Automatic Repulsive Clustering for Finding Separable Linear Subspace

Leo L. Duan and Joshua T. Vogelstein

Abstract

Mixture model based clustering methods are routinely used to divide the heterogeneous data into groups, within each the data are similar and characterized by a center. At the same time, it is often desired to maintain significant difference across groups so that the data can be clearly separated. Repulsive regularization among cluster centers serve this purpose, however, they suffer from the curse of dimensionality and specifying the repulsion parameter inevitably leads to sensitivity issue. In this article, we propose a different but simple regularization by assigning data completely into clusters without randomness, this creates automatic repulsion without need for tuning. This becomes especially useful in clustering with high dimensional data, where a separable linear subspace can be obtained. Simulations illustrate the strengths of the method and substantial gains are demonstrated in an application of clustering synaptomes.

KEY WORDS: Complete Membership, High Dimensional Clustering, Repulsive Regularization.

1 Introduction

Model based clustering (Fraley and Raftery, 2002) is used extensively in unsupervised learning. The common strategy is to treat the likelihood of each data y_i for $i = 1 \dots n$ as a weighted mixture of independent components $L(y_i) = \sum_{k=1}^{K} \pi_k f(y_i|\theta_k)$, where π_k is the weight and θ_k is the parameter for the kth component. The standard optimization procedure introduces an important latent variable z_i, that assigns each data to a component as a one-out-of-K random draw, coupling with an expectation-maximization (EM) (cite Dempster) algorithm to estimate π_k and θ_k. After the algorithm converges, one assigns the data to the most probable choice for z_i, dividing the data to K partitions. For

multivariate Gaussian data $y_i \in \mathbb{R}^p$, the covariance is quite useful to accommodate the different importances in each sub-dimension. For example, large variance on the diagonal could result in significant overlap of components, suggesting the sub-dimension is less importance than the others.

The above method fails in high dimensional data with $p \gg n$. Due to the rank, the $p \times p$ covariance matrix cannot be estimated; even with a p-element diagonal matrix, there is still large uncertainty due to the small n, leading to poor performance. To solve this problem, it is useful to consider dimension reduction by decomposing the matrix $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{V} + \boldsymbol{U}$ with $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ and $d \ll p$ and then use model-based clustering on $\boldsymbol{X}$. Various matrix factorization methods have been used to obtain the lower dimensional factor $\boldsymbol{X}$, such as principle component analysis (PCA) (Liu et al., 2003) and non-negative matrix factorization (Tamayo et al., 2007). A significant drawback, however, is that the top d learned subspaces in $\boldsymbol{X}$ do not guarantee a good separation. For example, PCA generates subspace that maximizes the total variance, but good separation in clustering is related to large between-group variance.

As a remedy, it is possible to re-adjust the orientation of $\boldsymbol{V}$ after clustering is done on $\boldsymbol{X}$. For example, when $\boldsymbol{X}$ is clustered, its mean can be expressed as a product of the latent variable probability $\boldsymbol{W} \in \mathbb{R}^{n \times k}$ over k components, and their corresponding d-dimensional centers $\boldsymbol{\mu} \in \mathbb{R}^{k \times d}$. Then using $\boldsymbol{W}\boldsymbol{\mu}$ to replace $\boldsymbol{X}$, one can update the estimate of $\boldsymbol{V}$. Alternating between matrix factorization and clustering aligns the subspace to a certain direction that optimizes the clustering model. However, this re-adjustment alone does not solve the low-separation issue. Indeed, model-based clustering only characterizes the degree of overlap through a mixture framework, but does not enforce good separation among the cluster centers.

Therefore, it is useful to consider regularization to obtain good separation in the reduced dimensional subspace. In this regard, repulsive regularization is useful. Examples include the determinantal point process (Kulesza and Taskar, 2012) and repulsive mixture (Petralia et al., 2012). These models show good performance in the original data space. When it comes to the latent low dimensional space, there are several critical issues: the amount of the repulsion is controlled by the hyper-parameter, which is difficult to tune when the outcome is not directly observable, creating sensitivity issue; the computation is costly due to the evaluation of the determinant or pairwise repulsion.

