Clustering high-dimensional data via feature selection

Tianqi Liu1 | Yu Lu2 | Biqing Zhu3 | Hongyu Zhao3,4

1Google Research, New York, New York, USA
2Two Sigma Investments, New York, New York, USA
3Interdepartmental Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT, USA
4Department of Biostatistics, School of Public Health, Yale University, New Haven, Connecticut, USA

Correspondence
Hongyu Zhao, Department of Biostatistics, School of Public Health, Yale University, New Haven, CT 06510, USA.
Email: hongyu.zhao@yale.edu

Funding information
NIH, Grant/Award Numbers: P50 CA196530, R56AG074015

Abstract
High-dimensional clustering analysis is a challenging problem in statistics and machine learning, with broad applications such as the analysis of microarray data and RNA-seq data. In this paper, we propose a new clustering procedure called spectral clustering with feature selection (SC-FS), where we first obtain an initial estimate of labels via spectral clustering, then select a small fraction of features with the largest $R$-squared with these labels, that is, the proportion of variation explained by group labels, and conduct clustering again using selected features. Under mild conditions, we prove that the proposed method identifies all informative features with high probability and achieves the minimax optimal clustering error rate for the sparse Gaussian mixture model. Applications of SC-FS to four real-world datasets demonstrate its usefulness in clustering high-dimensional data.

KEYWORDS
feature selection, high-dimensional data, spectral clustering

1 INTRODUCTION

Consider a high-dimensional clustering problem, where we observe $n$ vectors $Y_i \in \mathbb{R}^p, i = 1, 2, \ldots, n$, from $k$ clusters with $p > n$. The task is to group these observations into $k$ clusters such that the observations within the same cluster are more similar to each other than those from different ones.

Several statistical methods have been proposed to tackle the high-dimensional clustering problem (Dash and Liu, 2000; Xing and Karp, 2001; Pan and Shen, 2007; Kriegel et al., 2009; Guo et al., 2010; Krishnamurthy, 2011; Song et al., 2011; Witten and Tibshirani, 2012; Jin et al., 2016; Wu et al., 2016; Chakraborty et al., 2020; Liu et al., 2022). A popular choice is to add regularization to encourage sparsity: Pan and Shen (2007) added an $L_1$ penalty on the cluster mean of each feature, Guo et al. (2010) used a pairwise group-fusion penalty to reduce the difference between different groups, Witten and Tibshirani (2012) developed sparse $k$-means and sparse hierarchical clustering via sparse weighted loss of each feature. While the numerical results of these methods were promising, there was no theoretical justification for these methods. Besides enforcing sparsity, several works propose to cluster on latent space via matrix factorization, tensor decomposition, or random projection (Fern and Brodley, 2003; Kriegel et al., 2009; Rohe et al., 2011; Liu et al., 2022). Another way to address the high dimensionality is through feature selection (Dash and Liu, 2000; Xing and Karp, 2001; Chormunge and Jena, 2018). High-dimension feature screening has been well studied under supervised learning (Fan and Lv, 2008; Fan et al., 2009; Balasubramanian et al., 2013; Liu et al., 2016). For unsupervised learning, Jin et al. (2016) proposed influential features Principal Component Analysis (PCA), in which they considered selecting influential features by Kolmogorov–Smirnov (KS) scores. They obtained consistency clustering under the sparse Gaussian mixture model. However, their convergence rate is far from the optimal exponentially small clustering error. And the

[Correction added on 24th December 2022, after first online publication: In algorithm 1 and 3 equation has been added.]
computational cost of calculating KS scores is relatively high.

In this paper, we propose a computationally efficient and provably optimal method to solve high-dimensional clustering problems. Our approach is motivated by recent progress in single-cell RNA sequencing (scRNA-seq) data analysis (Patel et al., 2014; Zeisel et al., 2015; Chen and Zhou, 2018; Zamanighomi et al., 2018; Hao et al., 2021; Su et al., 2021). When clustering cell types from the same tissue, it is natural to assume that most of the genes are not differentially expressed and only cell-type-specific genes can be informative on identifying cell types. We can use pseudolabelling techniques (Lee, 2013) and select informative features in the pseudolabels. Formally, our approach consists of three stages, in which we first obtain an initial estimate of the labels by spectral clustering, and we then select informative features using R-squared of univariate regressions on estimated labels and finally run spectral clustering with Lloyd’s iterations on the selected features. Under mild conditions, we show that the proposed algorithm can successfully identify all informative features. More specifically, given any consistent initial estimate of labels, the second stage of our algorithm selects all informative features with overwhelming probability under the sparse Gaussian mixture model. With these informative features, we are able to run Lloyd iterations in stage three to achieve the optimal misclustering rate (Lu and Zhou, 2016). More specifically, we show that

**Theorem 1.1** (Informal). Under mild sample size and signal-to-noise ratio (SNR) conditions, our three-stage algorithm achieves an exponentially small misclustering rate, which is minimax optimal up to constant in the exponent, with high probability (w.h.p.)

We refer the readers to Theorem 3.4 in Section 3 for the exact conditions we need. Another contribution of our analysis is to derive a faster convergence rate of spectral clustering. Inspired by the recent perturbation results for singular subspaces (Cai and Zhang, 2016), we improve the error rate of spectral clustering from $O(\sqrt{p/n})$ to $O((p/n)^{1/4})$ when $p > n$. Our proposed method provides a new way to efficiently characterize subpopulations in a heterogeneous dataset, identify informative genes, and gain biological insights from high-dimensional datasets such as scRNA-seq data.

The rest of the paper is organized as follows. Section 2 introduces spectral clustering with feature selection (SC-FS) methodology. Theoretical results are provided in Section 3. Section 4 reports the results from numerical studies, including synthetic data study and four real data applications. Finally, we conclude the paper with some remarks and discussions in Section 5.

