point involved. We assume throughout this work that the angles formed by points within a subspace have a Gaussian distribution and those between points coming from different subspaces have a different Gaussian distribution. As seen in the previous section, the proposed algorithm for finding the final clustering from a given clustering with a large number of constituent clusters is

[2]For this illustration, we have considered the clusters - LUAD and PRAD from the dataset.

based on distances between distributions of angles within and between constituent clusters. We will model the angles using Assumption 1 and derive the threshold ζ_K theoretically in the next subsection.

B. Theoretical behaviour of scores

For theoretical analysis, we will work under Assumption 1. Consider a clustering $\mathcal{C}_K$. Let I_i and I_j be two constituent clusters in $\mathcal{C}_K$. Let $|I_i| = \omega_i$ and $|I_j| = \omega_j$. Here, we will do the following:

i Characterize d_{ij} mathematically looking at its statistical properties when I_i and I_j contain only points from the same subspace $\mathcal{U}_a$ and also when I_i and I_j contain points from different subspaces $\mathcal{U}_a$ and $\mathcal{U}_b$.

ii We will use this to develop the threshold ζ_K, which is a high probability upper bound on γ_K whenever there exists some $I_i, I_j \in \mathcal{C}_K$ which contains only points from the same subspace.

When we use Assumption 1, d_{ij} between constituent clusters I_i and I_j is the Bhattacharyya distance between two normal distributions. We calculate it empirically using angle samples as defined in (6). The properties of this are studied in detail in [67].

Under Assumption 1, we have only assumed pairwise independence of angles and not mutual independence when they involve the same data point. In this section, we will ensure that the estimates are generated by independent angles by designing the subsets W_i^t and B_{ij}^t as such. We want to ensure that the samples used for the estimates $\mu_{w_i t_{ij}}$ and $\mu_{b_{ij} t_{ij}}$ are independent samples and also ensure that $\mu_{w_i t_{ij}}$ and $\mu_{b_{ij} t_{ij}}$ are independent with respect to each other. For this, we have to ensure that we only pick at most two angles formed by a point in these estimates. Lemma 3 in Appendix B designs such subsets $W_i^{t_i}$ and $B_{ij}^{t_j}$ with $t_i = \lfloor \frac{\omega_i}{2} \rfloor$ and $t_j = \min(\omega_i, \omega_j)$, where the samples in these subsets are independent. Let $t_{ij} = \min(t_i, t_j) = \min(\lfloor \frac{\omega_i}{2} \rfloor, \omega_j)$.

Further in the analysis, we will assume that we use only t_{ij} samples each from the sets $W_i^{t_i}$ and $B_{ij}^{t_j}$ for getting our estimates. This helps in simplifying the analysis without compromising on its crux. Let W_i^t and B_{ij}^t be these sets, and $\mu_{w_i t_{ij}}$, $\sigma^2_{w_i t_{ij}}$ and $\mu_{b_{ij} t_{ij}}$, $\sigma^2_{b_{ij} t_{ij}}$ be the corresponding estimates. Let us also denote $X_{ij} = (\mu_{w_i t_{ij}} - \mu_{b_{ij} t_{ij}})$, $Y_{ij} = \sigma^2_{w_i t_{ij}} + \sigma^2_{b_{ij} t_{ij}}$, $Z_{ij} = \sigma^2_{w_i t_{ij}} / \sigma^2_{b_{ij} t_{ij}}$ and also $U_{ij} = X_{ij}^2 / Y_{ij}$ and $V_{ij} = Z_{ij} + \dfrac{1}{Z_{ij}}$. Then $d_{ij} = \frac{1}{4}\Big(U_{ij} + \log_e \Big[\frac{V_{ij}}{4} + \frac{1}{2}\Big]\Big)$. We will first look at the distribution and properties of these estimates.

Lemma 2. *Under Assumption 1:*

a) *If I_i and I_j contain points only from the same subspace $\mathcal{U}_a$, then the estimates $\mu_{w_i t_{ij}}$ and $\sigma^2_{w_i t_{ij}}$ are independent and so too are $\mu_{b_{ij} t_{ij}}$ and $\sigma^2_{b_{ij} t_{ij}}$. Also,*

$$\frac{t}{2\rho_a^2} X_{ij}^2 \sim \chi_1^2 \ \text{ and } \ \sigma^2_{w_i t_{ij}}, \sigma^2_{b_{ij} t_{ij}} \sim \frac{\rho_a^2}{t_{ij} - 1} \chi^2_{t_{ij}-1}. \quad (9)$$

b) *If I_i contain points only from subspace $\mathcal{U}_a$ and I_j from $\mathcal{U}_b$:*

$$\frac{t_{ij}}{\rho_a^2 + \rho_{ab}^2} X_{ij}^2 \sim \chi^2 \left(k = 1, \lambda = t_{ij} \frac{(\nu_a - \nu_{ab})^2}{\rho_a^2 + \rho_{ab}^2} \right),$$
$$\sigma^2_{w_i t_{ij}} \sim \frac{\rho_a^2}{t_{ij} - 1} \chi^2_{t_{ij}-1}, \ \sigma^2_{b_{ij} t_{ij}} \sim \frac{\rho_{ab}^2}{t_{ij} - 1} \chi^2_{t_{ij}-1}. \quad (10)$$

Proof. Please refer to Appendix B. □

Now, we will look at d_{ij}, given in (6) for both cases where the points come from the same subspace and different subspaces.

Theorem 1. *Suppose I_i and I_j contain points from the same subspace $\mathcal{U}_a$ with t_{ij} independent angle samples used for estimating the sample means and variances. Then under Assumption 1,*

$$d_{ij} \leq \frac{1}{\sqrt{t_{ij} - 1}} \qquad w.p \geq 1 - \epsilon_{t_{ij}}. \quad (11)$$

Here, $\epsilon_{t_{ij}} = 2 - F_{\beta'(\frac{1}{2}, t_{ij}-1)}\Big(\frac{t_{ij}}{(t_{ij}-1)^{\frac{3}{2}}}\Big) - F_{\beta'\left(\frac{t_{ij}-1}{2}, \frac{t_{ij}-1}{2}\right)}\Big(\frac{c+\sqrt{c^2-4}}{2}\Big) + F_{\beta'\left(\frac{t_{ij}-1}{2}, \frac{t_{ij}-1}{2}\right)}\Big(\frac{c-\sqrt{c^2-4}}{2}\Big)$, *where* $c = 4\Big(e^{\frac{2}{\sqrt{t_{ij}-1}}} - \frac{1}{2}\Big)$.

Proof. First, look at Y_{ij}, we use the result that if $A_1 \sim \chi^2_{a_1}$, $A_2 \sim \chi^2_{a_2}$, then $A_1 + A_2 \sim \chi^2_{a_1+a_2}$. Using this and (9) in Lemma 2, $\sigma^2_{w_i t_{ij}} + \sigma^2_{b_{ij} t_{ij}} \sim \frac{\rho_a^2}{t_{ij}-1} \chi^2_{2(t_{ij}-1)} \Rightarrow \frac{t_{ij}-1}{\rho_a^2} Y_{ij} \sim \chi^2_{2(t_{ij}-1)}$. Again from (9) in Lemma 2,

$$U_{ij} = \frac{X_{ij}^2}{Y_{ij}} = \frac{2(t_{ij}-1)}{t_{ij}} \frac{A_1}{A_2},$$

where $A_1 \sim \chi_1^2$ and $A_2 \sim \chi^2_{2(t_{ij}-1)}$. Using the result that the ratio of two independent chi-squared random variables follows a beta prime distribution [71], $\frac{A_1}{A_2} \sim \beta'\left(\frac{1}{2}, t_{ij} - 1\right)$. Hence, $\frac{t_{ij}}{2(t_{ij}-1)} U_{ij} \sim \beta'\left(\frac{1}{2}, t_{ij} - 1\right)$. This leads to the following:

$$U_{ij} \leq \frac{2}{\sqrt{t_{ij} - 1}} \quad w.p \quad F_{\beta'(\frac{1}{2}, t_{ij}-1)}\left(\frac{t_{ij}}{(t_{ij}-1)^{\frac{3}{2}}}\right). \quad (12)$$

Let $c = 4\left(e^{\frac{2}{\sqrt{t_{ij}-1}}} - \frac{1}{2}\right)$. Now, we look at:

$$\mathbb{P}\left(\log_e\left(\frac{V_{ij}}{4} + \frac{1}{2}\right) \leq \frac{2}{\sqrt{t_{ij}-1}}\right) \equiv \mathbb{P}\left(V_{ij} \leq c\right).$$

$$\mathbb{P}(V_{ij} \leq c) = \mathbb{P}\left(Z_{ij} + \frac{1}{Z_{ij}} \leq c\right) = \mathbb{P}\left(Z_{ij}^2 - cZ_{ij} + 1 \leq 0\right).$$