Motivated by these studies, we propose a new repulsive regularization on the component centers

in the low dimensional subspace. Instead of directly applying penalty on a short distance, we modify the latent probability matrix $\boldsymbol{W}$ to a complete membership binary matrix $\hat{\boldsymbol{Z}}$, which is learned when maximizing the conditional likelihood. This indirectly creates repulsion among the centers. Then alternating maximization can be utilized to find the subspace where the data are separable. This model is efficient to estimate and requires no tuning, hence we refer it as automatic repulsive clustering.

The article proceeds as follows: in section 2, the modeling framework and the estimation procedure are described; in section 3, theory is provided on the automatic repulsion; in section 4, simulation illustrates the advantages; in section 5, a real data application is demonstrated via synapse clustering.

2 Automatic Repulsive Clustering in Reduced Dimension

2.1 Clustering Under Reduced Dimension

We first summarize the framework that combines dimension reduction and clustering. We refer this as reduced dimension clustering.

Let $\boldsymbol{Y} \in \mathbb{R}^{n\times p}$ be the observed data, then we assume the clustering signal reside in a low dimensional matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ with $d \ll p$. The clustering of $\boldsymbol{X}$ can be represented as the assignment probability matrix $\boldsymbol{W} \in \mathbb{R}^{n\times k}$ with respect to k components $w_{i,k} = \frac{\pi_k f(x_i|\theta_k)}{\sum_k \pi_k f(x_i|\theta_k)}$,and the cluster mean $\boldsymbol{\mu} \in \mathbb{R}^{k\times d}$ adding a random residual $\boldsymbol{E} \in \mathbb{R}^{n\times d}$. Using matrix form:

$$\begin{aligned}\boldsymbol{Y} &= \boldsymbol{X}\boldsymbol{V} + \boldsymbol{U} \\ &= (\boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{E})\boldsymbol{V} + \boldsymbol{U}\end{aligned} \tag{1}$$

where $\boldsymbol{U} \in \mathbb{R}^{n\times p}$ is a matrix containing noise $U_{ij} \sim N(0, \sigma^2)$ and $\boldsymbol{E} \in \mathbb{R}^{n\times d}$ is the Gaussian noise with each row $\boldsymbol{E}'_{i,\cdot} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$. Common choice for $\boldsymbol{\Sigma}$ includes dense $d \times d$ matrix or simple diagonal matrix.

As a comparison, consider direct clustering on the original space $\mathbb{R}^{n\times p}$:

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{U}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{k\times p}$ and $\boldsymbol{U} \in \mathbb{R}^{n\times p}$.

The key difference lies in the error structure, in direct clustering, each row $\boldsymbol{U}'_{i,\cdot} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}^*)$ with $\boldsymbol{\Sigma}^*$ as a $p \times p$ matrix, due the large dimension p, it is difficult to impose structure in or estimate $\boldsymbol{\Sigma}^*$. In reduced dimension, since the subspace projection $\boldsymbol{V}$ is learned, each row of the error term $\boldsymbol{E}'_{i,\cdot}\boldsymbol{V} + \boldsymbol{U}'_{i,\cdot} \sim N(\boldsymbol{0}, \boldsymbol{V'\Sigma V} + \sigma^2\boldsymbol{I})$, which is a projection of low rank d matrix to the large p matrix. This low rank structure allows borrowing of strength across p different dimensions and is especially useful when the sample size n is small.

2.2 Automatic Repulsive Clustering (ARC)

We now introduce the automatic repulsive regularization. Typically the regularization is applied on $\boldsymbol{\mu}$ directly, causing sensitivity issue with tuning parameter and computing inconvenience. Instead, we regularize by replacing $\boldsymbol{W}$ with $\hat{\boldsymbol{Z}}$, which is the most probable choice of component for each data under $\hat{z}_{i,k} = 1\left(k = \underset{k}{\text{argmax}}\pi_k f(x_i|\theta_k)\right)$. As each μ_k is the average of x_i weighted by $w_{i,k}$, this leads to automatic repulsion among them, stated by the following theorem:

Theorem 1. *For any* $k \neq k^*$, $||\frac{\sum_i w_{i,k}x_i}{\sum_i w_{i,k}} - \frac{\sum_i w_{i,k^*}x_i}{\sum_i w_{i,k^*}}|| \leq ||\frac{\sum_i z_{i,k}x_i}{\sum_i z_{i,k}} - \frac{\sum_i z_{i,k^*}x_i}{\sum_i z_{i,k^*}}||$

The proof of this theorem is provided in the appendix.

