HHS Public Access

Author manuscript

IEEE Int Conf Smart Cloud. Author manuscript; available in PMC 2024 May 05.

Published in final edited form as:

IEEE Int Conf Smart Cloud. 2023 September ; 2023: 164–169. doi:10.1109/smartcloud58862.2023.00036.

Feature Selection for Unsupervised Machine Learning

Huyunting Huang[1], **Ziyang Tang**[1], **Tonglin Zhang**[1], **Baijian Yang**[1], **Qianqian Song**[2], **Jing Su**[3]

[1]*Purdue University* West Lafayette, Indiana

[2]Wake Forest School of Medicine

[3]Indiana University School of Medicine

Abstract

Compared to supervised machine learning (ML), the development of feature selection for unsupervised ML is far behind. To address this issue, the current research proposes a stepwise feature selection approach for clustering methods with a specification to the Gaussian mixture model (GMM) and the *k*-means. Rather than the existing GMM and *k*-means which are carried out based on all the features, the proposed method selects a subset of features to implement the two methods, respectively. The research finds that a better result can be obtained if the existing GMM and *k*-means methods are modified by nice initializations. Experiments based on Monte Carlo simulations show that the proposed method is more computationally efficient and the result is more accurate than the existing GMM and *k*-means methods based on all the features. The experiment based on a real-world dataset confirms this finding.

Keywords

adjusted rand index; Gaussian mixture model; *k*-means; stepwise

I. Introduction

Feature selection, also known as variable selection, is a popular machine learning (ML) approach for high-dimensional data. The goal is to select a few features (i.e., explanatory variables) from many candidates, such that the result can be better interpreted and understood. Feature selection is particularly important in the case when the number of features (i.e., p) is larger than the number of observations (i.e., n), known as the large p and small n problem. Currently, feature selection is mostly applied to supervised ML problems, where it assumes that there is a response variable to be interpreted by the explanatory variables. Although unsupervised ML problems are also important in practice, the corresponding feature selection method has not been well-understood. This motivates the goal of the current research.

Rather than supervised ML, unsupervised ML assumes that there is no response in the data. A well-known problem is clustering. Basically, clustering treats all variables as features. It

huan1182@purdue.edu .

assumes that there is no response in the data. The goal is to partition the data into many clusters (i.e., subsets), such that observations within clusters are the most homogeneous and observations between clusters are the most heterogeneous. Many clustering methods have been proposed. Examples include the k-means [1], the k-medians [2], the k-modes [3], the generalized k-means [4], and the Gaussian mixture model (GMM) [5]. Among those, the k-means and the GMM are considered the most straightforward and popular. In the literature, clustering is carried out based on all the features. An obvious drawback is that the resulting model may be too complicated if the number of features is large. To address this issue, an convenient way is to apply a feature selection method to select a subset of features. Here we propose a stepwise feature selection approach for clustering methods with specifications to the GMM and the k-means, which has obvious advantages over previous methods.

Although our idea can be implemented in any clustering method, we focus our presentation on the k-means and the GMM. We assume that data with n observations and p features can be generally expressed as $\mathscr{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ with $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\top \in \mathbb{R}^p$ representing the ith observation for $i \in \mathscr{R} = \{1, \ldots, n\}$, where $\mathscr{R}$ represents the set of observations (i.e., records). The goal of clustering is to partition $\mathscr{R}$ into many clusters (e.g., k clusters) denoted as $\mathscr{C} = \{C_1, \ldots, C_k\}$, which satisfies $C_r \cap C_s = \varnothing$ for any $r \neq s$ and $\bigcup_{r=1}^{k} C_r = \mathscr{R}$. To carry out the clustering method, it is necessary to provide the distance between C_r and C_s. The distance is often defined by dissimilarity between points with the form of $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ where $d(\cdot, \cdot)$ is a certain distance function between points. To carry out feature selection, we treat $d(\boldsymbol{x}_{iA}, \boldsymbol{x}_{jA})$ as the distance between the ith and jth observations, where $\boldsymbol{x}_{iA}$ and $\boldsymbol{x}_{jA}$ are sub-vectors of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ with their subscripts belonging to some $A \subseteq \mathscr{F} = \{1, \ldots, p\}$, respectively. If the partition provided by clustering with feature selection is close to that without, then the number of features is reduced from p to $|A|$; otherwise, another option of $A \subseteq \mathscr{F}$ is investigated. If p is large but $|A|$ is small, then the result of clustering with feature selection is much easier to understand and better to interpret than that without.

In our experiments, we evaluate our method via Monte Carlo simulations and real-world data. In our simulation study, we find that the number of features can be significantly reduced in both the k-means and GMM by our method. For real-world data, we apply our methods to single-cell spatial transcriptomics (SCST) multi-modal dataset [6]. The dataset has $n = 79{,}876$ observations and $p = 981$ variables. Because p is large, we use our method to select important features for clustering. We find that our method works well when 30 features are adopted. We successfully reduce the number of features from 981 to 30.

The contributions of this article are:

- We point out that feature selection is needed in unsupervised ML. This problem has not been well understood yet.
- We define the feature selection problem for a general clustering method with specifications to the k-means and GMM.

- We implement our feature selection method to a real data example with many variables. We successfully reduce the number of features to a low level, indicating that our method works well. We find that this cannot be achieved by previous methods.

The remainder of the paper is structured as follows. In Section II, we review the relevant background. In Section III, we propose our method. In Section IV, we evaluate our method by experiments, including both Monte Carlo simulation and real-world application. In Section V, we conclude the article.

II. Background

In the literature, feature selection is usually carried out by the PML approach for a supervised ML problem. An example is the high-dimensional linear model with a large number of features. The purpose is to set the estimates of the regression coefficients for all of the unimportant features to be zero. To achieve this goal, feature selection uses a Lagrangian form objective function with penalty functions added [7].