Consider $Z_{ij}^2 - cZ_{ij} + 1 \leq 0$. The roots of this quadratic are $(z_0, z_0') = \frac{c \pm \sqrt{c^2-4}}{2}$. Note that $e^{\frac{2}{\sqrt{t_{ij}-1}}} \geq 1 \Rightarrow c^2 \geq 4$. Thus, z_0 and z_0' are real with $z_0' \leq z_0$. Hence, $Z_{ij}^2 - cZ_{ij} + 1 \leq 0 \Rightarrow (Z_{ij} - z_0')(Z_{ij} - z_0) \leq 0 \Rightarrow Z_{ij} \in [z_0', z_0]$. Thus,

$$\mathbb{P}(Z_{ij}^2 - cZ_{ij} + 1 \leq 0) = \mathbb{P}\left(\frac{c-\sqrt{c^2-4}}{2} \leq Z_{ij} \leq \frac{c+\sqrt{c^2-4}}{2}\right).$$

Note that $Z_{ij} = \sigma^2_{w_i t_{ij}}/\sigma^2_{b_{ij} t_{ij}} = A_1/A_2$, where $A_1 \sim \chi^2_{t_{ij}-1}$ and $A_2 \sim \chi^2_{t_{ij}-1}$. Since Z_{ij} is the ratio of two independent chi-squared random variables, $Z_{ij} \sim \beta'\left(\frac{t_{ij}-1}{2}, \frac{t_{ij}-1}{2}\right)$. Thus, $w.p\ \ F_{\beta'\left(\frac{t_{ij}-1}{2}, \frac{t_{ij}-1}{2}\right)}\left(\frac{c+\sqrt{c^2-4}}{2}\right) - F_{\beta'\left(\frac{t_{ij}-1}{2}, \frac{t_{ij}-1}{2}\right)}\left(\frac{c-\sqrt{c^2-4}}{2}\right)$,

$$\log_e\left(\frac{V_{ij}}{4} + \frac{1}{2}\right) \leq \frac{2}{\sqrt{t_{ij}-1}}. \tag{13}$$

Note that $d_{ij} = \frac{1}{4}\Big(U_{ij} + \log_e\left[\frac{V_{ij}}{4} + \frac{1}{2}\right]\Big)$. We know that $\mathbb{P}(A \cap B) \geq \mathbb{P}(A) + \mathbb{P}(B) - 1$. So, combining (12) and (13), $w.p\ \geq 1 - \epsilon_{t_{ij}}$ with $\epsilon_{t_{ij}}$ as defined in the statement,

$$d_{ij} \leq \frac{1}{4}\Big(\frac{2}{\sqrt{t_{ij}-1}} + \frac{2}{\sqrt{t_{ij}-1}}\Big) = \frac{1}{\sqrt{t_{ij}-1}}. \qquad \square$$

Table I gives a numerical perspective of the bound and its probability. As seen, the lower bound on probability increases with t_{ij}.

TABLE I
PROBABILITIES AND BOUNDS IN THEOREM 1

t_{ij}	11	51	101	151
$\frac{1}{\sqrt{t_{ij}-1}}$	0.3162	0.1414	0.1	0.0816
$1-\epsilon_{t_{ij}}$	0.970174	0.999567	0.999980	0.999998

Through the next theorem, we will derive a lower bound for d_{ij}, when I_i and I_j contain points from different subspaces.

Theorem 2. *Suppose, I_i contain points only from subspace $\mathcal{U}_a$ and I_j from $\mathcal{U}_b$ with t_{ij} independent angle samples used for estimating the sample mean and variances. Let $M_{ab} = \frac{|\nu_a - \nu_{ab}|}{\sqrt{\rho_a^2 + \rho_{ab}^2}}$, $R_{ab} = \frac{\rho_a^2}{\rho_{ab}^2} + \frac{\rho_{ab}^2}{\rho_a^2}$ and $\alpha_{t_{ij}} = e^{\frac{4}{\sqrt{t_{ij}-1}}}$. Then,*

$$d_{ij} \geq \frac{1}{\sqrt{t_{ij}-1}} \qquad w.p\ \geq 1 - \delta^{ab}_{t_{ij}}, \tag{14}$$

where

$$\delta^{ab}_{t_{ij}} = 1 - \Big[F_{\chi^2_{t_{ij}-1}}\big((t_{ij}-1)\alpha_{t_{ij}}\big) - F_{\chi^2_{t_{ij}-1}}\big((t_{ij}-1)(2-\alpha_{t_{ij}})\big)\Big]^2 \times \Big[1 - F_{\chi^2(1, t_{ij}M^2_{ab})}\big(t_{ij}\log_e(1+M_{ab})\alpha_{t_{ij}}\big)\Big],\ \textit{when}$$

$$t_{ij} \geq 1 + \frac{16}{(\log_e \psi_{ab})^2}, \tag{15}$$

where,
$\psi_{ab} = \frac{\sqrt{(R_{ab}-2)^2(1+M_{ab})^2 + 32R_{ab}(1+M_{ab})} - (R_{ab}-2)(1+M_{ab})}{8}$.

Proof. Please refer to Appendix B $\square$

Note that the bound in (15) is a very conservative sufficient condition. A numerical perspective can be seen in Table II, where $t_{min} = \left\lceil 1 + \frac{16}{(\log_e \psi_{ab})^2} \right\rceil$. R_{ab} and M_{ab} give a sense of how well the distributions of angles are separated. As expected, when these quantities are large, t_{min} reduces, i.e. with a lesser number of angle samples, we get a larger d_{ij} value. One could also note that, for $0 < M_{ab} < e-1$, the probabilities are not very high at $t_{ij} = t_{min}$, in which case the high probability condition of $M_{ab} = 0$ is applicable since $U_{ij} \geq 0$ in any case.

TABLE II
PROBABILITIES AND BOUND IN THEOREM 2: $t_{min} = \left\lceil 1 + \frac{16}{(\log_e \psi_{ab})^2} \right\rceil$

	$M_{ab}=0$		$M_{ab}=2$		$M_{ab}=3$	
R_{ab}	t_{min}	$1-\delta^{ab}_{t_{min}}$	t_{min}	$1-\delta^{ab}_{t_{min}}$	t_{min}	$1-\delta^{ab}_{t_{min}}$
3	1575	0.994042	50	0.998565	35	0.998918
10	118	0.997716	40	0.998573	35	0.998918
20	68	0.998275	38	0.998440	35	0.998918

Suppose we have a clustering $\mathcal{C}_K$, such that each constituent cluster in $\mathcal{C}_K$ only contains points from the same subspace. Let $S_K^{\mathcal{O}} = \{(i,j) \mid \forall p \in I_i \text{ and } q \in I_j, \mathbf{x}_p \in \mathcal{U}_a \text{ and } \mathbf{x}_q \in \mathcal{U}_b,\ a \neq b,\ i,j \in 1,2,\ldots,K\}$ denote the set of all clustering index pairs such that the points in them belong to different subspaces and let $S_K^{\mathcal{I}} = \{(i,j) \mid \forall p \in I_i \text{ and } q \in I_j,\ \mathbf{x}_p,\ \mathbf{x}_q \in \mathcal{U}_a \text{ for some } a,\ i,j \in 1,2,\ldots,K\}$ denote clustering index pairs such that the points in them belong to one subspace. Let $d_{\mathcal{O}} = \min\limits_{(i,j)\in S_K^{\mathcal{O}}} d_{ij}$ be the minimum inter-subspace distance and suppose $t_{\mathcal{O}}$ angle samples were used for its computation. Let the corresponding closest subspaces be $\mathcal{U}_a$ and $\mathcal{U}_b$. Let $T_K^{\mathcal{I}} = \{t_{ij} \mid (i,j) \in S_K^{\mathcal{I}}\}$ denote the set of the number of independent angle samples used for computation of the distance between cluster pairs which contain points from only one subspace. Under the above notations and assumptions, we can state the following corollary, which is a direct consequence of Theorems 1 and 2 and defines the threshold ζ_K.