The interpretation of this regularization is to force the estimate centers $\boldsymbol{\mu}_k$'s to be far apart, so that each row $\boldsymbol{X}_i$ has one of assignment probability $w_{i,k} \approx 1$. Compared to the other regularization (Kulesza and Taskar, 2012), this is much simple and tuning free. As we show in the next section, the estimation can be carried out by replacing the expectation step in model-based clustering with a maximization step.

3 Estimation

We divide the estimation into two parts: estimate the subspace by updating matrix $\boldsymbol{X}$, give the clustering; use ARC to cluster $\boldsymbol{X}$. The estimation proceeds by alternating between these two steps:

3.1 Updating the Low Dimensional Subspace

Given the clustering matrix $\boldsymbol{Z}$ and $\boldsymbol{\mu}$, the log-likelihood function is:

$$\log L = -\frac{1}{2}\left\{\sum_{i}^{n}||\boldsymbol{Y}_{i,\cdot} - \boldsymbol{X}_{i,\cdot}\boldsymbol{V}||^2/\sigma^2 + \sum_{i}^{n}(\boldsymbol{X}_{i,\cdot} - \boldsymbol{Z}_{i,\cdot}\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{X}_{i,\cdot} - \boldsymbol{Z}_{i,\cdot}\boldsymbol{\mu}) + n\log\det\boldsymbol{\Sigma} + np\log\sigma^2\right\} + C$$

To ensure identifiability, we regularize $V_{i,j} \sim N(0,\nu)$, where ν is a large variance (e.g. 10^6). It is possible to maximize the log-likelihood alternatively over $\boldsymbol{V}$ and $\boldsymbol{X}$, however, this would underestimate the variability of the $\boldsymbol{V}$, leading to suboptimal result. Instead, we treat $\boldsymbol{V}$ as latent variable and use EM algorithm for optimization.

Note the conditional distribution for $\boldsymbol{V}$ is:

$$\boldsymbol{V}_{\cdot,j} \overset{indep}{\sim} N\left((\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\nu)^{-1}\boldsymbol{X}'\boldsymbol{Y}_{\cdot,j}, \sigma^2(\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\nu)^{-1}\right)$$

for $j = 1 \dots p$.

This leads to computing the expectation:

$$\mathbb{E}\boldsymbol{V} = (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\nu)^{-1}\boldsymbol{X}'\boldsymbol{Y}, \qquad \mathbb{E}\boldsymbol{V}\boldsymbol{V}' = p\sigma^2(\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\nu)^{-1} + (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\nu)^{-1}\boldsymbol{X}'\boldsymbol{Y}\boldsymbol{Y}'\boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\nu)^{-1}$$

Then maximize over $\boldsymbol{X}$:

$$\hat{\boldsymbol{X}} = \{\boldsymbol{Y}\mathbb{E}\boldsymbol{V}'/\sigma^2 + \boldsymbol{Z}\boldsymbol{\mu}\boldsymbol{\Sigma}^{-1}\}(\mathbb{E}\boldsymbol{V}\boldsymbol{V}'/\sigma^2 + \boldsymbol{\Sigma}^{-1})^{-1}$$

and over σ^2:

$$\hat{\sigma}^2 = \{\mathrm{vec}(\boldsymbol{Y})'\mathrm{vec}(\boldsymbol{Y}) - 2\mathrm{vec}(\hat{\boldsymbol{X}}')'\mathrm{vec}(\mathbb{E}\boldsymbol{V}\boldsymbol{Y}') + \mathrm{vec}(\hat{\boldsymbol{X}}')'\mathrm{vec}(\mathbb{E}\boldsymbol{V}\boldsymbol{V}'\hat{\boldsymbol{X}}')\}/np$$

where vec(.) denotes the column-wise vectorization.