## 2 METHODOLOGY

In this section, we formally introduce the sparse Gaussian mixture model considered in the paper. Then we present the three stages of our SC-FS algorithm.

### 2.1 Sparse Gaussian mixture model

Suppose there are $k$ clusters with center matrix $B \in \mathbb{R}^{k \times p}$, with rows $B_1, \ldots, B_k \in \mathbb{R}^p$ being centers of clusters. We observe independent samples from the following Gaussian mixture model:

$$Y_i = B_{z_i} + W_i, \quad i = 1, 2, \ldots, n$$

where $\{W_i\}$ are independent sub-Gaussian random vectors satisfying

$$E \exp(\gamma^T W_i) \leq \exp(||\gamma||^2 \sigma^2 / 2)$$

for any $\gamma \in \mathbb{R}^d$ and $z_i \in [k]$ is the cluster label of the $i$th sample. Let $\{p\}$ denote the set $\{1, 2, \ldots, p\}$. For $j \in [p]$, let $\sigma_j^2$ be the marginal variance of the $j$th feature. Here the variances for different features are not necessarily the same. For any subset of $A \subseteq [p]$, denote $\sigma_A = \max_{i \in A} \sigma_i$. Let $T_a$ be the $a$th cluster, that is, $T_a = \{i \in [n], z_i = a\}$ for $a \in [k]$.

As we discussed in the Introduction, there are many noninformative features under the “large $p$, small $n$” scenario. We refer to a feature as noninformative if its within-cluster means are the same across different clusters. Suppose there are $s$ informative features. Then the centers $B_1, \ldots, B_k$ only differ at $s$ coordinates. Without loss of generality, we assume there is a subset $S \subseteq [p]$ with cardinality $s$ such that $B_{ij} = 0$ for all $i \in [k]$ and $j \in S$, where $B_{ij}$ is the $j$th entry of center $B_i$. In practice, we can achieve this by centering and standard scaling each column.

### 2.2 Algorithm

In this section, we present our algorithm for clustering sparse Gaussian mixture data. The algorithm consists of three stages. In the first stage, we obtain an initial estimator of the labels by spectral clustering. Then we perform a feature selection step based on the initial label estimators. Finally, we run spectral clustering and Lloyd’s algorithm on the selected features.
2.2.1 Stage 1: Spectral Clustering

In order to get a good initial estimator of the labels, we first perform de-noising via singular value decomposition (SVD), which preserves the cluster structure on the left eigenvectors under the noiseless case. More precisely, we can rewrite our model (1) as \( Y = ZB + W \), where

\[
Z \in \mathcal{Z} = \{ A \in \{0,1\}^{n \times k}, \|A_{1i}\|_0 = 1, i \in [n]\} \tag{3}
\]

is a membership matrix that has exactly one 1 in each row. Then the SVD of the mean matrix \( ZB \) has the following property.

**Lemma 1.** Let \( UDV^T \) be the SVD of \( ZB \), where \( B \) is full rank. Then \( U = ZQ \) with \( Q \in \mathbb{R}^{k \times k} \) and \( \|Q_{uu} - Q_{uv}\|_2 = \sqrt{\frac{1}{n_u^2} + \frac{1}{n_v^2}} \) for all \( 1 \leq u < v \leq k \). Moreover, \( \sigma_k(ZB) \geq \sqrt{\alpha n \sigma_k(B)} \), where \( \alpha \) is the smallest cluster size.

This lemma is an immediate consequence of Lemma 2.1 of Lei and Rinaldo (2013) by noticing that the left singular vectors of \( ZB \) are orthonormal eigenvectors of \( ZBB^T Z^T \). Lemma 1 implies that there are only \( k \) different rows of \( U \), and we can recover the cluster labels from it. Intuitively, when we have noisy observations of the \( ZB \) matrix, \( \hat{U} \), the leading \( k \) left singular vectors of sample matrix \( Y \), should not differ from \( U \) much. Since the rows of \( U \) are well separated, we could run a distance-based clustering algorithm on the rows \( \hat{U} \) to estimate the labels. Theoretically, the \( k \)-means problem is NP-hard, and hence we use a polynomial-time approximation scheme of \( k \)-means. One possible choice is the \((1 + \epsilon)\)-approximate \( k \)-means algorithm proposed in Kumar and Sabharwal (2004). Another choice is the kmeans++ algorithm (Arthur and Vassilvitskii, 2007). Although kmeans++ is only guaranteed to be a \((1 + \log k)\)-approximation in expectation, it usually enjoys good performance in practice.

The above ideas are summarized in Algorithm 1. We would like to remark that this spectral clustering algorithm is different from the popular one used in the Gaussian mixture literature (Kannan and Vempala, 2009; Kumar and Kannan, 2010; Awasthi and Sheffet, 2012), which runs a clustering algorithm on the best rank \( k \) projections of the data matrix \( Y \). As we shall see in Section 3.1, while these two algorithms theoretically work equally well for the low-dimensional Gaussian mixture models, Algorithm 1 is better for the high-dimensional sparse Gaussian mixtures. Moreover, Algorithm 1 is computationally more efficient since it runs clustering algorithms on an \( n \times k \) matrix \( \hat{U} \), in contrast to the \( n \times p \) matrix using the best rank-\( k \) projections.