Corollary 1 (To Theorems 1 and 2)**.** *Suppose the clustering $\mathcal{C}_K$ as described above, with mergeable pair (I_{i^*}, I_{j^*}), has a non-empty $S_K^{\mathcal{I}}$. Let t_K be the number of independent samples used for estimates in the mergeable pair. If $t_{\mathcal{O}} \geq 1 + \frac{16}{(\log_e \psi_{ab})^2}$ and $t_{\mathcal{O}} \leq$ at least one element in $T_K^{\mathcal{I}}$, then $w.p \geq 1 - \epsilon_{t_K} - \delta^{ab}_{t_{\mathcal{O}}}$, the mergeable pair contains points from the same subspace and $\gamma_K \leq \zeta_K$, when*

$$\zeta_K = \frac{1}{\sqrt{t_K - 1}}. \tag{16}$$

Proof. As per Theorem 1, for any $(i,j) \in S_K^{\mathcal{I}}$, $d_{ij} \leq \frac{1}{\sqrt{t_{ij}-1}}$. From Theorem 2, for the closest inter-subspace cluster pair $d_{\mathcal{O}} \geq \frac{1}{\sqrt{t_{\mathcal{O}}-1}}\ w.p\ \geq 1 - \delta^{ab}_{t_{\mathcal{O}}}$. Also, we have assumed that $t_{\mathcal{O}} \leq$ at least one element in $T_K^{\mathcal{I}}$, which means that there exist some t_{ij} with $(i,j) \in S_K^{\mathcal{I}}$, such that $d_{\mathcal{O}} \geq \frac{1}{\sqrt{t_{ij}-1}}$. Hence, $d_{\mathcal{O}} \geq d_{ij}$ for some $(i,j) \in S_K^{\mathcal{I}}$. So the mergeable pair (I_{i^*}, I_{j^*}) contains points from the same subspace and hence $d_{i^*j^*} \leq \frac{1}{\sqrt{t_K - 1}}$ with probability $1 - \epsilon_{t_K}$ from Theorem 1. Hence, $w.p\ \geq 1 - \epsilon_{t_K} - \delta^{ab}_{t_{\mathcal{O}}}$, the statement is true. $\square$

From the above, it is clear that, at any stage K of the algorithm, if suitable conditions are satisfied, Algorithm 1 merges two clusters from the same subspace and hence the merging process keeps merging constituent clusters with only indices of points from the same subspace until it reaches a true clustering. Through Corollary 1, we have shown that until then

$\gamma_K \leq \zeta_K$ with a high probability. And at this point, since the mergeable pair will contain points from different subspaces, γ_K will be high, and we select this crossover instance as the estimated clustering in Algorithm 2. We demonstrate this behaviour of γ_K in Fig. 2. Here, we have considered Model 1 and the data points are drawn from $L = 6$ subspaces, each of dimension 7 in $\mathbb{R}^{100}$. It can be seen that γ_K is close to 0 when $K > 6$ and when $K = 6$, γ_K is bounded away from 0 and $\gamma_K > \zeta_K$. From the above observations, we can state the following remark:

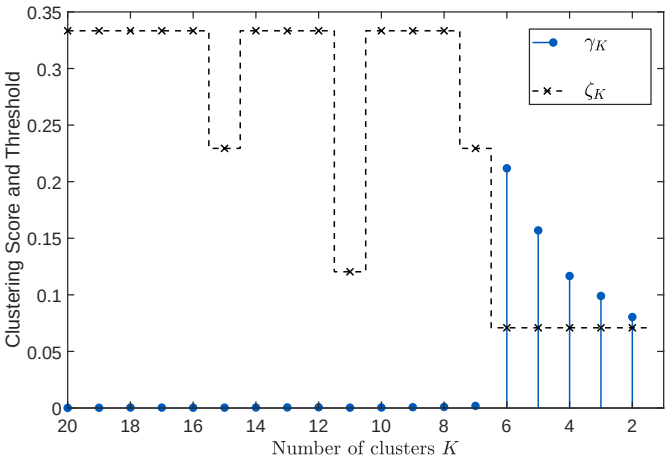


Fig. 2. Clustering score γ_K and threshold ζ_K in Model 1 with $L = 6$ subspaces in $\mathbb{R}^{100}$ each is of dimension 7

Remark 2. *Suppose we are given an initial clustering $\mathcal{C}_P$ such that each constituent cluster in $\mathcal{C}_P$ contains only indices of points from the same subspace. Under suitable conditions as described in Theorems 1 and 2 and Corollary 1, an estimate $\hat{\mathcal{C}}_{\hat{L}}$, which is arrived at by the merging process in Algorithm 1 and the selection process in Algorithm 2, is a true clustering with a high probability.*

Note that the conditions derived are very conservative sufficient conditions for the algorithm to merge correctly at any stage. For instance, consider Model 1 where $M_{ab} = 0$. It has been observed that even for the cases with smaller R_{ab}, the algorithm works perfectly, starting with initial clusters having just 3 or 4 points (i.e., $t_{ij} = 2$). With $n = 100$ and ranks of the subspaces $r = 10$, $R_{ab} = \frac{n-2}{r-2} + \frac{r-2}{n-2} = 12.33$ and Theorem 2 dictates $t_{min} = 96$ to achieve $1 - \delta^{ab}_{t_{min}} = 0.9979$. But with just $t_{ij} = 2$, the algorithm achieves error-free clusterings, as shown in Table III. The error is also minimum in many real datasets as illustrated in Section V. Thus, the algorithm works with minimal error in much harsher conditions than those suggested by Theorems 1 and 2 and Corollary 1. It is extremely difficult to get tighter bounds and conditions as the distribution of empirical Bhattacharyya distance has not been characterized and to do so is beyond the scope of this work.

C. A Note on Complexity

The complexity for computing all the $\binom{N}{2}$ angles between n-dimensional points is $O(N^2 n)$. For P constituent clusters in the initial clustering, one has to compute the statistical distances for all possible combinations. When these are computed as described in the previous sections, one could precompute the sum of angles and the squared sum of angles for all possible combinations, leading to a complexity of $O(P^2)$. In the merging process, using the precomputed values, one can update the distances using simple arithmetic, leading to a complexity of $O(P)$. Hence, the overall complexity is $O(max(N^2 n, P^2))$. Runtime comparisons are provided in Section VI.

V. Numerical Results

In this section, we validate the performance of our algorithm numerically through simulations in synthetic as well as real datasets. We compare the performance with state-of-the-art subspace clustering algorithms - SSC-ADMM [22], SSC-OMP [25], EnSC [27], TSC [41], ALC [20] and LRR [56] and another recent algorithm BDR-Z [31]. The codes for these algorithms are obtained from respective authors. All these algorithms, except ALC, require us to provide an estimate of the number of clusters, L apriori, which we denote as L_{in}. However, as mentioned in [41], we can estimate L by eigen-gap heuristic - we denote L estimated using this technique as $\hat{L}_{eg}$. We also denote the number of clusters estimated by our algorithm as $\hat{L}_{our}$. Throughout the experiments, for the existing algorithms, we have used the default or tuned parameters provided by the authors in their codes, unless stated explicitly. For our algorithm, we use the initial clustering described in Section III-C unless otherwise specified. The best performance in each experiment is given in **boldface**.

A. Metrics for comparison

We compare the performance in terms of *Clustering Error* (CE) and *Normalized Mutual Information* (NMI). Clustering error [41] is defined as the fraction of points misclassified by the algorithm. It is computed as follows. Let c_i and $\tilde{c}_i$, $i = 1, 2, \ldots N$ denote respectively the true cluster label of the point $\mathbf{m}_i$ and the label assigned to it by the algorithm. Then,

$$\text{CE} = \min_{\pi} \left(1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left(c_i, \pi(\tilde{c}_i)\right) \right),$$

where $\mathbb{I}(a, b) = 1$ if $a = b$, 0 otherwise. $\pi(\tilde{c}_i)$ is the one-one reassignment of the label $\tilde{c}_i$, $\pi(\tilde{c}_i) \in \{1, 2, \ldots, L\}$ where $\tilde{c}_i \in \{1, 2, \ldots, \hat{L}\}$. We compute Normalized Mutual Information [72] as

$$\text{NMI} = \frac{\mathcal{I}(\mathcal{C}_L; \hat{\mathcal{C}}_{\hat{L}})}{0.5\big(\mathcal{H}(\mathcal{C}_L) + \mathcal{H}(\hat{\mathcal{C}}_{\hat{L}})\big)},$$

where $\mathcal{H}(\cdot)$ and $\mathcal{I}(\cdot;\cdot)$ respectively denote the empirically computed entropy of the cluster and mutual information between clusters. CE close to 0 and NMI close to 1 are desirable.