As the loss function, the expected log-likelihood is:

$$\mathbb{E}\log L = -\frac{1}{2}[\{\mathrm{vec}(\boldsymbol{Y})'\mathrm{vec}(\boldsymbol{Y}) - 2\mathrm{vec}(\hat{\boldsymbol{X}}')'\mathrm{vec}(\mathbb{E}\boldsymbol{V}\boldsymbol{Y}') + \mathrm{vec}(\hat{\boldsymbol{X}}')'\mathrm{vec}(\mathbb{E}\boldsymbol{V}\boldsymbol{V}'\hat{\boldsymbol{X}}')\}/\sigma^2 +$$

$$\sum_{i}^{n}(\boldsymbol{X}_{i,\cdot} - \boldsymbol{Z}_{i,\cdot}\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{X}_{i,\cdot} - \boldsymbol{Z}_{i,\cdot}\boldsymbol{\mu}) + n\log\det\boldsymbol{\Sigma} + np\log\sigma^2]$$

3.2 Clustering

The clustering can be carried out by alternative maximization over $\hat{Z}$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}$.

$$\hat{z}_{i,k} = 1\left(k = \underset{k}{\text{argmax}} \pi_k f(x_i|\theta_k)\right)$$

$$\boldsymbol{\mu}_k = \frac{\sum_i z_{i,k} \boldsymbol{X}_i}{\sum_i z_{i,k}}$$

$$\boldsymbol{\Sigma} = \frac{\sum_k \sum_i z_{i,k} \boldsymbol{X}_i \boldsymbol{X}_i'}{\sum_k \sum_i z_{i,k}}$$

Then the whole estimation can be carried out by alternating in the two steps. To summarize, the estimating algorithm is shown in Algorithm 1.

initialization;
while *Change in expected likelihood* $\Delta \mathbb{E} \log L >$ *threshold* **do**
  Using $\hat{\boldsymbol{Z}}\hat{\boldsymbol{\mu}}$, compute expectation: $\mathbb{E}\boldsymbol{V}$ and $\mathbb{E}\boldsymbol{V}\boldsymbol{V}'$;
  Using expected values, maximize: $\hat{\boldsymbol{X}}$;
  while *change in estimated* $\Delta\hat{\boldsymbol{\mu}} >$ *threshold* **do**
    Compute MLEs: $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$;
    Compute Most Probable Assignment: $\hat{\boldsymbol{Z}}$;
  end
  Compute expected likelihood $\mathbb{E} \log L$;
end