**Algorithm 1** Spectral clustering

**Input:** \( Y_1, Y_2, \ldots, Y_n \). The number of clusters \( k \).

**Output:** Estimated clusters \( G_1, G_2, \ldots, G_k \).

1. Compute \( \hat{U} \in \mathbb{R}^{n \times k} \) consisting of the leading \( k \) left singular vectors (ordered in singular values) of \( Y = [Y_1, \ldots, Y_n]^T \).
2. Run \((1 + \epsilon)\)-approximation \( k \)-means on the rows of \( \hat{U} \), that is, find \( \hat{Q} \in \mathbb{R}^{k \times k} \) and \( \hat{Z} \in \mathcal{Z} \) such that
   \[
   \|2\hat{Q} - \hat{Q}\|_F^2 \leq (1 + \epsilon) \min_{Z \in \mathcal{Z}} \|Z - \hat{Q}\|_F^2.
   \]

2.2.2 Stage 2: Feature Selection using R-squared

To select informative features, the first thought would be to compare the sum of squares \( \sum_{i=1}^k Y_{ij}^2 \) of different columns. The larger the sum of squares is, the more likely it is an informative feature. Indeed, when there is no signal, that is, \( j \in S^c \), the sum of squares is a sum of independent chi-square random features with expectation \( n \sigma_j^2 \). And when there is a signal, the expectation of the sum of squares is \( \sum_{a=1}^k n_a^2 B_{aj}^2 + n \sigma_j^2 \). If \( \sigma_j \)'s are the same for all \( j \), one would expect this method to correctly select informative features. However, \( \sigma_j \)'s may vary in practice and we could have some \( j_1 \) and \( j_2 \) such that \( \sum_{a=1}^k n_a^2 B_{aj_1}^2 + n \sigma_{j_1}^2 \ll n \sigma_{j_2}^2 \). To avoid this problem, we need to normalize the variance of each column.

To motivate our feature selection procedure, we consider a special case of symmetric, two balanced clusters with means \( \theta \) and \( -\theta \), where \( \theta \in \mathbb{R}^p \). Let \( T_i \in \{1,2\} \) be the true label of the \( i \)th sample, whose mean is \((2T_i - 3)\theta \). For a noninformative feature \( j \in S^c, \theta_j = 0 \). Thus \( \text{Var}(Y_{ij}|T_i) = \text{Var}(Y_{ij}) \). For informative feature \( j \in S, \theta_j \neq 0 \). For an informative feature, on the other hand, we have \( \text{Var}(Y_{ij}|T_i) < \text{Var}(Y_{ij}) \) for \( j \in S \). Let \( \bar{T}_i \in \{1,2\} \) be the cluster label for the \( i \)th sample obtained from stage 1, it is natural to consider the quantity

\[
R_j^2 = 1 - \frac{\mathbb{E}[\text{Var}(Y_{ij}|\bar{T}_i)]}{\text{Var}(Y_{ij})}.
\]

**Proposition 2.1.** For \( i \)th example, let \( a_{kl} = \mathbb{P}(T_i = k, \bar{T}_i = l) \) for \( k, l \in \{1,2\} \).

\[
R_j^2 = \frac{\theta_j^2}{\theta_j^2 + \sigma_j^2} \left( \frac{(a_{11} - a_{21})^2}{(a_{11} + a_{12})} + \frac{(a_{22} - a_{12})^2}{(a_{22} + a_{12})} \right).
\]

In the case of pure initial random guess \( a_{11} = a_{21} \) and \( a_{22} = a_{12} \), \( R_j^2 = 0 \). If the initial estimator \( \bar{T} \) is slightly better than a random guess, we have \( R_j^2 > 0 \) for the
informative feature. We can distinguish between $j \in S$ and $j \in S^c$ via $R_j^2$. Besides, when $j \in S$, $R_j^2$ depends on \( \text{SNR}_j^2 / \sigma_j^2 \). The higher the SNR, the weaker condition we need on the initial estimator to get the same $R_j^2$. We refer to Section 3.2 for our detailed analysis of the sample version and the general number of clusters.

### Stage 3: Spectral clustering and Lloyd’s algorithm

With the features selected in stage 2, the problem is reduced to low-dimensional Gaussian mixtures, which have been studied extensively in the literature. Among them, the most popular algorithms for Gaussian mixtures are Lloyd’s algorithm (Lloyd, 1982), Expectation Maximization (EM) algorithm (Dempster et al., 1977), methods of moments (Lindsay and Basak, 1993), and tensor decompositions (Anandkumar et al., 2012). For stage 3, we use the spectral clustering Algorithm 1 on selected features, followed by Lloyd's iterations. Lloyd’s algorithm, often referred to as the $k$-means algorithm, enjoys good statistical and computational guarantees for Gaussian mixture models (Lu and Zhou, 2016). Given an initial estimator of the labels or centers, it iteratively updates the labels and centers on the selected features until convergence. A precise description is given in Algorithm 3. We refer the readers to Lu and Zhou (2016) for more discussions on Lloyd’s algorithm.

In summary, we first conduct spectral clustering to estimate noisy cluster labels, then we apply $R^2$ to select top informative features, and finally, we apply spectral clustering again on selected features. To further reduce the error, we apply Lloyd’s algorithm after the last stage.

## 3 | CONVERGENCE ANALYSIS

To better present our theoretical results, let us first introduce some notations and assumptions. For any partition

### Algorithm 2: Feature selection using $R^2$

**Input:** $Y_1, Y_2, \ldots, Y_n$. The number of clusters $k$. Initial estimates of clusters $G_1, G_2, \ldots, G_k$. A threshold $\tau \in (0,1)$.