Two typical clustering methods in unsupervised ML are the GMM and the k-means. The GMM assumes that the data are collected from a mixture model with k components with the distribution of the rth component given by the PDF of $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denoted as $\varphi(\boldsymbol{x}_i; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r), r = 1, \ldots, k$. Let $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ik})^\top$ be the ground truth of the ith observation with $z_{ir} \in \{0, 1\}$ and $\sum_{r=1}^{k} z_{ir} = 1$. Then, $\boldsymbol{z}_i$ is the cluster assignment of $\boldsymbol{x}_i$, meaning that $z_{ir} = 1$ iff $\boldsymbol{x}_i$ belongs to the r cluster. The mixture model can be expressed by a complete data version and an incomplete data version. The complete data version assumes that both $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ are available, leading to the complete data set as $\mathcal{D}_c = \{(\boldsymbol{x}_i; \boldsymbol{z}_i) : i = 1, \ldots, n\}$ with the underlying distribution as

$$\boldsymbol{x}_i \mid \boldsymbol{z}_i \sim^{iid} \sum_{r=1}^{k} z_{ir} \varphi_r(\boldsymbol{x}_i; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r). \tag{1}$$

The incomplete data version treats $\boldsymbol{z}_i$ as unobserved latent variables, leading to the observed data set as $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. Assume that $\boldsymbol{z}_i$ are iid from a Dirichlet distribution with probability vector $\pi = (\pi_1, \ldots, \pi_k)^\top$. By integrating $\boldsymbol{z}_i$ out from (1), the distribution of $\mathcal{D}$ is obtained as

$$\boldsymbol{x}_i \sim^{iid} \sum_{r=1}^{k} \pi_r \varphi_r(\boldsymbol{x}_i; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r). \tag{2}$$

A usable clustering method can only be developed under (2), implying that (1) can only be used for theoretical evaluations.

If $\mathbf{\Sigma}_r = \sigma^2\mathbf{I}$ for $r = 1, \ldots, k$, then (2) is the k-means (i.e., spherical) model. If $\mathbf{\Sigma}_r$ for all $r = 1, \ldots, k$ are distinct, then (2) is the quadratic discriminant analysis (QDA) model. The linear discriminant analysis (LDA) model is derived if we assume that all $\mathbf{\Sigma}_r$ are identical. To be consistent with the k-means problem, it is usually assumed that $\boldsymbol{\mu}_r$ are all distinct, leading to the GMM with distinct mean vectors.

The GMM clustering is carried out by an EM algorithm. At the current iteration (i.e., the tth iteration), the EM-algorithm updates current iterated values $\pi_r^{(t)}$, $\boldsymbol{\mu}_r^{(t)}$, and $\mathbf{\Sigma}_r^{(t)}$ of π_r, $\boldsymbol{\mu}_r$, and $\mathbf{\Sigma}_r$ based on the previous $\pi_r^{(t-1)}$, $\boldsymbol{\mu}_r^{(t-1)}$, and $\mathbf{\Sigma}_r^{(t-1)}$. In the end, the EM algorithm estimates the partition by

$$\hat{C}_r = \{i : \hat{z}_{ir} = \underset{j \in \{1, \ldots, k\}}{\operatorname{argmax}} \hat{z}_{ij}\},\ r = 1, \ldots, k, \tag{3}$$

where $\hat{\boldsymbol{z}}_i = (\hat{z}_{i1}, \ldots, \hat{z}_{ik})^\top$ is the ith final imputed $\boldsymbol{z}_i$.

The k-means directly computes the current iterative value $\boldsymbol{z}_i^{(t)}$ of $\boldsymbol{z}_i$ given the previous centroids $\boldsymbol{\mu}_1^{(t-1)}, \ldots, \boldsymbol{\mu}_k^{(t-1)}$. It then update the current centroids and obtains $\boldsymbol{\mu}_1^{(t)}, \ldots, \boldsymbol{\mu}_k^{(t)}$. In the end, it estimates the partition by $\hat{C}_r = \{i : \hat{z}_{ir} = 1\}$ and the parameters by $\hat{\boldsymbol{\mu}}_r = (1/|\hat{C}_r|)\sum_{i \in \hat{C}_r} \boldsymbol{x}_i,\ r = 1, \ldots, k$, where $\hat{\boldsymbol{z}}_i = (\hat{z}_{i1}, \ldots, \hat{z}_{ik})^\top$ is the final imputed vector of the ground truth. Neither the EM algorithm nor the k-means method uses the ground truth z_i in the derivation of $\hat{C}_r$. Instead, they use the imputed $\hat{z}_i$. Therefore, they are usable.

Although a few variable selection methods for clustering have been proposed, computational prohibition has been identified in the case when the number of variables is moderate due to exponential growth of the computational burden with the number of variables [8]. This issue has been overcome by several methods, such as the sparse k-means [9], and the model-based variable selection [10]–[13]. These methods can be implemented by the `sparcl`, `clustvarsel`, `VarSelLCM` packages of `R`. However, a recent study points out that most of those have not been evaluated by a comprehensive experimental study and there is a lack of theoretical evaluations about how variable selection affects the performance of clustering [14]. This concern is addressed by our work.

III. Methodology

Feature selection for unsupervised learning is fundamentally different from that for supervised ML. The goal is to select a subset of features such that the result of clustering based on the subset can be as accurate as or even more accurate than that based on the entire set. In particular, let $A = \{j_1, \ldots, j_a\} \subseteq \mathcal{F}$ be a candidate subset of features, where $a = |A|$ is the cardinality of A. As both a and A are unknown, there may be as large as 2^p subsets to be considered if the brute force approach is adopted. This is impossible if p is only moderate (e.g., $p = 20$). Thus, we discard the brute force method and propose a stepwise approach to

determine the best A. We find that the complexity of our method is $o(p^2n)$, indicating that it can be easily implemented even if p is extremely large. We introduce our method below.

We present our method for the case when A is given first and then move our interest to the case when A is selected by the stepwise approach. For a given A, we have two ways to implement a clustering method. In the first, we only use the features contained by A. We treat $\boldsymbol{x}_{iA} = (x_{ij_1}, \ldots, x_{ij_a})^\top$ as the feature vector of the ith observation. We obtain a partition of $\mathscr{R}$ denoted as $\mathscr{C}_A = \{C_{1A}, \ldots, C_{kA}\}$. In the second, we use all of the features. We treat $\boldsymbol{x}_i$ as the feature vector of the ith observation. We obtain a partition of $\mathscr{R}$ denoted as $\mathscr{C} = \{C_1, \ldots, C_k\}$. Because the first only uses a features but the second uses all of the p features, we expect that $\mathscr{C}_A$ and $\mathscr{C}$ are different. We need to study the difference between $\mathscr{C}_A$ and $\mathscr{C}$.