B. Results on Synthetic Datasets

1) Random Subspace Models: We first illustrate the results if the data points are from Model 1. It is known that for the points $\mathbf{x}_i$'s to be uniformly distributed in $\mathcal{U}_k \cap \mathbb{S}^{n-1}$, the

TABLE III
PERFORMANCE OF SUBSPACE CLUSTERING ALGORITHMS ON SYNTHETIC DATASETS

		Ours		**SSC-ADMM**		**SSC-OMP**		**EnSC**		**BDR-Z**		**TSC**		**ALC**		**LRR**		**λ-Means**	
		CE	**NMI**	**CE**	**NMI**	**CE**	**NMI**	**CE**	**NMI**	**CE**	**NMI**	**CE**	**NMI**	**CE**	**NMI**	**CE**	**NMI**	**CE**	**NMI**
Subspace-Normal	$L=4$	**0**	**1**	**0**	**1**	0.002	0.997	0.001	0.998	0.022	0.985	**0**	**1**	**0**	**1**	0.066	0.973	NA	NA
	$L=7$	**0**	**1**	0.001	0.999	0.101	0.949	0.062	0.969	0.001	0.999	**0**	**1**	**0**	**1**	0.074	0.969	NA	NA
	$L=10$	**0**	**1**	0.02	0.989	0.11	0.954	0.126	0.95	0.024	0.989	0.02	0.99	**0**	**1**	0.12	0.961	NA	NA
Subspace-Uniform	$L=4$	**0**	**1**	0.023	0.984	0.001	0.998	0.002	0.997	0.025	0.988	**0**	**1**	**0**	**1**	0.034	0.967	NA	NA
	$L=7$	**0**	**1**	0.024	0.987	0.023	0.987	0.024	0.987	0.024	0.987	**0**	**1**	**0**	**1**	0.04	0.977	NA	NA
	$L=10$	**0**	**1**	0.04	0.979	0.02	0.989	0.021	0.989	0.04	0.979	0.02	0.989	**0**	**1**	0.005	0.998	NA	NA
Subspace-Dependent	$L=12$	**0**	**1**	0.078	0.969	0.089	0.955	0.077	0.969	0.124	0.952	0.078	0.97	0.917	0	0.319	0.901	NA	NA
	$L=16$	**0**	**1**	0.023	0.99	0.097	0.962	0.08	0.982	0.143	0.958	0.053	0.981	0.937	0.005	0.186	0.946	NA	NA
	$L=20$	**0**	**1**	0.021	0.99	0.115	0.961	0.09	0.984	0.147	0.956	0.07	0.976	0.95	0.003	0.162	0.965	NA	NA
DP Process	$\rho/\sigma=1$	**0.0007**	**0.975**	0.392	0.007	0.725	0.006	0.725	0.006	0.645	0.006	0.581	0.006	0.281	0	0.282	0.006	0.31	0
	$\rho/\sigma=5$	**0.0006**	**0.954**	0.438	0.007	0.74	0.007	0.74	0.007	0.661	0.006	0.619	0.007	0.327	0	0.328	0.006	0.306	0
	$\rho/\sigma=9$	**0.0008**	**0.972**	0.411	0.008	0.729	0.005	0.729	0.005	0.652	0.005	0.594	0.006	0.321	0	0.323	0.007	0.007	0.949

individual coordinates of $\mathbf{m}_i$'s have to sampled independently from a standard normal distribution [39]. We call this dataset as 'Subspace-Normal'. We also show the results in the random subspace model when we sample individual coordinates of $\mathbf{m}_i$'s from a standard uniform distribution. We call this one as 'Subspace-Uniform'. For both these scenarios, we have taken 1000 points in $\mathbb{R}^{100}$, show results for $L = 4, 7, 10$ with a roughly equal number of points in each L. The dimension of each subspace is chosen to be 10. The results shown in Table III are averaged over 50 trials for each L. For existing algorithms (except ALC) we provide the number of clusters estimated from eigen-gap heuristic as input, i.e., $L_{in} = \hat{L}_{eg}$. For ALC, we set have tried several values for ϵ and $\epsilon = 1$ gave perfect recovery of subspaces in all the trials. We use the same ϵ throughout the remainder of the paper. In Table III, we also show results for dependent subspaces. Here, we choose 100 basis vectors for $\mathbb{R}^{100}$. Then, for each subspace, we choose $r_i = 10$ basis functions randomly from the collection and form $L = 12, 16, 20$ such subspaces. Since $r_i \times L$ is greater than or close to n, we are bound to end up with subspaces which share common bases (more than 1 on many occasions), making them dependent. The data points from these subspaces are formed by the linear combination of the basis where the coordinates are randomly chosen from a standard uniform distribution.

TABLE IV
ERROR ON ESTIMATED NUMBER OF CLUSTERS ON SYNTHETIC DATASETS

	$\lvert L - \hat{L} \rvert$	**Ours**	**Eigen-gap**	**ALC**
Subspace-Normal	Mean	**0**	13.23	**0**
	Median	0	0	0
	Std	0	93.43	0
Subspace-Uniform	Mean	**0**	14.94	**0**
	Median	0	0	0
	Std	0	84.387	0
Subspace-Dependent	Mean	**0**	10.415	13.25
	Median	0	0	13.25
	Std	0	28.84	0

In the subspace model, almost all the algorithms with $\hat{L}_{eg}$ supplied as input perform fairly well, while our algorithm achieves perfect clustering every time, with TSC achieving perfect clustering in most trials. ALC achieves perfect clustering for independent subspaces for all L values but fails for dependent subspaces with the same ϵ value, where it always ends up with one cluster. In dependent subspaces, the performance degrades considerably for many algorithms like LRR and ALC, while it degrades marginally for others. In all the cases, our algorithm clusters perfectly. It is evident that the quality of the estimate of L, $\hat{L}_{eg}$ determines the success of the algorithms. In Table IV, we show the absolute error in the estimate of L, using eigen-gap, ALC, as well as our method, where the values are averaged over all the cases in Table III for all the synthetic datasets except DP process. Our method estimates the number correctly in all trials, while eigen-gap estimates it correctly for most trials, but overestimates the number by a very large value for a few of the trials as seen by the large values for the mean and standard deviation of the error with the median remaining at 0. ALC estimates it correctly with 0 error for independent subspaces. However, it fails for the case of dependent subspaces.

2) Dirichlet Process Model: To illustrate the versatility of our algorithm in adapting to other clustering data models which are not subspace models, we show the results when the

TABLE V
DETAILS OF SOME REAL DATASETS

Dataset	n	N	L
Gene Expression Cancer RNA-Seq [57] [58]	16383	801	5
Novartis multi-tissue [59]	1000	103	4
Wireless Indoor Localization [60] [58]	7	2000	4
Phoneme [61]	256	4509	5
MNIST [62]	784	40000	10
Extended Yale-B [63], [64]	32256	2432	38
Hopkins-155 [65]	30-200	39-556	2,3

data points are obtained from Dirichlet Process (DP) [47]. The results are also summarized in Table III. For each of the listed ρ/σ, we performed 50 trials, generating 1000 points from $\mathbb{R}^{100}$ each time. Here, ρ represent spread of cluster centroids, and σ represent spread of points around their respective centroid. Larger ρ/σ signifies widely separated dense clusters. For this dataset, we also include results from λ-means algorithm [46], a parameter tuning algorithm developed for DP model data. It tunes for the parameter λ in DP-means algorithm [47]. For each ρ/σ, we tune for λ in the first trial and use that value for remaining trials. It is observed that λ-means performs badly when ρ/σ is small. This is because it is a distance-metric based algorithm and at smaller ρ/σ, clusters are not well separated. For the case of DP dataset, we provide the true number of clusters L, as input to all the algorithms instead of eigen-gap estimates. Even then, the performances of existing algorithms are poor as evident from Table III. This shows the lack of adaptability to a non-subspace model for clustering. We can observe that our algorithm performs consistently better for all ρ/σ with mean CE $\leq 0.08\%$ and mean NMI≥ 0.954.

C. Results on Real Datasets

We illustrate the performances of our algorithm and other subspace clustering algorithms on some real datasets. Table V provides the details of the datasets we have used. The first two datasets are gene expression datasets. In such applications, the number of clusters in the dataset may not be known apriori. We have also performed experiments on popular image datasets, namely MNIST [62] and extended Yale-B [63], [64] as well as the motion segmentation problem in Hopkins-155 dataset [65]. The results for the first four datasets are given in Table VI, where all the existing algorithm except ALC are provided with $L_{in} = \hat{L}_{eg}$. For ALC, we provide $\epsilon = 1$. The comparison of estimates of the number of clusters using our algorithm with eigen-gap and ALC for various datasets are given in Table VII.

From Table VI, we can see that our algorithm outperforms all the other algorithms in most of the cases for these datasets. This can be interpreted as follows. The eigen-gap estimates for L, given in Table VII, that we provide as an input to the other algorithms are not necessarily very accurate. This, along with the improper setting of tuning parameters, affects the algorithm performances in Table VI. Our algorithm by virtue of being non-parametric performs well across datasets without tuning or ground truth knowledge. Though our algorithm is developed for high dimensional data, its performance in wireless indoor localization dataset illustrates that we can also use it for low dimensional datasets. These results show the adaptability of the proposed algorithm across datasets of different types from diverse domains.