**Output:** An index set $\hat{S}$.

1. For $j = 1, 2, \ldots, p$, calculate:
   1a. Estimated centers: $\hat{B}_{ij} = \frac{1}{|G_i|} \sum_{i \in G_i} Y_{ij}$
   1b. Residual sum of squares: $c_j = \sum_{i=1}^k \sum_{i \in G_i} (Y_{ij} - \hat{B}_{ij})^2$
   1c. Total sum of squares: $m_j = \sum_{i \in [n]} (Y_{ij} - \hat{\bar{Y}}_j)^2$
   1d. Score: $SC_j = \frac{c_j}{m_j}$
2. Output $\hat{S} = \{j \in [n], SC_j \leq \tau\}$

### Algorithm 3: Spectral Lloyd algorithm

**Input:** $Y_1, Y_2, \ldots, Y_n$. The number of clusters $k$. An index set of selected features $\hat{S}$.

**Output:** Estimated cluster labels $z^{(T)}_1, z^{(T)}_2, \ldots, z^{(T)}_n$.

1. Run Algorithm 1 on $\{\hat{Y}_i\}$ to get an initial estimate of labels, $z^{(0)}_1, z^{(0)}_2, \ldots, z^{(0)}_n$, where $\hat{Y}_i$ is the subvector of $Y_i$ with support $\hat{S}$.
2. Run the following iterations for $t = 1, 2, \ldots, T$.
   2a. For $(a, j) \in [k] \times \hat{S}$, calculate:
      
      $$\hat{B}_{aj} = \frac{\sum_{i \in \hat{S}} Y_{ij} 1[z_j^{(t-1)} = a]}{\sum_{i \in [n]} 1[z_i^{(t-1)} = a]}$$
   2b. For $i \in [n]$, calculate:
      
      $$z_i^{(t)} = \arg \min_{a \in [k]} \sum_{j \in \hat{S}} (Y_{ij} - \hat{B}_{aj})^2$$

### The following theorem provides an upper bound on group-wise misclustering error of spectral clustering algorithm 1 for the Gaussian mixture model.

### 3.1 | Error rate of spectral clustering

The following theorem provides an upper bound on group-wise misclustering error of spectral clustering algorithm 1 for the Gaussian mixture model.
Theorem 3.1. Let G be the partition returned by Algorithm 1 and B_S be the submatrix of B consisting of s nonzero columns. Assume the k-th singular value
\[ \sigma_k(B_S) \geq C \max \left\{ \sigma \sqrt{\frac{k}{\alpha}} \left( \frac{\sigma^2 k p}{\alpha^2 n} \right)^{1/4} \right\} \]
for a sufficiently large constant C. Then the group-wise misclustering error rate
\[ B(G, T) \leq \frac{C \sigma^2 k (\alpha n \sigma^2(B_S) + p)}{\alpha^3 n \sigma^2(B_S)} \]
with a probability greater than 1 − exp(−C_2n) for some universal constants C_1 and C_2.

It guarantees a relatively small misclustering error, for example, 10%, under condition (8). It only has a (p/n)^1/4 dependence on the dimensionality of the problem in condition (8). Thus it is applicable to the high-dimensional problem and can be satisfied under many interesting cases. For example, when B_S is a random matrix, its minimum eigenvalue can be lower bounded by c \sqrt{s} for some constant c with high probability (Vershynin, 2010), where s is the number of informative features. Then condition (8) is reduced to s \geq \max\{\sigma^2, \sigma(p/n)^{1/4}\} by regarding k and \alpha as constants.

As discussed in Section 2.2.1, another version of the spectral clustering algorithm is to run a distance-based clustering algorithm on the rows of \( \hat{Y} \), the rank-k approximation of the data matrix \( \hat{Y} \), instead of on the estimated eigenspace \( \hat{U} \). The condition (Awasthi and Sheffet, 2012; Lu and Zhou, 2016) we need for this spectral clustering algorithm is
\[ \min_{u \neq v \in [k]} \|B_{u, v} - B_{u, v}\| \geq C_4 \sigma \sqrt{\frac{k}{\alpha}} \left( 1 + \frac{kp}{n} \right) \]
for some sufficiently large constant C_4, since there are only s nonzero entries of each row of B. It requires s \geq \sigma \sqrt{p/n} when k and \alpha are constants. Thus, Algorithm 1 works better for the high-dimensional setting.

3.2 Feature selection guarantees

The next theorem provides theoretical guarantees of the feature selection step.

Theorem 3.2. Assume SNR > C_0 for some sufficiently large constant C_0. Then there exists a constant c such that for any given estimated partition G (could be data dependent) with B(G, T) \leq ca.

(a) When j \in S, we have SC_j \leq 0.9 with a probability greater than 1 − exp(−cn).
(b) When j \in S^c, we have SC_j > 0.9 with a probability greater than 1 − exp(−c\alpha n).

Therefore, when an = \Omega(\log p), a choice of \( \tau = 0.9 \) successfully selects all the informative features with a probability greater than 1 − exp(−c\alpha n).

Given any initializer with B(G, T) \leq ca, we are guaranteed to select all the informative features with high probability when an = \Omega(\log p). It implies that the number of features p is allowed to grow exponentially fast of the sample size n. Such scaling also appears in the feature selection problem under the sparse linear regression model (Wainwright, 2009). Since feature selection only depends on the error rate of the initial guess, we can also choose other clustering approaches in stage 1 as long as the error rate is satisfactory.

3.3 Error rate of the Lloyd’s algorithm

Finally, we have the following result from Lu and Zhou (2016) to characterize the performance of Lloyd’s algorithm.

Theorem 3.3. Let \( \Delta = \min_{u \neq v \in [k]} \|B_{u, v} - B_{u, v}\| \). Assume n\alpha^2 \geq Ck \log n, n \geq ks, and \( \Delta \geq C\sigma_s \sqrt{k/\alpha} \) for a sufficiently large constant C. Given any initializer \( G_0 \) satisfying
\[ B(G_0, T) < \min_{u \neq v \in [k]} \|B_{u, v} - B_{u, v}\| = \frac{1}{4\lambda} \]
with probability 1 − \nu. Then
\[ \frac{1}{n} \sum_{i=1}^{n} I[z_i^{(s)} \neq z_i] \leq \exp \left( -\frac{\Delta^2}{16\sigma^2_S} \right) \]
for all s \geq 4\log n with probability greater than 1 − \nu − 4/n − 2\exp(−\Delta/\sigma_S).