We use the likelihood approach to measure the difference between $\mathscr{C}_A$ and $\mathscr{C}$. We specify the approach to the GMM and the k-means methods, respectively. Because $\boldsymbol{z}_i$ has been imputed, we can use (1) to compute the imputed complete data loglikelihood under $\mathscr{C}$ as

$$\ell(\mathscr{C}) = \sum_{r=1}^{k} \sum_{i \in C_r} \log[\varphi(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_r, \hat{\boldsymbol{\Sigma}}_r)], \tag{4}$$

where $\hat{\boldsymbol{\mu}}_r = \sum_{i \in C_r} \boldsymbol{x}_i / |C_r|$ and $\hat{\boldsymbol{\Sigma}}_r = \sum_{i \in C_r} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_r)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_r)^\top / (|C_r| - 1)$ are the estimates of $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}_r$, respectively. Similarly, we can compute the imputed complete data loglikelihood based on $\mathscr{C}_A$ by $\ell(\mathscr{C}_A)$ after $\hat{\boldsymbol{\mu}}_r$ and $\hat{\boldsymbol{\Sigma}}_r$ are replaced with $\hat{\boldsymbol{\mu}}_{rA} = \sum_{i \in C_{rA}} \boldsymbol{x}_i / |C_{rA}|$ and $\hat{\boldsymbol{\Sigma}}_{rA} = \sum_{i \in C_{rA}} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{rA})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{rA})^\top / (|C_{rA}| - 1)$ under $\mathscr{C}_A$ in (4), respectively. We use

$$d(\mathscr{C}, \mathscr{C}_A) = \ell(\mathscr{C}) - \ell(\mathscr{C}_A) \tag{5}$$

to measure the difference between $\mathscr{C}$ and $\mathscr{C}_A$ in the GMM method. In the k-means method, we use $\boldsymbol{\Sigma}_r = \sigma^2 \mathbf{I}$ for all $r = 1, \ldots, k$ to compute the modified imputed complete data loglikelihood $\tilde{\ell}(\mathscr{C})$ under $\mathscr{C}$. Similarly, we compute the modified imputed complete data log likelihood $\tilde{\ell}(\mathscr{C}_A)$ under $\mathscr{C}_A$. We use

$$\tilde{d}(\mathscr{C}, \mathscr{C}_A) = \tilde{\ell}(\mathscr{C}) - \tilde{\ell}(\mathscr{C}_A) \tag{6}$$

to measure the difference between $\mathscr{C}$ and $\mathscr{C}_A$ in the k-means method. We treat the difference given by (5) in the GMM method or (6) in the k-means method as the loss of A. It is denoted as $\text{loss}(A)$.

As (5) and (6) can only be applied based on a given A, we devise our method for the selection of the best A. In particular, we compute $\text{loss}(A)$ under a number of $A \subseteq \mathscr{F}$ with the

best A determined by the minimum loss value. To reduce the number of candidate subsets, we propose a stepwise approach to search for the best A. It reduces the number of candidate subsets from 2^p to p^2, implying that our method can be implemented even if p is large.

The stepwise approach starts with the empty set and adds one of the most important features to A once a time at each step of the iteration. The process continues until no more important features are identified. In the first step, we search for the most important feature in the entire $\mathscr{F}$. To achieve this, we compute loss(A) with $A = \{j\}$ for all $j \in \mathscr{F}$. The most important j is determined by

$$j_{\min}^{(1)} = \underset{j \in \mathscr{F}}{\operatorname{argmin}}\, loss(\{j\}). \tag{7}$$

The first step provides $A = \{j_{\min}^{(1)}\}$ with $a = 1$. In the tth step for any $t > 1$, let $A^{(t-1)} = \{j_1, j_2, \ldots, j_{t-1}\}$ be the set of important features selected by the previous $t - 1$ steps. In the tth step, we search the most important feature in $\mathscr{F}$ but not in $A^{(t-1)}$. To achieve this, we compute $loss(A)$ with $A = A^{(t-1)} \cup \left\{j\right\}$ for all $j \in \mathscr{F} \setminus A^{(t-1)}$. The most important j is determined by

$$j_{\min}^{(t)} = \underset{j \in \mathscr{F} \setminus A^{(t-1)}}{\operatorname{argmin}}\, loss(A^{(t+1)} \cup \{j\}). \tag{8}$$

The tth step updates the set of important features by $A^{(t)} = A^{(t-1)} \cup \{j_{\min}^{(t)}\}$ with $a = t$. We keep doing this until we cannot find any important features. To determine this, we can use the well-known BIC approach. In this research, we find that the BIC approach is not necessary to be used. This is fundamentally different from variable selection for a supervised learning problem, where BIC or a modification of BIC is considered as necessary. Then, we propose Algorithm 1.

Algorithm 1

Feature selection for the GMM or the k-means

Input: Data set $\mathscr{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ and the number of clusters k

Output: labels $1, \ldots, k$ for each $\boldsymbol{x}_i \in \mathscr{D}$ with the best $A \subseteq \mathscr{F}$

Initialization

1: Determine the first $j_{\min}^{(1)}$ by (7) with $loss(A) = d(\mathscr{C}, \mathscr{C}_A)$ given by (5) if the GMM method is adopted or $loss(A) = \tilde{d}(\mathscr{C}, \mathscr{C}_A)$ given by (6) if the k-means method is adopted

Begin Iteration

2: Let $A^{(t-1)} = \{j_{\min}^{(1)}, j_{\min}^{(2)}, \ldots, j_{\min}^{(t-1)}\}$ be the previous set of important features and determine the current $j_{\min}^{(t)}$ by (8) and update $A^{(t)} = A^{(t-1)} \cup \{j_{\min}^{(t)}\}$

3: Stop if $t = p$ or no important feature is found; otherwise continue

End Iteration

4: Output

An important issue is to specify k (i.e., the number of clusters) in Algorithm 1. This can be easily solved. There are two scenarios. If k is given, then we can simply use the value of k; otherwise, k should be determined by the GMM or the k-means with an unknown k. The determination of the number of clusters is considered a challenging problem in the implementation of a clustering method. This issue has been previously investigated in the literature. The idea is to implement a given clustering method to a set of candidates of k with the best k to be selected by a predefined criterion. A few criteria have been proposed in the literature. Examples include the minimum message length (MML) criterion [15], the minimum description length (MDL) criterion [16], the Bayesian information criterion (BIC) [4], the silhouette score [17], and the Gap Statistics [18]. We evaluate this issue and find that the determination of k is not a concern. The reason is that we can use the same k determined by the case when all features are used. We assume that k does not vary with A in Algorithm 1. Therefore, k can be assumed to be known in variable selection for a clustering method.