The results in Table VII reconfirm what we observed in Table IV that eigen-gap heuristic occasionally selects a very large number of clusters. For instance, consider the wireless indoor localization dataset, the eigen-gap heuristic provides 1999 clusters, i.e., it considered each point (except a pair) as a cluster. This results in high CE for the algorithms using that estimate. Here, our algorithm outputs 11 clusters, and the CE is quite low, suggesting that the excess 4 clusters are smaller in size. Also, in Phoneme dataset, our method has recovered the exact number of clusters, and it predicted one additional cluster in Gene Expression Cancer RNA-Seq and Novartis multi-tissue datasets. Hence, our algorithm is better at finding the number of clusters in these datasets.

1) Results on image datasets: We have also performed experiments on three image datasets - the popular MNIST dataset [62], face clustering using the extended Yale-B dataset [63], [64] and Hopkins-155 dataset [65] for motion segmentation. For each image in MNIST, we use extracted features from ScatNet [73] and then projected the features to dimension 500 using PCA and use them for all the algorithms. Due to memory limitations, we performed experiments by taking only 40000 data samples. Motivated from [74], we obtained DSIFT features for extended Yale-B dataset and projected them to dimension 500 using PCA and then obtained results for all the algorithms. We use the dataset as it is for Hopkins-155.

The parameters of EnSC are tuned for MNIST dataset, and those of SSC-ADMM are tuned to Hopkins. Note that eigen-gap heuristic seems to give extremely bad results in estimating the number of clusters (see Table VII) in image datasets. Our method does not require the number of clusters apriori. However, it requires a good set of initial clusters. In Table VIII, all existing algorithms are given the true number of clusters. Otherwise, if the eigen-gap heuristic is used, the results would be extremely poor with CE close to 1 and NMI closer to 0. Hence to be also fair to our method, which does not know the true number of clusters, we provide it with a pure set of initial clusters. For MNIST, we provide 2000 initial clusters, for extended Yale-B we give 266 clusters, and for Hopkins, we use 5 points per initial cluster in each video.

As seen from Table VIII, our method performs at par with the state-of-the-art in MNIST, using ScatNet features. Many algorithms for the large MNIST dataset are really slow. for instance, ALC did not converge even after days of running, and hence those results are not reported. Our algorithm, however, even ran with the whole dataset (70000 points), without a problem. It had a CE of 0.0018 and NMI of 0.9951 for the whole MNIST dataset when provided with 2800 initial clusters. For extended Yale-B using the feature extraction we described earlier, our algorithm outperforms all other methods with the lowest CE and highest NMI, even when other algorithms are provided with the right number of clusters and

TABLE VI
PERFORMANCE OF SUBSPACE CLUSTERING ALGORITHMS ON SOME REAL DATASETS

Algorithm	Gene Expression Cancer RNA-Seq		Novartis multi-tissue		Wireless Indoor Localization		Phoneme	
	CE	NMI	CE	NMI	CE	NMI	CE	NMI
Ours	**0.0087**	**0.9769**	**0.1845**	**0.7247**	**0.1720**	**0.7510**	**0.2790**	0.6222
SSC-ADMM	0.0724	0.9363	0.8058	0.5814	0.9975	0.3085	0.3387	0.7688
SSC-OMP	0.0849	0.8716	0.8155	0.5042	0.9975	0.3085	0.5445	0.2571
EnSC	0.0674	0.9386	0.8058	0.5896	0.9975	0.3085	0.3251	**0.7688**
BDR-Z	0.0587	0.9454	0.8058	0.5814	0.9980	0.3084	0.4400	0.3722
TSC	0.0612	0.9441	0.8058	0.6210	0.9975	0.3085	0.3227	0.7629
ALC	0.7079	0.6298	0.7282	0.3469	0.7500	0	0.8960	0.4390
LRR	0.0637	0.9168	0.6311	0.6557	0.9975	0.3085	0.4424	0.5682

TABLE VII
ESTIMATED NUMBER OF CLUSTERS IN SOME REAL DATASETS

Dataset	Estimated number of clusters		
	$\hat{L}_{our}$	$\hat{L}_{eg}$	$\hat{L}_{ALC}$
Gene Expression Cancer RNA-Seq	6	6	18
Novartis multi-tissue	5	21	1
Wireless Indoor Localization	11	1999	1
Phoneme	**5**	3	45
MNIST	**10**	39997	-
Extended Yale-B	39	2431	269
Hopkins-155 (2 objects)	2.0083	118.32	1.333
Hopkins-155 (3 objects)	2.2	215.68	1.314

TABLE VIII
PERFORMANCE OF SUBSPACE CLUSTERING ALGORITHMS ON IMAGE DATASETS

Algorithm	Metric	Dataset		
		MNIST	Ext. Yale-B	Hopkins-155
Ours	CE	**0.0015**	**0.0074**	0.0938
	NMI	**0.9959**	**0.9979**	0.7406
SSC-ADMM	CE	0.1523	0.0185	**0.0479**
	NMI	0.8540	0.9843	**0.8719**
SSC-OMP	CE	0.0532	0.0888	0.7087
	NMI	0.8784	0.9748	0.0280
EnSC	CE	0.0404	0.0465	0.2176
	NMI	0.9085	0.9803	0.5603
BDR-Z	CE	0.3496	0.0366	0.1531
	NMI	0.5504	0.9874	0.7278
TSC	CE	0.1650	0.0247	0.3735
	NMI	0.8013	0.9884	0.3673
ALC	CE	–	0.3466	0.5900
	NMI	–	0.7942	0.2393
LRR	CE	0.1831	0.3433	0.1246
	NMI	0.8536	0.8810	0.7939

the parameters tuned through grid search. The influence of feature extraction for images on the success of the proposed method is discussed in detail in the next section. For the motion segmentation problem in Hopkins-155, our results are again at par with the state-of-the-art with only SSC-ADMM (which is tuned perfectly for this dataset) outperforming us marginally in terms of CE.

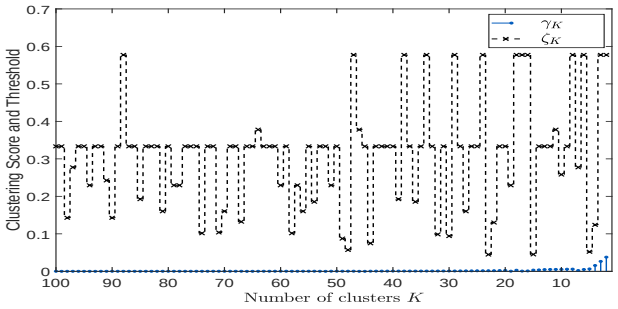
VI. UTILITY OF OUR ALGORITHM

As stated previously, the proposed algorithm is designed such that it can perform parameter free clustering on a dataset, where the data vectors are such that there is sufficient difference in the statistical distribution between angles formed by points within a subspace and between subspaces. As observed from Fig. 2, whenever this assumption holds, there is a drastic jump in γ_K and rapid fall of ζ_K when $K = L$. Thus, the possibility of success of the algorithm, can also be very easily visualized if one were to look at the evolution of γ_K and ζ_K. In a dataset where the algorithm would do well, the γ_K value spikes noticeably at a certain point K where there is also a drastic fall in ζ_K.

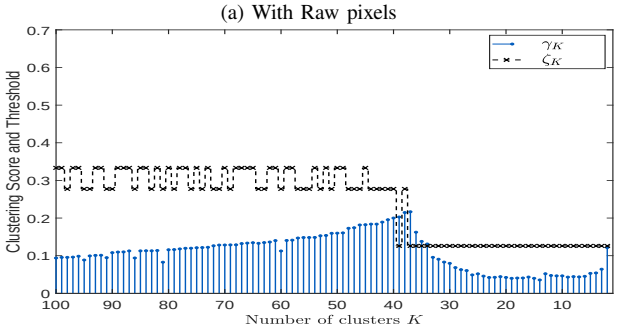
However, if the data are such that the distributions of angles formed by points within a subspace and of those between subspaces are very similar, then the algorithm will fail. There are many datasets, as highlighted in Section V, where the assumption holds approximately, and the algorithm can cluster effectively at high speed, without a hyperparameter. For some other popular datasets like the extended Yale-B, with the feature vector being vectorized image pixel values, γ_K evolves smoothly, and γ_K never crosses ζ_K, as shown in Fig. 3 (a), indicating that the algorithm shall fail if we were to use the

TABLE IX
RUN TIME OF ALGORITHMS ON EXT. YALE-B DATASET

Algorithm	No. of parameters (including L_{in})	Run time (in seconds)
Ours	0	2.5
SSC-ADMM	2	21.3
SSC-OMP	2	4.4
EnSC	3	6.6
BDR-Z	3	444.5
TSC	2	7.6
ALC	1	1397.1
LRR	2	6267.1