Theorem 3.3 states that we can achieve an exponentially small misclustering error after [4 \log n] Lloyd’s iterations given any initializer that satisfies condition (11). Suppose we have selected all the informative features in stage 2. By applying Theorem 3.1 on the submatrix B_S, we obtain
\[ B(G_0, T) \leq \frac{C_1 \sigma^2_k}{\alpha^2 \sigma^2(B_S)} \leq \frac{1}{4\lambda} \]
when \( \sigma_k(B_S) \geq C_2 \sigma_S \sqrt{\lambda k/\alpha^2} \) for some sufficiently large constant C_2.
Combining the results of Theorems 3.1–3.3, we are able to give theoretical guarantees of our SC-FS algorithm. Let \( \tilde{z} = \{\tilde{z}_1, ..., \tilde{z}_n\} \) be the estimated labels returned by running the SC-FS algorithm with \( \tau = 0.9 \) and \( T = [4\log n] \). The following result upper bounds the misclustering error rate of \( \tilde{z} \).

**Theorem 3.4.** Assume \( an \geq C(\log p + k \log n / \alpha + \alpha ks) \), \( SNR \geq C, \Delta \geq C\sigma_s \sqrt{k/\alpha} \) and

\[
\sigma_k(B_S) \geq \frac{C}{\alpha} \max \left\{ \sigma \sqrt{\frac{k}{\alpha}}, \sigma_S \sqrt{\lambda k}, \left( \frac{\sigma_k^2 p}{\alpha^2 n} \right)^{1/4} \right\}
\]

for a sufficiently large constant \( C \). Then

\[
\frac{1}{n} \sum_{i=1}^{n} 1[\tilde{z}_i \neq z_i] \leq \exp \left( -\frac{\Delta^2}{16\sigma^2_S} \right),
\]

with a probability greater than \( 1 - 8/n - 4\exp(-\Delta/\sigma_S) \).

By Theorem 3.3 in Lu and Zhou (2016), the minimax lower bound for clustering the Gaussian mixture model is \( \exp(-\Delta^2 / 8\sigma^2_S) \). The worst case constructed in Lu and Zhou (2016) can be naturally generalized to the sparse Gaussian mixture model. Therefore, the proposed SC-FS algorithm is rate optimal up to a constant factor in the exponent. Note that the misclustering rate only takes value in \( \{0, 1/n, 2/n, ..., 1\} \). Theorem 3.4 guarantees a perfect clustering when \( \Delta > 4\sigma_S \log n \).

### 3.4 Tuning parameter selection

#### 3.4.1 Number of clusters

For each possible \( k = 1, ..., 20 \), we conduct the following steps:

1. Conduct SVD on data matrix and obtain top \( k \) left singular vectors as matrix \( U \in \mathbb{R}^{p \times k} \).
2. Conduct the \( k \)-means clustering algorithm of \( U \).
3. Calculate the ratio of within-cluster sum of squares and the total sum of squares as the unexplained variation ratio \( \eta(k) \). And let \( \bar{\eta}(k) = 1 - \eta(k) \) be the variation explained ratio.

We plot \( \bar{\eta}(k) \) versus \( k \) and select the change point as the number of clusters.

#### 3.4.2 Feature selection threshold

The actual threshold depends on the error rate of the initializer and the quality \( an/\log p \). As suggested by Theorem 3.2, we could use \( \tau = 0.9 \) as practical guidance of the feature selection threshold.

### 4 Numerical Experiments

#### 4.1 Synthetic data generation

Let \( k \) be the number of clusters, \( n \) be the number of samples, \( p \) be the number of features, \( s \) be the number of informative features, and \( \sigma_k \) be the signal strength introduced in Theorem 3.1. For a set of \( (k, n, p, s, \sigma_k) \), we generate data as follows:

1. Generate elements of \( B \in \mathbb{R}^{k \times s} \) as left singular matrix of independently and identically distributed (i.i.d.) \( s \times s \) standard Gaussian random matrix. We get \( B \in \mathbb{R}^{k \times p} \) as \( B = [\sigma_1 \tilde{B}, 0_{k \times (p-s)}] \).
2. Generate the cluster label \( z_i \in \{1, ..., k\} \) of the \( i \)th sample by randomly assigning. Then generate membership matrix \( Z \in \mathbb{R}^{n \times k} \) with \( Z_{ij} = 1 \) (if \( j = z_i \)).
3. Generate data matrix \( Y = ZB + W \), where \( W \) is the standard Gaussian noise matrix (or \( t_2 \) noise matrix if specified). Then we scale the columns of the data matrix.

#### 4.2 Convergence rate of spectral clustering

In this simulation, we numerically evaluated the convergence rate of spectral clustering. To study the effect of the number of features \( p \) on the error rate of spectral clustering, we fixed the number of clusters \( k = 4 \), the number of observations \( n = 100 \), the number of features \( p = 100 \), the number of informative features \( s = 100 \), and the signal strength \( \sigma_k = 4 \). We varied \( p \) from 100 to 1000, \( n \) from 100 to 1000, and \( \sigma_k \) from 2 to 5 to study the error convergence rate regarding each factor \( (n, p, \sigma_k) \) with two other factors fixed. For each setting of \( (k, n, p, s, \sigma_k) \), we generated synthetic data according to Section 4.1 with Gaussian noise and applied spectral clustering according to Algorithm 1. We repeated the above process 50 times and computed the average error rate. The scatter plots are shown in Figure 1. We observe a linear relationship between the error rate and \( p \) and also the expected rate for \( n \) and \( \sigma_k \).