IV. Experiments

We investigate the properties of our method via Monte Carlo simulation and a real-world data example. In both, we use the adjusted rand index (ARI) for the evaluation of the performance. ARI is one of the well-known measures for the accuracy of a clustering method. It is defined as the number of true positives and negatives divided by the total number of pairs. A true positive is a pair of observations claimed in the same cluster by a clustering method and also claimed by the truth. A true negative is a pair of observations claimed in the different clusters by a clustering method and also claimed by the truth. The ARI value is between −1 and 1, with a low value indicating that the result provided by a clustering method does not agree with the truth and 1 indicating that the result is identical to the truth. As the computation of ARI needs the ground truth, it is only used after feature selection is obtained. We determine the best $A \subseteq \mathscr{R}$ by $loss(A)$, which does not need the ground truth. Therefore, ARI is only used for the performance of feature selection.

A. Simulation

We consider two cases in our simulation. In the first case, we simulate data from $\mathbb{R}^{30}$ with $k = 10$ clusters. We represent the feature set as $\mathscr{F} = \{1, \ldots, 30\}$ and the cluster centers as $\boldsymbol{\mu}_r = (\mu_{r1}, \ldots, \mu_{r30})^\top$ for $r = 1, \ldots, 10$. We assume that the first 3 features are extremely important, the next 3 are weakly important, and the remaining 24 are unimportant. We use a three-step procedure to generate the clusters. In the first step, we generate centres by $\mu_{r1}, \mu_{r2}, \mu_{r3} \sim^{\text{iid}} \mathcal{N}(0, 1)$, $\mu_{r4}, \mu_{r5}, \mu_{r6} \sim^{iid} \mathcal{N}(0, \phi^2)$, and $\mu_{r7} = \cdots = \mu_{r30} = 0$. In the second step, we generated the cluster sizes $n_r \sim^{iid} \mathcal{P}(25)$. In the third step, we generate the observations within the clusters. For each cluster, we independently generate n_r observations from $\mathcal{N}(\boldsymbol{\mu}_r, 0.1^2\mathbf{I})$. Thus, the total number of observations within the rth cluster is n_r. The total

number of observations in the entire data set is $n = \sum_{r=1}^{k} n_r$. The distance between the clusters is primarily controlled by the first three features with the adjustment by the second three features based on ϕ with $\phi = 0.0, 0.1, 0.2, 0.3$, respectively.

We investigate four clustering methods. All of them assumes that $\mathbf{\Sigma}_r = \sigma^2\mathbf{I}$ for all $r = 1, \ldots, 10$. The GMM partitions the data by $loss(A)$ given by (5) in Algorithm 1. The iterations of the basic k-means and the k-means++ methods are the same. They partition the data with $loss(A)$ given by (6) in Algorithm 1. The difference is that the basic k-means randomly chooses its initialization, but the k-means++ uses a probability distribution to determine its initialization. As the performance of both the k-means and the k-means++ is bad, we also consider another version of the k-means method proposed by [19]. As it improves the initialization of the k-means by the max-min principal, we denote this method as k-meansMM.

We simulate 100 datasets for each selected ϕ value. For each generated data set, we use Algorithm 1 to select features. After the best A is determined, we compare their performance by examining their ARI values. We calculate the average ARI values based on the 100 replications (Table I). We find that the k-meansMM method is the best and the GMM is the worst. To understand this issue, we study the ARI curves obtained from each of the simulated datasets (e.g., Figure 1). We find that the curves of the GMM, the basic k-means, and the k-means++ are unstable, leading to their low ARI values. In the k-meansMM, it is enough to use the three most important features, implying that 90% of the features can be ignored. Overall, the K-meansMM performs the best. It is significantly better than the GMM, the basic k-means, and the k-means++ in feature selection, implying that initialization is a critical issue in the clustering methods.

In the second case, we simulated data from $\mathbb{R}^{1000}$ with $k = 2$ clusters (i.e., $p = 1000$). We choose cluster centers as $\boldsymbol{\mu}_r = (\mu_{r1}, \cdots, \mu_{rp})^\top$ with $\mu_{11} = \mu_{12} = 0.16$, $\mu_{21} = \mu_{22} = -0.16$, and $\mu_{rj} = 0$ if $j \geq 3, r = 1, 2$. For each cluster, we independently generated $n_r = 10$ observations from $\mathcal{N}(\boldsymbol{\mu}_r, 0.1^2\mathbf{I})$. We then implement the basic k-means, the k-means++, the k-meansMM, and the GMM to the first q features. We calculate the average ARI values based on 1000 replications (Figure 2). We find that the ARI values decrease with q. Note that only the first 2 features are useful. The simulation indicates that the performance of clustering becomes bad if non-informative features are used. If non-informative variables are removed by a variable selection method, then the accuracy of clustering becomes better for all of the methods that we have studied. Therefore, we conclude that variable selection can improve the performance of clustering.

B. Application

We apply our method to the single-cell spatial transcriptomics (SCST) multi-modal data set [6]. The SCST data set collects the gene expression based on the SCST images for lung cancer from the NanoString *CosMx*™ SMI platform (Figure 3). The date set mainly contains six kinds of cells, including 37281 tumor, 13368 fibroblast, 11664 lymphocyte, 7560 Mcell, 5731 neutrophil, and 4272 endothelial cells. Regarding the NanoString *Lung-9-1* dataset,

the composite images of the DAPI, PanCK, CD45, and CD3 channels from 20 fields of views (FOVs), the cell center coordinates (from the cell metadata file), the single-cell gene expression file of 960 genes are used. For each cell, four images of 120-by-120 pixels with the cell at the center are cropped from the images. The spatial adjacent graph is constructed based on the cell-to-cell distance (Euclidian distance) ≤ 80 pixels. NanoString's annotations of cell types are obtained from their provided Giotto object. A feature extractor was applied to project the gene expression into the high-dimensional latent space, which provided 21 additional variables [20].