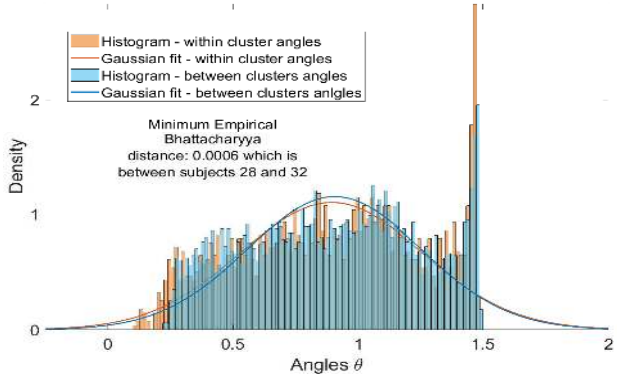


(a) With Raw pixels

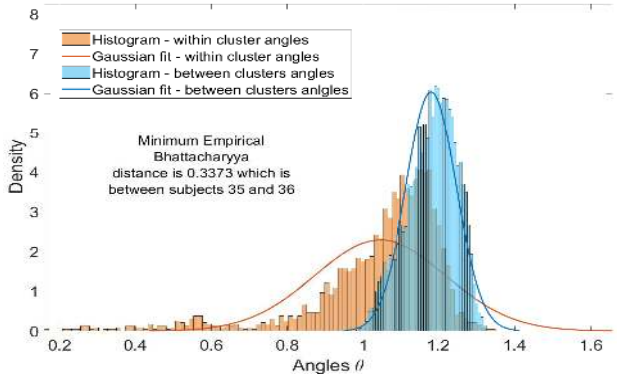


(b) With DSIFT features

Fig. 3. Clustering score γ_K and threshold ζ_K in Ext. Yale-B dataset



(a) With Raw pixels



(b) With DSIFT features

Fig. 4. Distribution of within-cluster angles and between-clusters angles in Ext. Yale-B dataset corresponding to minimum empirical Bhattacharyya distance

raw pixels as data vectors. This is potentially due to the fact that the inter-cluster diversity is less in this dataset, as noticed in [75]. However, a suitable feature extraction technique, like DSIFT used in Section V-C for extended Yale-B, ensures that there is sufficient separation between the statistical distribution of within-cluster and between cluster angles. This can be seen manifested in the behaviour of γ_K, as shown in Fig. 3 (b). The difference in the empirical statistical distributions of within-cluster and between-cluster angles before and after feature extraction for extended Yale-B dataset can be seen in Figs. 4 (a) and (b). There could exist a suitable feature extraction technique for every dataset, which could make the algorithm perform effective clustering in that dataset as well. But we have not pursued that line of research here since it is too domain-specific. We have demonstrated in Section V, the effectiveness of the algorithm in diverse domains and not just in image datasets.

Furthermore, even though our algorithm may fail in some datasets, the possibility of failure can be readily identified by looking at γ_K and ζ_K, without having to know the ground truth. Since the proposed algorithm is computationally efficient and tuning parameter free, one can apply the method to any dataset and see if the current feature vectors can be clustered effectively with the proposed method, with minimal effort by observing the evolution of γ_K and ζ_K. Most of the other state of the art methods involve using tuning parameters, and to set them appropriately, one must have pre-hand knowledge of the ground truth. Even using the ground truth or with training sets, the time it takes to set appropriate tuning parameters is quite high. We provide the run time comparison for a single run of the algorithms in extended Yale-B dataset in Table IX. The second column gives information about the number of parameters required as input to the algorithm. For example, SSC-ADMM requires the number of clusters and one more tuning parameter. All the algorithms are run on the same system for fairness, and all algorithms are provided with the true number of clusters as an input. Note that our algorithm is roughly 8 times faster than SSC-ADMM, 550 times faster than ALC and 2500 times faster than LRR. It is evident that many existing algorithms take more time to obtain clustering, even with the predefined set of parameters. Setting the tuning parameters would take significant multiple of this time unless there is a predefined way to tune parameters other than grid

search and cross-validation.

VII. Conclusions

In this paper, we have proposed a parameter free algorithm for subspace clustering, which distinguishes between the points from different subspaces using the characteristics of the distribution of angles subtended by the points. The algorithm, which works without apriori parameter knowledge, starts with a fine clustering and merges the clusters iteratively until the clustering score crosses a threshold. We have theoretically analysed the algorithm and derived the threshold under an assumption on the data model and also proposed a parameter free initial clustering method. The performance of the proposed algorithm has been studied extensively in both synthetic as well as many real datasets. It has been observed that the proposed method performs on par with other existing methods which use true parameter knowledge, in terms of clustering error and estimated number of clusters and outperforms them in many cases, especially when the true parameters are unknown. In this work, we have used empirical Bhattacharyya distance as a discriminating criterion. However, one can use any other statistical distance provided one can derive the appropriate thresholds. This could be an interesting direction for future research.

Appendix A
Proofs of results in Section IV-A

Proof of Lemma 1. Results in [40] is built on the basis of Lemma 12 from [39] which gives the distribution of angles between randomly chosen points in $\mathbb{S}^{n-1}$. Part a) is straight from Lemma 2 in [40]. Note that the angle between two points from different subspaces in Model 1 is statistically same as that between two points chosen uniformly at random from $\mathbb{S}^{n-1}$ as in Lemma 9 in [40] and part b) follows Lemma 1 in [40]. For the convergence in distribution in (b): Let $\tau = \sqrt{n-2}\left(\theta_{ij} - \frac{\pi}{2}\right)$. Using expression for $h_n(\theta)$, the log density of τ can be obtained as

$$\log_e g(\tau) = C_n + (n-2)\log_e \cos\left(\frac{\tau}{\sqrt{n-2}}\right), \qquad \tau \in \left[-\sqrt{n-2}\,\frac{\pi}{2}, \sqrt{n-2}\,\frac{\pi}{2}\right],$$

where $\exp(C_n)$ is the normalization term depending on n alone. Using Taylor expansion about $\tau = 0$,

$$\begin{aligned}\log_e g(\tau) &= C_n + (n-2)\left[-\frac{\tau^2}{2(n-2)} - \frac{\tau^4}{12(n-2)^2} - \dots\right] \\ &= C_n - \frac{\tau^2}{2} + O\left(\frac{1}{n}\right).\end{aligned}$$

$\Rightarrow g(\tau) \propto e^{-\frac{\tau^2}{2}}$ at the rate of $O\left(\frac{1}{n}\right)$. Thus, $\tau \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$ and hence $\theta_{ij} \xrightarrow{\mathcal{D}} \mathcal{N}\left(\frac{\pi}{2}, \frac{1}{n-2}\right)$ at the rate of $O\left(\frac{1}{n}\right)$. □

Appendix B
Proof of results in Section IV-B

The following lemma is used to design subsets of independent samples for calculation of estimates.

Lemma 3. *Let the constituent clusters be $I_i = \{i_1, i_2, \dots, i_{\omega_i}\}$ and $I_j = \{j_1, j_2, \dots, j_{\omega_j}\}$ and let them contain only indices of points from the same subspace $\mathcal{U}_a$. Then, $W_i = \{\theta_{i_p i_q} \mid p, q = 1, 2, \dots, \omega_i,\ p < q\}$ and $B_{ij} = \{\theta_{i_p j_q} \mid p = 1, 2, \dots, \omega_i,\ q = 1, 2, \dots, \omega_j\}$. Define $W_i^{t_i} = \{\theta_{i_{(2k-1)} i_{(2k)}} \mid k = 1, 2, \dots, \lfloor \frac{\omega_i}{2} \rfloor\}$. Then, $|W_i^{t_i}| = t_i = \lfloor \frac{\omega_i}{2} \rfloor$. Let $\omega = \min(\omega_i, \omega_j)$. Define $B_{ij}^{t_j} = \{\theta_{i_p j_p} \mid p = 1, 2, \dots, \omega\}$. Then, $|B_{ij}^{t_j}| = t_j = \omega$. The following holds on the estimates under Assumption 1:*

a) *$\mu_{w_i t_i}$ and $\mu_{b_{ij} t_j}$ are calculated using independent angle samples.*
b) *$\mu_{w_i t_i}$ and $\mu_{b_{ij} t_j}$ are independent.*
c) *The corresponding variance estimates $\sigma^2_{w_i t_i}$ and $\sigma^2_{b_{ij} t_j}$ are also independent.*

Proof. As seen in the design of the set $W_i^{t_i}$, only one angle is chosen per data point in the set and hence the estimates using this set uses independent samples under Assumption 1. Also, the between angle set $B_{ij}^{t_j}$ contains only one angle per data point which are independent. This proves part a). When we see both the sets together, they contain at most 2 angles formed by a point and under Assumption 1, the angles are pairwise independent if it involves the same point. Hence, $W_i^{t_i} \cup B_{ij}^{t_j}$ contains only independent samples. Hence, the estimates which use disjoint samples from $W_i^{t_i} \cup B_{ij}^{t_j}$ are independent. This proves part b) and c). □