In terms of spectral clustering with sparse informative features, we can improve the clustering result to a great extent if the number of informative features \( s \) is much smaller than the total number of features, given that we have selected all informative features. Even if we fail to select all informative features, we can still have a better clustering result as long as we have selected enough features such that the SNR does not decrease too much after feature selection.
FIGURE 1  Convergence rate of the error rate

TABLE 1  Feature selection $F_1$ scores averaged over 50 runs. Numbers in the brackets are the standard deviations

| $\sigma_k$ | $n/log p$ | Initial guess error rate | 0.05   | 0.1    | 0.15   | 0.2    | 0.3    |
|------------|-----------|--------------------------|--------|--------|--------|--------|--------|
| 5          | 10        |                          | 0.620  | 0.604  | 0.578  | 0.540  | 0.456  |
|            | 50        |                          | 0.744  | 0.698  | 0.626  | 0.548  | 0.311  |
| 5          | 100       |                          | 0.742  | 0.671  | 0.593  | 0.507  | 0.256  |
| 10         | 10        |                          | 0.736  | 0.731  | 0.720  | 0.701  | 0.664  |
| 10         | 50        |                          | 0.958  | 0.949  | 0.935  | 0.909  | 0.821  |
| 10         | 100       |                          | 0.956  | 0.948  | 0.932  | 0.908  | 0.816  |

4.3 Feature selection $F_1$

In this simulation, we studied the relationship between feature selection success metrics and quality of initial guess. We fixed $k = 4$, $s = 100$, $p = 500$, and varied $n \in \{10, 50, 100\} \log p$. Let the true label of the $i$th observation be $l_i$, and the initial guessed label be $\hat{l}_i$. We define the initial guess error rate as

$$M(l, \hat{l}) = \frac{1}{n} \min_{\pi} \left| \left\{ i : l_i \neq \pi(\hat{l}_i) \right\} \right|.$$  \hspace{1cm} (16)

We create guessed labels with the given error rate taking values from $\{0.05, 0.1, 0.15, 0.2, 0.3\}$. We set $\sigma_k \in \{5, 10\}$.

Let $S$ be the set of true informative features with $|S| = s$, and $\hat{S}$ be the set of estimated informative features based on $R^2$ Algorithm 2. We compute the $F_1$ score to measure the feature selection quality. For a set of $(n, p, s, \sigma_k)$, we generated data as described in Section 4.1 and repeated the experiment 50 times. Given the membership matrix $Z$, we generated the guessed label $\hat{Z}$ equal to $Z$ with probability $1 - \eta$, and equal to one of other $k - 1$ values with equal probability $\eta/(k - 1)$.

We can observe that as signal strength $\sigma_k$ increases, $n$ increases, and misclustering rate of initial guess decreases, the feature selection performance improves (Table 1). When the signal strength and number of samples are large enough, the selected features are of high quality. This observation is consistent with Equation (5) and Theorem 3.2.

4.4 Comparisons of synthetic data

4.4.1 Gaussian noise

In this simulation, we fixed $k = 4$, $p = 8000$, $s = 500$, $\sigma_k = 6$, and $n/log p = 15, 20, 25, 30$. We generated synthetic data according to Section 4.1. We denote SC-FS1 as spectral clustering in stage 3, and SC-FS2 as Lloyd iteration following SC-FS1. We compared our methods SC-FS1 and SC-FS2 with spectral clustering, spectral plus Lloyd clustering (specLloyd, for short) (Lu and Zhou, 2016), model-based clustering (mclust) (Scrucca et al., 2016), and sparse K-means (spKmeans, for short) (Witten and Tibshirani, 2012). As shown in Table 2, our proposed methods performed the best and Lloyd’s iteration in stage 3 improved SC-FS1 to a small extent. By comparing specLloyd with the proposed method, we can observe
that feature selection in stage 2 can reduce the error rate.

4.4.2 Heavy-tailed noise

In this simulation, we compare the methods in a heavy-tailed noise case to study the robustness of the proposed method. We followed the same setting as in Section 4.4.1 in generating the synthetic data, except that we used standard $t_2$ distribution to generate noise. The proposed approach shows an advantage under the heavy-tailed noise case (Table 3), while spKmeans does not converge well with sample size growth. To some extent, this suggests that our proposed approach is robust to heavy-tailed noise.

4.5 Real data

4.5.1 Dataset description

We compared clustering results of our method with other methods on four publicly available high-dimensional datasets. We selected these datasets because they represent a wide range of high-dimensional data with different numbers of data points and classes from various fields. Characteristics of the four real datasets are summarized in Table 4. The details of four datasets are as follows:

(1) Zheng: The peripheral blood mononuclear cells scRNA-seq data were generated by the 10x Genomics GemCode protocol. We obtained the data from the package DuoClustering2018 (Duò et al., 2019) with ExperimentHub ID “EH1532.” The data consist of eight cell types in approximately equal proportions.

(2) Yeoh: The bone marrow microarray data were downloaded from the R package datamicroarray (Ramey, 2016). The 248 samples were obtained from pediatric acute lymphoblastic leukemia patients with six subtypes, including T-ALL, E2A-PBX1, TEL-AML1, BCR-ABL, MLL, and HK50. The number of features, that is, genes, is 12,625.

(3) BBC: This dataset has 2225 articles with 1490 for training and 735 for testing. Each article has one label from five categories: business, entertainment, politics, sports, or tech. We downloaded the data from Greene and Cunningham (2006) and used the training data to compare among different clustering algorithms. We did not use test data because there are no labels available from the dataset. The 1490 articles with five categories were processed by the term frequency-inverse document frequency (tf-idf) vectorizer. We obtained 24,746 features as a result.