We apply Algorithm 1 with $k = 6$ to three clustering methods. The first is the GMM-LDA, which assumes that $\mathbf{\Sigma}_r$ are all identical. The second is the GMM-Sphere, which assumes $\mathbf{\Sigma}_r = \sigma^2\mathbf{I}$ for all $r = 1, 2, 3, 4, 5, 6$. The third is the k-means method. We consider two initialization frameworks. The first uses a random initialization. The second searches for a nice initialization by investigating hundreds of initializations with the best one reported by that with the minimum loss value. We carry out feature selection to the three clustering methods with two initialization frameworks, implying that we have six methods. For each of those, we use $loss(A)$ to select the best A with a number of candidates of $a = |A|$. After A is derived, we evaluate their performance by examining their ARI values (Table II). We find that the best a is about 27. To confirm this, we study the curves of $1 - R^2 = SSE/SST$, where SSE is the sum of squares of errors and SST is the sum of squares of the total. The best options should have the lowest $1 - R^2$ values. In the end, we conclude that the GMM-LDA with a nice initialization is the best method for the implementation of our method.

We check the GMM-LDA, the GMM-Sphere, and the k-means when all the 981 features are used. Our result shows that the ARI values of the GMM-LDA and the GMM-Spheres with a random initialization are 0.554 and 0.280, respectively. The ARI value of the k-means with a random initialization is 0.375. If the nice initialization approach is considered, then the ARI values of the GMM-Sphere and the k-means are 0.368 and 0.463, respectively. We are not able to derive that for the GMM-LDA, because each computation takes more than 5 fours, implying the derivation needs over a thousand hours.

In the end, we compare our method with a few previous methods. These include the sparse clustering (by `sparcl` package of `R`) [9], the model-based clustering (by `clustvarsel` package of `R`) [10], and another model-based clustering (by `VarSelLCM` package of `R`) [11]. Our experiment shows that the `sparcl` was out-of-memory with an error message saying that it could not allocate a vector of size 47.5GB, the `clustvarsel` did not provide anything within two days, and the `VarSelLCM` selected all 980 features by 1.72 days with ARI 0.216. As the computational time was less than 15 minutes, we conclude that our method is more computationally efficient and more accurate than our competitors.

V. Conclusion and Future Work

We treat our method as the first variable selection method for unsupervised machine learning problems because this problem has never been studied previously. We expect that our idea can be applied to arbitrary clustering methods, although we focus on the GMM and

k-means. To carry out variable selection, it is important to investigate the initialization issue in existing clustering methods. We have proposed an approach to the *k*-means and GMM methods. For other clustering methods beyond the *k*-means and the GMM, this should also be investigated. This is left to future research.

References

[1]. MacQueen J, "Classification and analysis of multivariate observations," in 5th Berkeley Symp. Math. Statist. Probability. University of California Los Angeles LA USA, 1967, pp. 281–297.

[2]. Cardot H, Cénac P, and Monnez J-M, "A fast and recursive algorithm for clustering large datasets with k-medians," Computational Statistics & Data Analysis, vol. 56, no. 6, pp. 1434–1449, 2012.

[3]. Chaturvedi A, Green PE, and Caroll JD, "K-modes clustering," Journal of classification, vol. 18, pp. 35–55, 2001.

[4]. Zhang T and Lin G, "Generalized k-means in glms with applications to the outbreak of covid-19 in the united states," Computational Statistics & Data Analysis, vol. 159, p. 107217, 2021. [PubMed: 33723467]

[5]. Löffler M, Zhang AY, and Zhou HH, "Optimality of spectral clustering in the gaussian mixture model," The Annals of Statistics, vol. 49, no. 5, pp. 2506–2530, 2021.

[6]. He S, Bhatt R, Birditt B, Brown C, Brown E, Chantranuvatana K, Danaher P, Dunaway D, Filanoski B, Garrison RG et al. , "Highplex multiomic analysis in ffpe tissue at single-cellular and subcellular resolution by spatial molecular imaging," bioRxiv, pp. 2021–11, 2021.

[7]. Tibshirani R, "Regression shrinkage and selection via the lasso," Journal of the Royal Statistical Society: Series B (Methodological), vol. 58, no. 1, pp. 267–288, 1996.

[8]. Steinley D and Brusco MJ, "Selection of variables in cluster analysis: An empirical comparison of eight procedures," Psychometrika, vol. 73, pp. 125–144, 2008.

[9]. Witten DM and Tibshirani R, "A framework for feature selection in clustering," Journal of the American Statistical Association, vol. 105, no. 490, pp. 713–726, 2010 [PubMed: 20811510]

[10]. Raftery AE and Dean N, "Variable selection for model-based clustering," Journal of the American Statistical Association, vol. 101, no. 473, pp. 168–178, 2006.

[11]. Marbac M and Sedki M, "Variable selection for model-based clustering using the integrated complete-data likelihood," Statistics and Computing, vol. 27, pp. 1049–1063, 2017.

[12]. Qiu H, Zheng Q, Memmi G, Lu J, Qiu M, and Thuraisingham B, "Deep residual learning-based enhanced jpeg compression in the internet of things," IEEE Transactions on Industrial Informatics, vol. 17, no. 3, pp. 2124–2133, 2020.

[13]. Ling C, Jiang J, Wang J, Thai MT, Xue R, Song J, Qiu M, and Zhao L, "Deep graph representation learning and optimization for influence maximization," in International Conference on Machine Learning. PMLR, 2023, pp. 21 350–21 361

[14]. Hancer E, "A new multi-objective differential evolution approach for simultaneous clustering and feature selection," Engineering applications of artificial intelligence, vol. 87, p. 103307, 2020.

[15]. Figueiredo MAT and Jain AK, "Unsupervised learning of finite mixture models," IEEE Transactions on pattern analysis and machine intelligence, vol. 24, no. 3, pp. 381–396, 2002.

[16]. Hansen MH and Yu B, "Model selection and the principle of minimum description length," Journal of the American Statistical Association, vol. 96, no. 454, pp. 746–774, 2001.

[17]. Rousseeuw PJ, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," Journal of computational and applied mathematics, vol. 20, pp. 53–65, 1987.