Proof of Lemma 2. Part a): Since we are working with independent samples as designed in Lemma 3 sampled from a Gaussian distribution with mean ν_a and variance ρ_a^2 as per Assumption 1, $\mu_{w_i t_{ij}}, \mu_{b_{ij} t_{ij}} \sim \mathcal{N}(\nu_a, \frac{\rho_a^2}{t_{ij}})$ and $\sigma^2_{w_i t_{ij}},\ \sigma^2_{b_{ij} t_{ij}} \sim \frac{\rho_a^2}{t_{ij}-1}\chi^2_{t_{ij}-1}$ which is a straight forward result on sample mean and variance of a Gaussian random sample as in Theorem 5.3.1 in [76]. Then, X_{ij} is the difference of independent Gaussian random variables with same mean.

$$\Rightarrow \quad X_{ij} \sim \mathcal{N}\left(0, \frac{2\rho_a^2}{t}\right) \quad \text{and} \quad \frac{t}{2\rho_a^2} X_{ij}^2 \sim \chi_1^2.$$

Part b) Similar to Part a), here we have

$$\sigma^2_{w_i t_{ij}} \sim \frac{\rho_a^2}{t_{ij}-1}\chi^2_{t_{ij}-1} \qquad \sigma^2_{b_{ij} t_{ij}} \sim \frac{\rho_{ab}^2}{t_{ij}-1}\chi^2_{t_{ij}-1}$$

$$(\mu_{w_i t_{ij}} - \mu_{b_{ij} t_{ij}}) \sim \mathcal{N}\left(\nu_a - \nu_{ab}, \frac{\rho_a^2 + \rho_{ab}^2}{t_{ij}}\right).$$

From the distribution of difference in means, we can also define the distribution of its square as a scaled non-central χ^2 distribution.

$$\frac{t_{ij}}{\rho_a^2 + \rho_{ab}^2} X_{ij}^2 \sim \chi^2\left(k = 1, \lambda = t_{ij}\frac{(\nu_a - \nu_{ab})^2}{\rho_a^2 + \rho_{ab}^2}\right). \qquad \square$$

Proof of Theorem 2. From (10) in Lemma 2:

$$\begin{aligned}&\frac{t_{ij}}{\rho_a^2 + \rho_{ab}^2} X_{ij}^2 \geq t_{ij}\log_e(1 + M_{ab})\alpha_{t_{ij}} \ \ w.p \\ &\qquad 1 - F_{\chi^2(1, t_{ij} M_{ab}^2)}\left(t_{ij}\log_e(1 + M_{ab})\alpha_{t_{ij}}\right)\end{aligned} \tag{17}$$

Using the distributions for sample variances in (10),

$$\begin{aligned}&2\rho_a^2 - \rho_a^2\alpha_{t_{ij}} \leq \sigma^2_{w_i t_{ij}} \leq \rho_a^2\alpha_{t_{ij}} \qquad w.p \\ &F_{\chi^2_{t_{ij}-1}}\left((t_{ij}-1)\alpha_{t_{ij}}\right) - F_{\chi^2_{t_{ij}-1}}\left((t_{ij}-1)(2-\alpha_{t_{ij}})\right)\end{aligned} \tag{18}$$

and

$$2\rho_{ab}^2 - \rho_{ab}^2\alpha_{t_{ij}} \leq \sigma_{b_{ij}t_{ij}}^2 \leq \rho_{ab}^2\alpha_{t_{ij}} \qquad w.p$$
$$F_{\chi^2_{t_{ij}-1}}\big((t_{ij}-1)\alpha_{t_{ij}}\big) - F_{\chi^2_{t_{ij}-1}}\big((t_{ij}-1)(2-\alpha_{t_{ij}})\big). \quad (19)$$

The independent events (17), (18) and (19) occur together $w.p$ $\Big[F_{\chi^2_{t_{ij}-1}}\big((t_{ij}-1)\alpha_{t_{ij}}\big) - F_{\chi^2_{t_{ij}-1}}\big((t_{ij}-1)(2-\alpha_{t_{ij}})\big)\Big]^2 \times \Big[1 - F_{\chi^2(1,t_{ij}M_{ab}^2)}\big(t_{ij}\log_e(1+M_{ab})\alpha_{t_{ij}}\big)\Big] = 1-\delta_{t_{ij}}^{ab}$. Consider $Y_{ij} \leq (\rho_a^2+\rho_{ab}^2)\alpha_{t_{ij}}$. From (18) and (19), This occurs $w.p \geq 1-\delta_{t_{ij}}^{ab}$. Thus, $U_{ij} = X_{ij}^2/Y_{ij} \geq \dfrac{X_{ij}^2}{(\rho_a^2+\rho_{ab}^2)\alpha_{t_{ij}}}$. Using (17),

$$\frac{1}{4}U_{ij} \geq \frac{\log_e(1+M_{ab})}{4} \qquad w.p \geq 1-\delta_{t_{ij}}^{ab}. \quad (20)$$

Also from (18) and (19),

$$V_{ij} = \frac{\sigma_{w_i t_{ij}}^2}{\sigma_{b_{ij}t_{ij}}^2} + \frac{\sigma_{b_{ij}t_{ij}}^2}{\sigma_{w_i t_{ij}}^2} \geq \frac{\rho_a^2(2-\alpha_{t_{ij}})}{\rho_{ab}^2\alpha_{t_{ij}}} + \frac{\rho_{ab}^2(2-\alpha_{t_{ij}})}{\rho_a^2\alpha_{t_{ij}}} = \frac{2-\alpha_{t_{ij}}}{\alpha_{t_{ij}}}R_{ab}.$$

Hence, $w.p \geq 1-\delta_{t_{ij}}^{ab}$,

$$\frac{1}{4}\log_e\Big[\frac{V_{ij}}{4}+\frac{1}{2}\Big] \geq \frac{1}{4}\log_e\Big[\Big(\frac{2-\alpha_{t_{ij}}}{\alpha_{t_{ij}}}\Big)\frac{R_{ab}}{4}+\frac{1}{2}\Big]. \quad (21)$$

Now let us look at d_{ij}. From (20) and (21),

$$d_{ij} \geq \frac{\log_e(1+M_{ab})}{4} + \frac{1}{4}\log_e\Big[\Big(\frac{2-\alpha_{t_{ij}}}{\alpha_{t_{ij}}}\Big)\frac{R_{ab}}{4}+\frac{1}{2}\Big].$$

To ensure $d_{ij} \geq \frac{1}{\sqrt{t_{ij}-1}}$ $w.p \geq 1-\delta_{t_{ij}}^{ab}$, it is sufficient that:

$$\frac{\log_e(1+M_{ab})}{4} + \frac{1}{4}\log_e\Big[\Big(\frac{2-\alpha_{t_{ij}}}{\alpha_{t_{ij}}}\Big)\frac{R_{ab}}{4}+\frac{1}{2}\Big] \geq \frac{1}{\sqrt{t_{ij}-1}}.$$

$$\Rightarrow \Big(\frac{2-\alpha_{t_{ij}}}{\alpha_{t_{ij}}}\Big)R_{ab}(1+M_{ab}) \geq 4\alpha_{t_{ij}} - 2(1+M_{ab}).$$

$$\Rightarrow 4\alpha_{t_{ij}}^2 + (R_{ab}-2)(1+M_{ab})\alpha_{t_{ij}} - 2R_{ab}(1+M_{ab}) \leq 0. \quad (22)$$

The roots of the above quadratic are obtained as $(\psi_{ab}, \psi'_{ab}) = \frac{-(1+M_{ab})(R_{ab}-2) \pm \sqrt{(1+M_{ab})^2(R_{ab}-2)^2 + 32R_{ab}(1+M_{ab})}}{8}$. If $x > 0$ then $x + x^{-1} \geq 2 \Rightarrow R_{ab} \geq 2$. Also, $M_{ab} \geq 0$. Thus, the roots are real and $\psi'_{ab} \leq \psi_{ab}$. Therefore, from (22), we have $4(\alpha_{t_{ij}} - \psi'_{ab})(\alpha_{t_{ij}} - \psi_{ab}) \leq 0 \Rightarrow \psi'_{ab} \leq \alpha_{t_{ij}} \leq \psi_{ab}$. If we assume $\psi_{ab} < 1$ and use the fact that $M_{ab} \geq 0$, we will arrive at the contradiction $R_{ab} < 2$. Thus, $\psi_{ab} \geq 1$. Also note that $\alpha_{t_{ij}} \geq 1$ and $\psi'_{ab} \leq 0$. Hence, $1 \leq \alpha_{t_{ij}} \leq \psi_{ab}$ where

$$\psi_{ab} = \frac{\sqrt{(R_{ab}-2)^2(1+M_{ab})^2 + 32R_{ab}(1+M_{ab})} - (R_{ab}-2)(1+M_{ab})}{8}.$$

Thus, $d_{ij} \geq \frac{1}{\sqrt{t_{ij}-1}}$, $w.p \geq 1-\delta_{t_{ij}}^{ab}$ if $\alpha_{t_{ij}} \leq \psi_{ab}$

$$\Rightarrow \; t_{ij} \geq 1 + \frac{16}{(\log_e \psi_{ab})^2}. \qquad \square$$

REFERENCES

[1] R. Xu and D. C. Wunsch II, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[2] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *Int. Conf. Database Theory*. Springer, 1999, pp. 217–235.