(4) Agnews: This dataset is a collection of more than 1 million news articles. The AG’s news topic classification dataset was constructed by choosing the four largest
TABLE 5  Adjusted Rand index on four real datasets

| Dataset  | SC-FS1 | SC-FS2 | spectral | spKmeans | Kmeans |
|----------|--------|--------|----------|----------|--------|
| Zheng    | 0.431  | 0.437  | 0.330    | 0.418    | 0.319  |
| Yeoh     | 0.647  | 0.579  | 0.554    | 0.337    | 0.258  |
| BBC      | 0.647  | 0.658  | 0.647    | 0.0440   | 0.573  |
| agnews   | 0.192  | 0.205  | 0.201    | 0.0151   | 0.180  |

classes from the original corpus. Each class contains 30,000 training samples and 1900 testing samples. The total number of training samples is 120,000 and that of testing samples is 7600. We downloaded the data from Zhang et al. (2015) and used the test set to compare different clustering algorithms. We used the test data because it has thousands of examples with tens of thousands of features (after tf-idf), which fits the high-dimensional setting. The 7600 articles with four categories are also processed by the tf-idf vectorizer. We obtained 21,853 features as a result.

4.5.2  Numerical comparisons among different methods

We performed comparisons of SC-FS on the four datasets to test its performance with three other methods including spectral clustering (Rohe et al., 2011), sparse K-means (Witten and Tibshirani, 2012), and K-means (MacQueen et al., 1967). For sparse K-means, we subsampled 1500 data points for Zheng, Yeoh, and agnews to avoid run time and memory issues. The adjusted Rand index (ARI) is shown in Table 5. SC-FS2 performed the best on three out of four datasets, and SC-FS1 resulted in the highest ARI on the remaining dataset, followed by spectral clustering.

5  CONCLUSIONS

In this article, we proposed a three-stage algorithm that is minimax optimal for estimating the underlying cluster labels under the generative model of the sparse Gaussian mixture model (1). Our method is able to identify all informative features given any initial estimator with o(1) clustering error and theoretically verified the optimality of the proposed method under sparse Gaussian mixture assumptions. We further demonstrated the power of the methods via extensive simulation studies and real data analysis. For further directions, it is interesting to explore the performance of our algorithm under other generative models with heavy tails. Based on the proposed framework, it is also interesting to compare other clustering and feature selection methods including nonlinear methods such as kernel methods and neural networks.

DATA AVAILABILITY STATEMENT

The data that support the findings in this paper are openly available in Kaggle BBC (Broadcasting company) News Classification at https://www.kaggle.com/c/learn-ai-bbc, and AG News at https://github.com/mhjabreel/CharCNN_Keras/tree/master/data/ag_news_csv.

OPEN RESEARCH BADGES

This article has earned Open Data and Open Materials badges. Data and code are available at https://doi.org/10.7910/DVN/DHLRSI.

ORCID

Tianqi Liu https://orcid.org/0000-0003-4497-3317
Biqing Zhu https://orcid.org/0000-0002-7428-6297
Hongyu Zhao https://orcid.org/0000-0003-1195-9607

REFERENCES

Anandkumar, A., Hsu, D. and Kakade, S.M. (2012) A method of moments for mixture models and hidden Markov models. In 25th Annual Conference on Learning Theory. JMLR: Workshop and Conference Proceedings, 23, 33.1–33.34
Arthur, D. and Vassilvitskii, S. (2007) k-means++: the advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 1027–1035.
Awasthi, P. and Sheffet, O. (2012) Improved spectral-norm bounds for clustering. In: Goemans, M., Jansen, K., Rolim, J.D.P., and Trevisan, L. (Eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques. Berlin: Springer, pp. 37–49.
Balasubramanian, K., Sriperumbudur, B. and Lebanon, G. (2013) Ultrahigh dimensional feature screening via RKHS embeddings. In: 16th International Conference on Artificial Intelligence and Statistics (AISTATS 2013), Scottsdale, AZ. Journal of Machine Learning Research, 31, 126–134.
Cai, T.T. and Zhang, A. (2016) Rate-optimal perturbation bounds for singular subspaces with applications to high-dimensional statistics [Preprint], arXiv:1605.00353.
Chakraborty, S., Paul, D., Das, S. and Xu, J. (2020) Entropy weighted power k-means clustering. In: International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, 108, 691–701
Chen, M. and Zhou, X. (2018) Viper: variability-preserving imputation for accurate gene expression recovery in single-cell RNA sequencing studies. Genome Biology, 19, 1–15.
Chormunse, S. and Jena, S. (2018) Correlation based feature selection with clustering for high dimensional data. Journal of Electrical Systems and Information Technology, 5, 542–549.

Dash, M. and Liu, H. (2000) Feature selection for clustering. In: Terano, T., Liu, H., Chen, A.L.P. (Eds.) Knowledge Discovery and Data Mining. Current Issues and New Applications, (Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000). Lecture Notes in Computer Science, Vol. 1805. Berlin: Springer, pp. 110–121.

Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39, 1-22.

Duò, A., Soneson, C., Duò, M.A., biocViews SingleCellData, E., ExperimentHub, I. and SingleCellExperiment, S. (2019) Package ‘duoClustering2018’.

Fan, J. and Lv, J. (2008) Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70, 849–911.

Fan, J., Samworth, R. and Wu, Y. (2009) Ultrahigh dimensional feature selection: beyond the linear model. The Journal of Machine Learning Research, 10, 2013–2038.

Fern, X.Z. and Brodley, C.E. (2003) Random projection for high dimensional data clustering: a cluster ensemble approach. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), Palo Alto, CA: AAAI Press, 186–193.