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

[19]. Zhang T, "Asymptotics for the *k*-means," arXiv preprint arXiv:2211.10015, 2022.

[20]. Tang Z, Zhang T, Yang B, Su J, and Song Q, "Sigra: Single-cell spatial elucidation through image-augmented graph transformer," bioRxiv, pp. 2022–08, 2022.

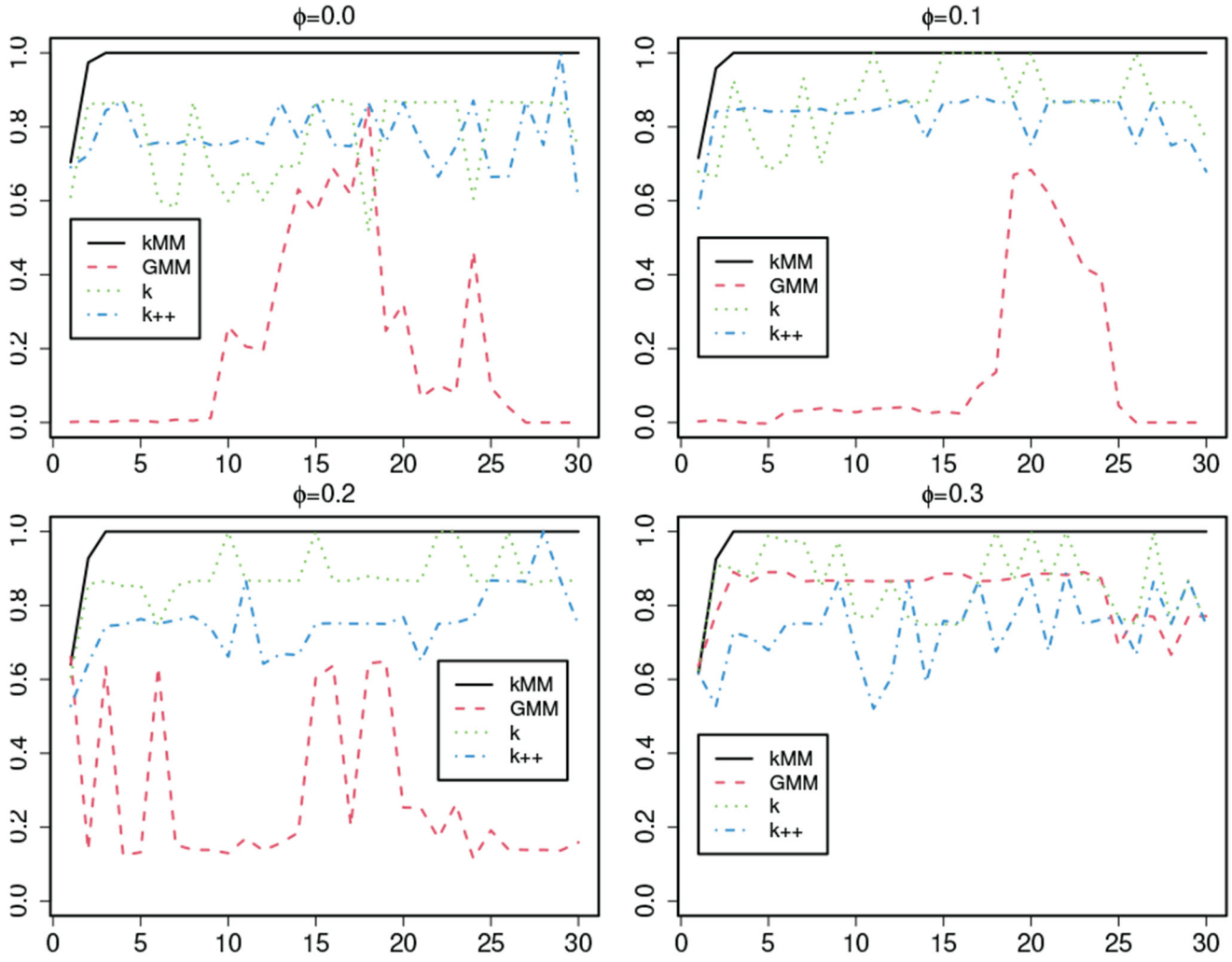

Fig. 1:

ARI curves obtained from a simulated dataset with feature sets selected by Algorithm 1 with respect to the *k*-meansMM (*k*MM), the GMM, the basic *k*-means (*k*), and the *k*-means++ (*k*++) methods, where the horizontal axis represents the number of clusters and the vertical axis represents the ARI values.