[3] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[4] V. S. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[5] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[6] I. Jolliffe, *Principal component analysis (Springer Series in Statistics)*. Berlin, Germany: Springer, 2002.

[7] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.

[8] W. Hong, J. Wright, K. Huang, and Y. Ma, "Multiscale hybrid linear models for lossy image representation," *IEEE Trans. Image Processing*, vol. 15, no. 12, pp. 3655–3671, 2006.

[9] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using powerfactorization and gpca," *Int. J. Computer Vision*, vol. 79, no. 1, pp. 85–105, 2008.

[10] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2003, pp. 11–18.

[11] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.

[12] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[13] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *Int. Conf. on Decision and Control*. IEEE, 2003, pp. 167–172.

[14] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: a survey," *IEEE Trans. Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370–1386, 2004.

[15] E. Achtert, C. Böhm, H.-P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek, "Finding hierarchies of subspace clusters," in *European Conf. Principles of Data Mining and Knowledge Discovery*. Springer, 2006, pp. 446–453.

[16] N. Agarwal, E. Haque, H. Liu, and L. Parsons, "Research paper recommender systems: A subspace clustering approach," in *Int. Conf. Web-Age Info. Management*. Springer, 2005, pp. 475–491.

[17] X. Zhou, J. Liang, Y. Hu, and L. Guo, "Text document latent subspace clustering by plsa factors," in *IEEE/WIC/ACM Int. Joint Conf. Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 2. IEEE, 2014, pp. 442–448.

[18] Y. Wu, Z. Zhang, T. S. Huang, and J. Y. Lin, "Multibody grouping via orthogonal subspace decomposition," in *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2001, pp. 252–257.

[19] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Int. Conf. Computer Vision Workshops*. IEEE, 2009, pp. 234–241.

[20] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1546–1562, 2007.

[21] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[22] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[23] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[24] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *J. Machine Learning Research*, vol. 14, no. 1, pp. 2487–2517, 2013.

[25] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 3918–3927.

[26] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," in *European Conf. on Computer Vision*. Springer, 2012, pp. 347–360.

[27] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 3928–3937.

[28] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *IEEE Conf. Computer Vision and Pattern Recognition*. IEEE, 2011, pp. 1801–1807.

[29] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When lrr meets ssc," in *Adv. Neural Info. Processing Systems*, 2013, pp. 64–72.

[30] Y.-X. Wang, H. Xu, and C. Leng, "Provable subspace clustering: When lrr meets ssc," *IEEE Trans. Info. Theory*, vol. 65, no. 9, pp. 5406–5432, 2019.

[31] C. Lu, J. Feng, Z. Lin, T. Mei, and S. Yan, "Subspace clustering by block diagonal representation." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 487–501, 2019.

[32] A. Leonardis, H. Bischof, and J. Maver, "Multiple eigenspaces," *Pattern recognition*, vol. 35, no. 11, pp. 2613–2627, 2002.

[33] Z. Fan, J. Zhou, and Y. Wu, "Multibody grouping by inference of multiple subspaces from high-dimensional data using oriented-frames," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 91–105, 2005.

[34] M. Rahmani and G. K. Atia, "Innovation pursuit: A new approach to subspace clustering," *IEEE Trans. Signal Processing*, vol. 65, no. 23, pp. 6276–6291, 2017.

[35] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39 501–39 514, 2018.

[36] C. Song, F. Liu, Y. Huang, L. Wang, and T. Tan, "Auto-encoder based data clustering," in *Iberoamerican Congress on Pattern Recognition*. Springer, 2013, pp. 117–124.

[37] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Adv. Neural Info. Processing Systems*, 2017, pp. 24–33.

[38] Y. Chen, L. Zhang, and Z. Yi, "Subspace clustering using a low-rank constrained autoencoder," *Info. Sciences*, vol. 424, pp. 27–38, 2018.

[39] T. Cai, J. Fan, and T. Jiang, "Distributions of angles in random packing on spheres," *J. Machine Learning Research*, vol. 14, no. 1, pp. 1837–1864, 2013.

[40] V. Menon and S. Kalyani, "Structured and unstructured outlier identification for robust pca: A fast parameter free algorithm," *IEEE Trans. Signal Processing*, vol. 67, no. 9, pp. 2439–2452, 2019.

[41] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Trans. Info. Theory*, vol. 61, no. 11, pp. 6320–6342, 2015.

[42] M. Rahmani and G. K. Atia, "Coherence pursuit: Fast, simple, and robust principal component analysis," *IEEE Trans. Signal Processing*, vol. 65, no. 23, pp. 6260–6275, 2017.

[43] A. Gitlin, B. Tao, L. Balzano, and J. Lipor, "Improving k-subspaces via coherence pursuit," *IEEE J. Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1575–1588, 2018.

[44] J. Lipor, D. Hong, Y. S. Tan, and L. Balzano, "Subspace clustering using ensembles of k-subspaces," *arXiv preprint arXiv:1709.04744*, 2017.

[45] J. Dopazo, E. Zanders, I. Dragoni, G. Amphlett, and F. Falciani, "Methods and approaches in the analysis of gene expression data," *J. immunological methods*, vol. 250, no. 1-2, pp. 93–112, 2001.

[46] M. Comiter, M. Cha, H. Kung, and S. Teerapittayanon, "Lambda means clustering: automatic parameter search and distributed computing implementation," in *Int. Conf. Pattern Recognition*. IEEE, 2016, pp. 2331–2337.

[47] B. Kulis and M. I. Jordan, "Revisiting k-means: New algorithms via bayesian nonparametrics," in *Int. Conf. Machine Learning*. Omnipress, 2012, pp. 1131–1138.

[48] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.

[49] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Int. Conf. Machine Learning*. Morgan Kaufmann Publishers Inc., 2000, pp. 727–734.

[50] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. the Royal Statistical Society: Series B*, vol. 63, no. 2, pp. 411–423, 2001.

[51] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *Int. Conf. Tools with Artificial Intelligence*. IEEE, 2004, pp. 576–584.

[52] A. Gupta, S. Datta, and S. Das, "Fast automatic estimation of the number of clusters from the minimum inter-center distance for k-means clustering," *Pattern Recognition Letters*, vol. 116, pp. 72–79, 2018.

[53] S. Kallummil and S. Kalyani, "Signal and noise statistics oblivious orthogonal matching pursuit," in *Int. Conf. Machine Learning*, 2018, pp. 2434–2443.

[54] S. Kallummil and S. Kalyani, "High snr consistent compressive sensing without signal and noise statistics," *arXiv preprint arXiv:1811.07131*, 2018.

[55] S. Kallummil and S. Kalyani, "Noise statistics oblivious gard for robust regression with sparse outliers," *IEEE Trans. Signal Processing*, vol. 67, no. 2, pp. 383–398, 2019.

[56] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation." in *Int. Conf. Machine learning*, vol. 1, 2010, p. 8.

[57] J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network *et al.*, "The cancer genome atlas pan-cancer analysis project," *Nature genetics*, vol. 45, no. 10, p. 1113, 2013.

[58] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[59] "Cancer program datasets." [Online]. Available: http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi

[60] J. G. Rohra, B. Perumal, S. J. Narayanan, P. Thakur, and R. B. Bhatt, "User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks," in *Int. Conf. Soft Computing for Problem Solving*. Springer, 2017, pp. 286–295.

[61] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *The Annals of Statistics*, pp. 73–102, 1995.

[62] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[63] 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 Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[64] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[65] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," in *IEEE Conf. Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8.

[66] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[67] A. K. Jain, "On an estimate of the bhattacharyya distance," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 6, no. 11, pp. 763–766, 1976.

[68] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *Adv. Neural Info. Processing Systems*, 2014, pp. 2753–2761.

[69] T. T. Cai and T. Jiang, "Phase transition in limiting distributions of coherence of high-dimensional random matrices," *J. Multivariate Analysis*, vol. 107, pp. 24–39, 2012.

[70] 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.

[71] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*. John Wiley and Sons, New York, 1995, vol. 2.

[72] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *J. Machine Learning Research*, vol. 3, no. Dec, pp. 583–617, 2002.

[73] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[74] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans. Image Processing*, vol. 27, no. 10, pp. 5076–5086, 2018.

[75] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Int. J. Computer Vision*, vol. 100, no. 3, pp. 217–240, 2012.

[76] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002, vol. 2.