Greene, D. and Cunningham, P. (2006) Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proceedings of the 23rd International Conference on Machine learning (ICML’06). New York: ACM Press, pp. 377–384.

Guo, J., Levina, E., Michailidis, G. and Zhu, J. (2010) Pairwise variable selection for high-dimensional model-based clustering. Biometrics, 66, 793–804.

Hao, Y., Hao, S., Andersen-Nissen, E., Mauck III, W.M., Zheng, S., Butler, A., et al. (2021) Integrated analysis of multimodal single-cell data. Cell, 184(15), 3573-3587.e29.

Jin, J., Wang, W., et al. (2016) Influential features PCA for high dimensional clustering. The Annals of Statistics, 44, 2323–2359.

Kannan, R. and Vempala, S. (2009) Spectral algorithms. Foundations and Trends in Theoretical Computer Science, 4, 157–288.

Kriegel, H.-P., Kröger, P. and Zimek, A. (2009) Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM Transactions on Knowledge Discovery from Data (TKDD), 3, 1–58.

Krishnamurthy, A. (2011) High-dimensional clustering with sparse Gaussian mixture models [Unpublished paper]. Carnegie Mellon University.

Kumar, A. and Kannan, R. (2010) Clustering with spectral norm and the k-means algorithm. In 2010 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS). Piscataway, NJ: IEEE Press, pp. 299–308.

Kumar, A. and Sabharwal, Y. (2004) A simple linear time (1+ epsilon)-approximation algorithm for k-means clustering in any dimensions.

Lee, D.-H. (2013) Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks. In: ICML2013: Workshop on Challenges in Representation Learning, volume 3.

Lei, J. and Rinaldo, A. (2013) Consistency of spectral clustering in sparse stochastic block models [Preprint]. arXiv:1312.2050.

Lindsay, B.G. and Basak, P. (1993) Multivariate normal mixtures: a fast consistent method of moments. Journal of the American Statistical Association, 88, 468–476.

Liu, T., Lee, K.-Y. and Zhao, H. (2016) Ultrahigh dimensional feature selection via kernel canonical correlation analysis [Preprint]. arXiv:1604.07354.

Liu, T., Yuan, M. and Zhao, H. (2022) Characterizing spatiotemporal transcriptome of the human brain via low-rank tensor decomposition. Statistics in Biosciences. Advance online publication. https://doi.org/10.1007/s12561-021-09331-5

Lloyd, S. (1982) Least squares quantization in PCM. IEEE Transactions on Information Theory, 28, 129–137.

Lu, Y. and Zhou, H.-H. (2016) Statistical and computational guarantees of Lloyd’s algorithm and its variants [Preprint]. arXiv:1612.02099.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Oakland, CA, volume 1. Berkeley, CA: University of California Press, pp. 281–297.

Pan, W. and Shen, X. (2007) Penalized model-based clustering with application to variable selection. Journal of Machine Learning Research, 8, 1145–1164.

Patel, A.P., Tirosi, I., Trombetta, J.J., Shalek, A.K., Gillespie, S.M., Wakimoto, H., et al. (2014) Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. Science, 344, 1396–1401.

Ramey, J. (2016) Datamicroarray: collection of data sets for classification. Available at: https://github.com/ramhiser/datamicroarray.

Rohe, K., Chatterjee, S. and Yu, B. (2011) Spectral clustering and the high-dimensional stochastic blockmodel. The Annals of Statistics, 39(4), 1878–1915.

Scrucca, L., Fop, M., Murphy, T.B. and Raftery, A.E. (2016) mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. The R Journal, 8, 289.

Song, Q., Ni, J. and Wang, G. (2011) A fast clustering-based feature subset selection algorithm for high-dimensional data. IEEE Transactions on Knowledge and Data Engineering, 25, 1–14.

Su, K., Yu, T. and Wu, H. (2021) Accurate feature selection improves single-cell RNA-seq cell clustering. Briefings in Bioinformatics, 22(5), bbab034.

Vershynin, R. (2010) Introduction to the non-asymptotic analysis of random matrices [Preprint]. arXiv:1011.3027.

Wainwright, M.J. (2009) Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ1-constrained quadratic programming (Lasso). IEEE Transactions on Information Theory, 55, 2183–2202.

Witten, D.M. and Tibshirani, R. (2012) A framework for feature selection in clustering. Journal of the American Statistical Association, 105(490), 713–726.

Wu, C., Kwon, S., Shen, X. and Pan, W. (2016) A new algorithm and theory for penalized regression-based clustering. Journal of Machine Learning Research, 17, 1–25.
Xing, E.P. and Karp, R.M. (2001) Cliff: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17, S306–S315.

Zamanighomi, M., Lin, Z., Daley, T., Chen, X., Duren, Z., Schep, A., et al. (2018) Unsupervised clustering and epigenetic classification of single cells. *Nature Communications*, 9, 1–8.

Zeisel, A., Muñoz-Manchado, A.B., Codeluppi, S., Lönnerberg, P., La Manno, G., Juréus, A., et al. (2015) Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347, 1138–1142.

Zhang, X., Zhao, J. and LeCun, Y. (2015) Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28, 649–657.

**SUPPORTING INFORMATION**

Web Appendices, Tables, and Figures referenced in Section 4 are available with this paper at the Biometrics website on Wiley Online Library. The code is available both on the Biometrics website and at [https://github.com/TerenceLiu4444/SCFS](https://github.com/TerenceLiu4444/SCFS).

**How to cite this article:** Liu, T., Lu, Y., Zhu, B. & Zhao, H. (2023) Clustering high-dimensional data via feature selection. *Biometrics*, 79, 940–950. https://doi.org/10.1111/biom.13665