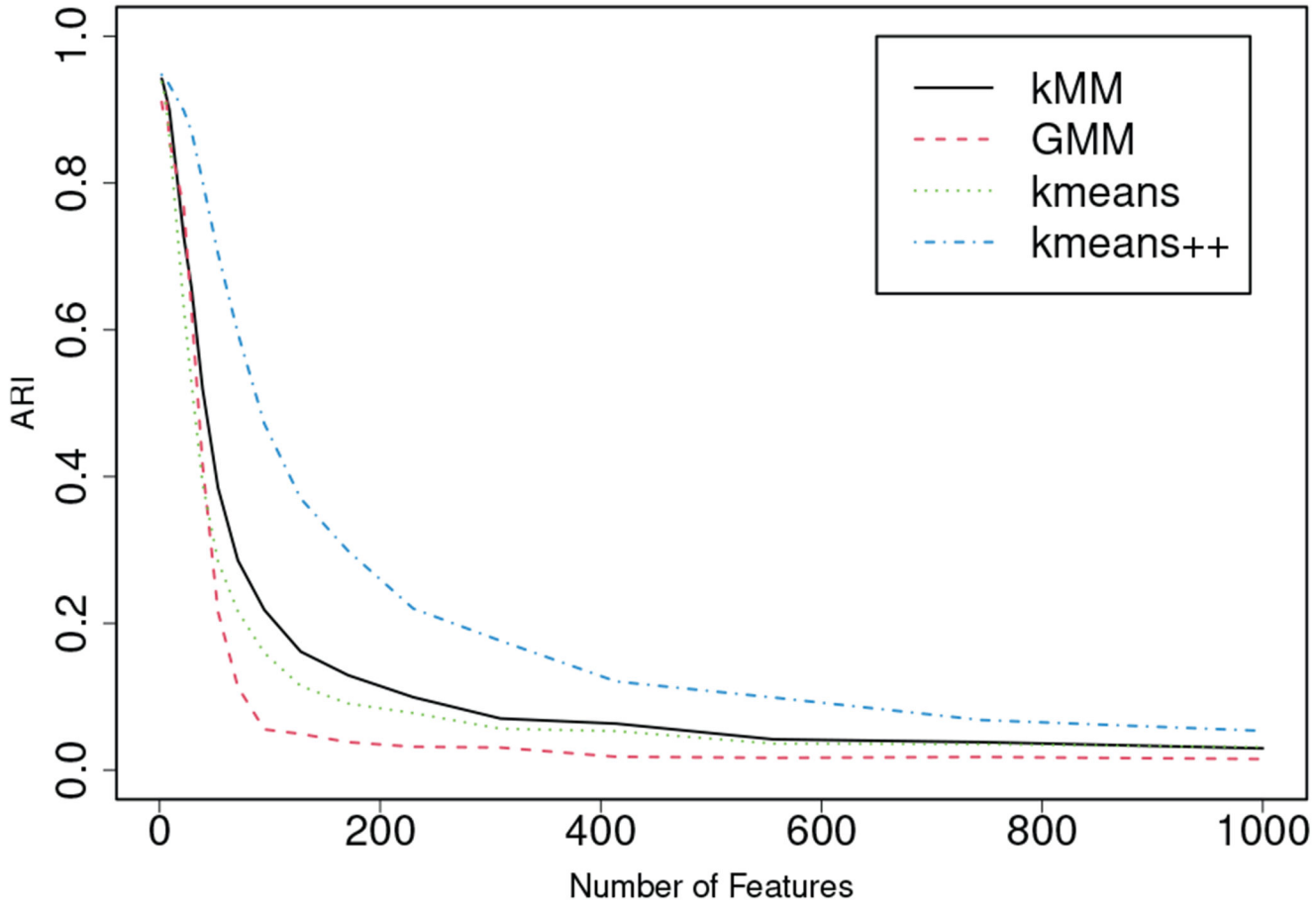

Fig. 2:

ARI curves obtained from simulation with 1000 replications when $k = 2$ and $\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_i, 0.1^2\mathbf{I})$ independently, where $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1, \mu_{11} = \mu_{12} = 0.16, \mu_{1j} = 0$ for all $j \geq 3$.

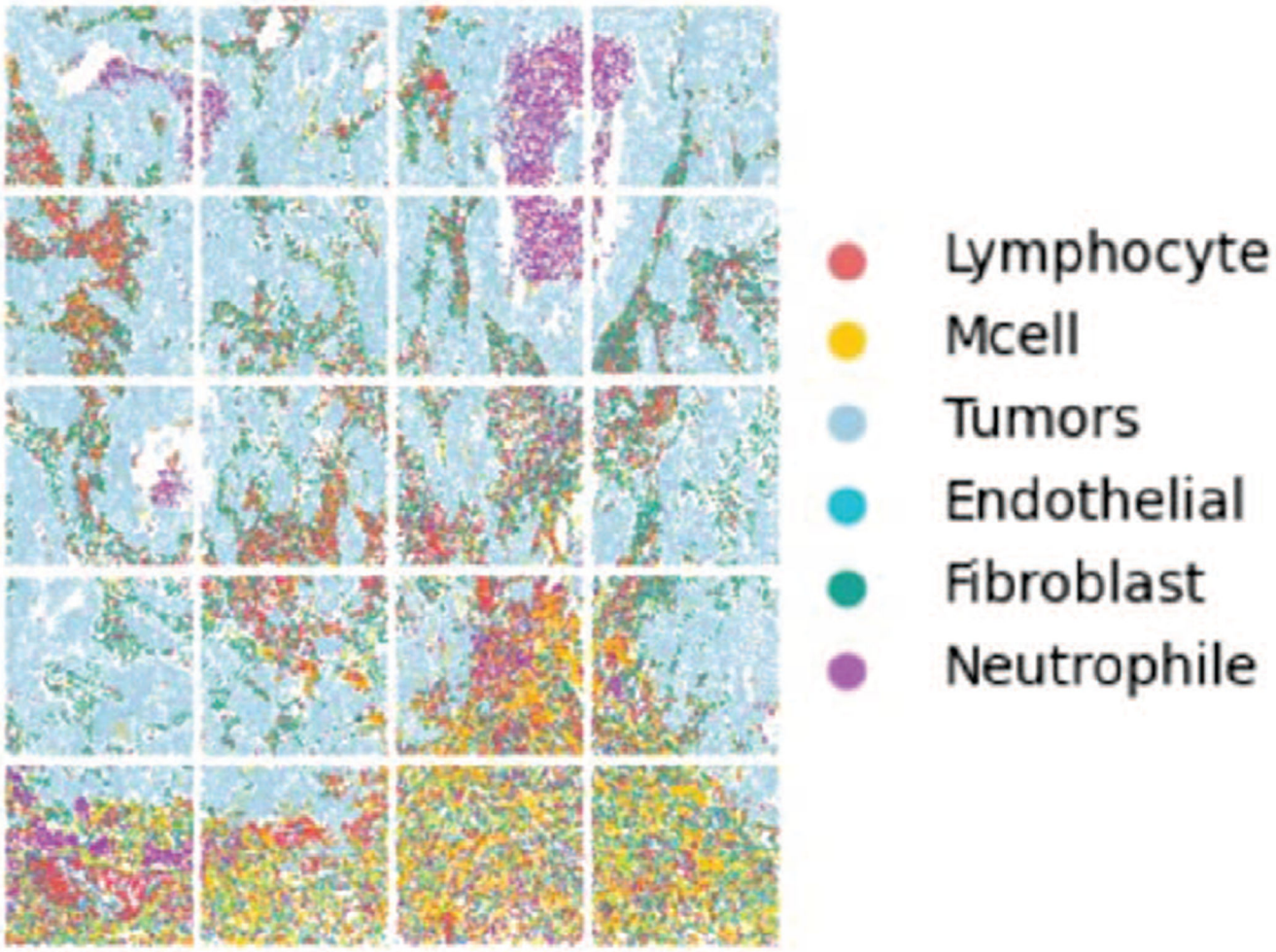

Fig. 3:
The SCST Images for Lung Cancer from the NanoString *CosMx*™ SMI platform

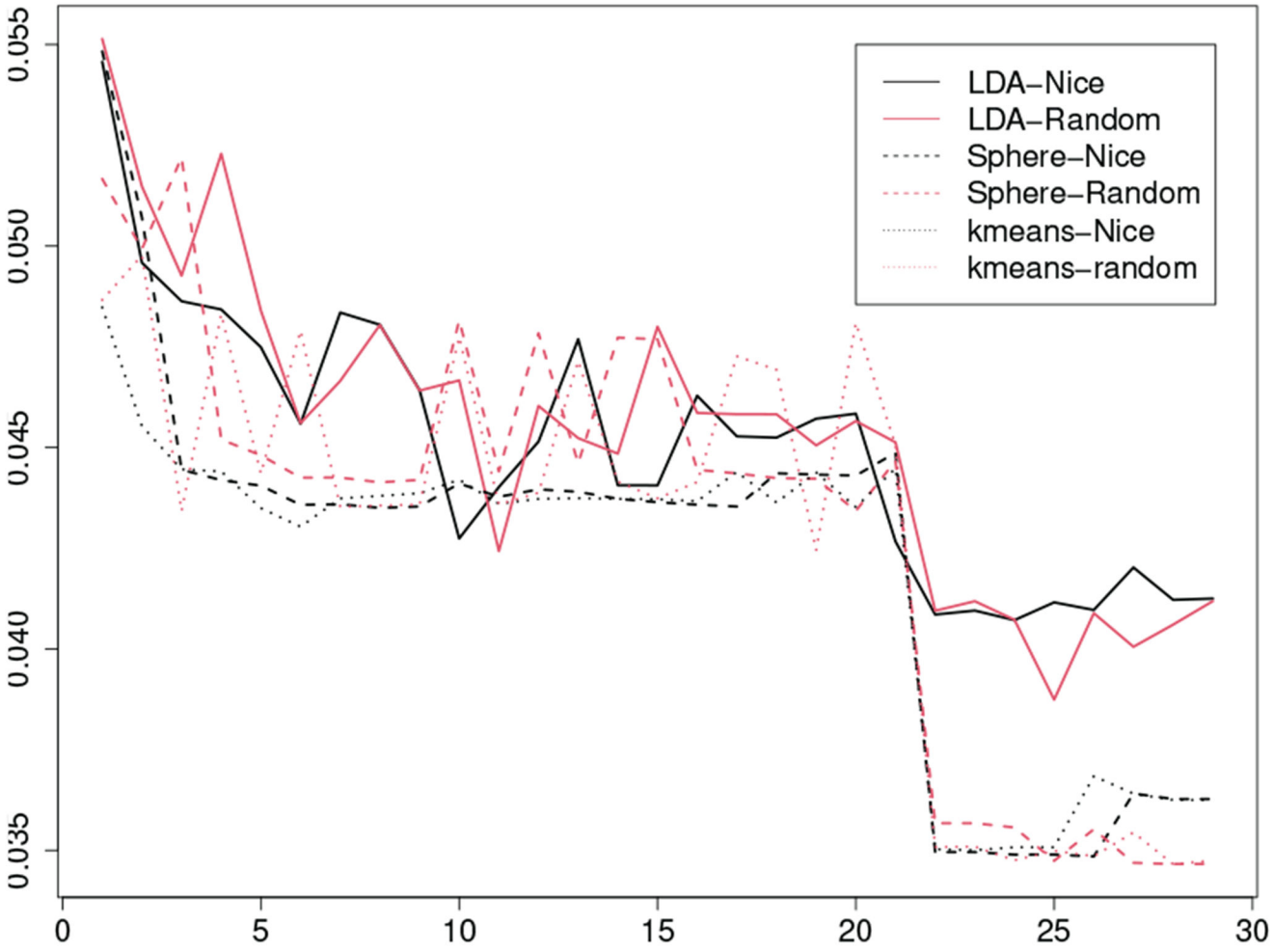

Fig. 4:

The $1 - R^2$ curves for the GMM-LDA, the GMM-Sphere, and the k-means with a nice initialization and a random initialization, respectively, where the horizontal axis represents the number of features and the vertical axis represents the values of $1 - R^2 = SSE/SST$.

TABLE I:

Simulated ARI values obtained from 100 replications for the comparison of feature selection with respect to the k-meansMM (kMM), the GMM, the basic k-means (k), and the k-means++ (k++) methods.

		Number of Features (a)				
Method	ϕ	**1**	**2**	**3**	**4**	**5**
kMM	0.0	0.599	0.928	0.963	0.962	0.963
	0.1	0.595	0.925	0.960	0.958	0.958
	0.2	0.606	0.933	0.987	0.988	0.989
	0.3	0.593	0.938	0.992	0.996	0.996
GMM	0.0	0.357	0.500	0.491	0.483	0.486
	0.1	0.410	0.558	0.543	0.549	0.546
	0.2	0.445	0.607	0.620	0.596	0.602
	0.3	0.442	0.622	0.621	0.608	0.615
k	0.0	0.591	0.764	0.798	0.798	0.801
	0.1	0.577	0.762	0.803	0.813	0.801
	0.2	0.594	0.787	0.812	0.804	0.805
	0.3	0.582	0.779	0.821	0.828	0.827
k++	0.0	0.588	0.761	0.790	0.788	0.783
	0.1	0.580	0.756	0.792	0.800	0.792
	0.2	0.586	0.756	0.786	0.791	0.793
	0.3	0.578	0.783	0.818	0.814	0.818

TABLE II:

ARI of feature selection for the GMM-LDA, the GMM-Sphere, and the k-means clustering methods with random and nice initialization respectively for the SCST multimodal data

	Number of Features (a)				
Method	**26**	**27**	**28**	**29**	**30**
LDA Nice	0.450	0.659	0.536	0.537	0.537
LDA Random	0.378	0.457	0.430	0.500	0.462
Sphere Nice	0.392	0.395	0.534	0.545	0.545
Sphere Random	0.340	0.273	0.349	0.391	0.391
k-means Nice	0.387	0.499	0.518	0.535	0.534
k-means Random	0.376	0.390	0.310	0.389	0.286