Algorithm 1: Estimation Algorithm for Automatic Repulsive Clustering

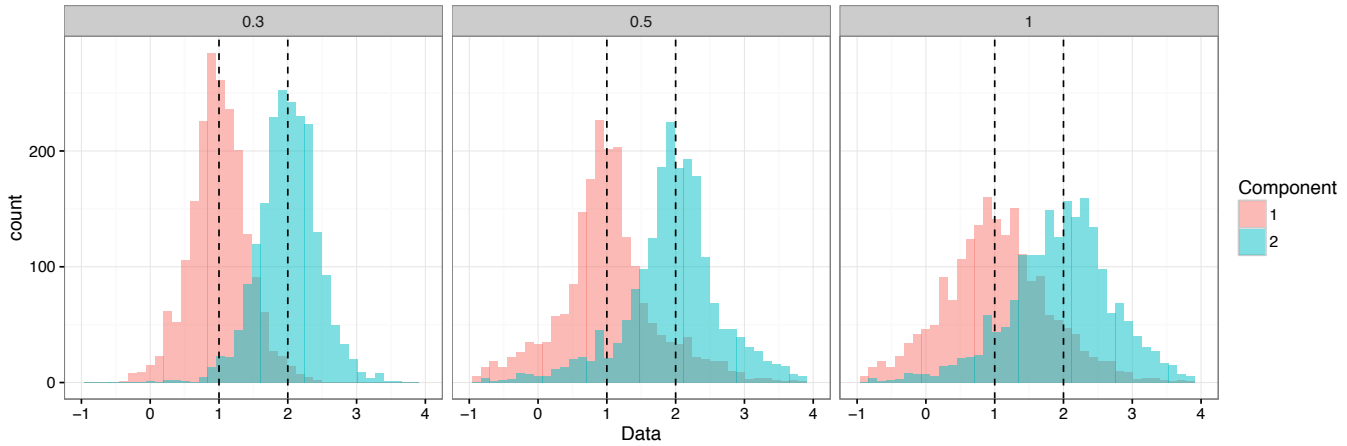
4 Experiments

4.1 Automatic Repulsion

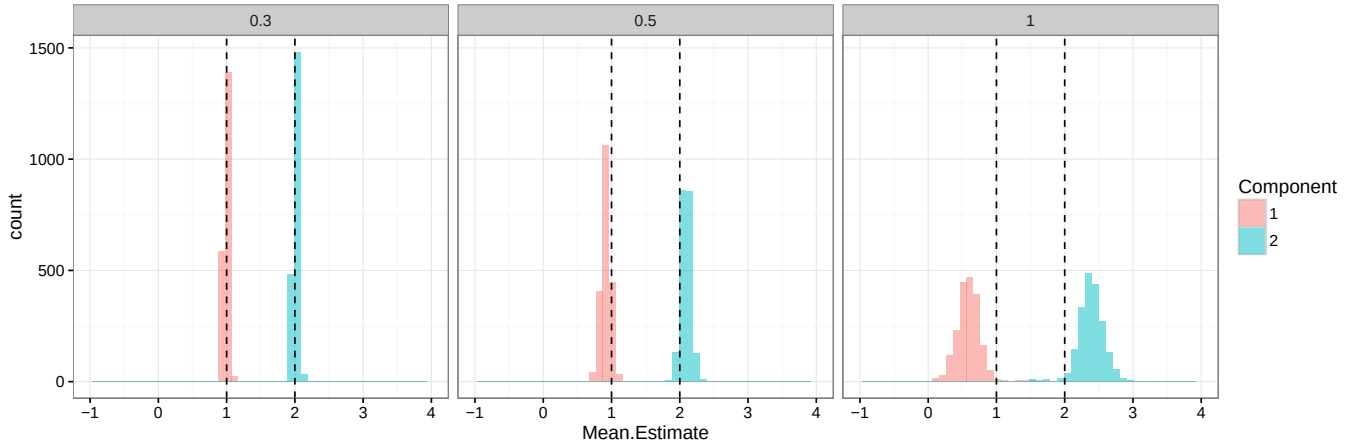
Before we carry out clustering simulations, we first illustrate the effects of automatic repulsion. We generate a mixture of two Gaussian components with mean 1 and 2. We then change the variance in 0.3^2, 0.5^2 and 1^2, so that the data exhibit different degree of overlap (Figure 1a). We then use ARC to estimate the two component means. We repeat the experiment for $1,000$ times so that the distribution of the mean estimates can be visualized (Figure 1b).

When the data has little overlap (with variance 0.3^2), the mean estimate for the two components

show almost no repulsion. As the variance increases (with variance 0.5^2), the mean estimates start to be pushed apart. When the overlapping becomes severe (with variance 1^2), the two mean estimates are further so that they still remain well separated. This behavior ensures the clustering generates clear boundary between clusters. The regularization is automatic and requires no tuning.



(a) The distribution of the data.



(b) The distribution of the mean under the automatic repulsion.

Figure 1: Simulation illustrating the automatic repulsion in the mean estimates, under different degree of overlap (standard deviation) in the generated data.

4.2 Clustering Synthetic Data

To illustrate the effect of clustering in low dimension, we simulate 3 component data with $n_1 = 50$, $n_2 = 150$ and $n_3 = 100$ in two dimensions, with means $(1, 2)$, $(2, 1)$ and $(3, 3)$. We set the variance to be 0.2 so that the components are well separated.

Then we project the matrix into p dimensions via a random matrix V with $V_{ij} \sim N(0, 1)$. And

we test three models (i) the Gaussian mixture model (GMM) directly on the the high dimension, (ii) GMM on the reduced dimension learned by PCA and (iii) ARC on adapted reduced dimension. We obtain the clustering labels estimated from the models and compute the adjusted random index with respect to the true labels. Each experiment is repeated 10 times with mean and standard deviation reported.

We first simply vary p from 50 to $1,000$ to test the performance of models. Since each dimension contains signals, not surprisingly, the simple GMM on the large dimensions performs the best among all three models (Table 1, row 1), followed closely by the ARC model. However, such setting is very unrealistic as often most of the large dimensions are irrelevant. To simulate this setting, we keep the projected dimensions fixed at 20 but add $(p-20)$ interfering dimensions from pure noise $y_{i,j} \sim N(0,1)$. The performance of the high dimensional GMM starts to deteriorate when the irrelevant dimensions to start to interfere with the signals (Table 1, row 2).

Without irrelevant dimensions

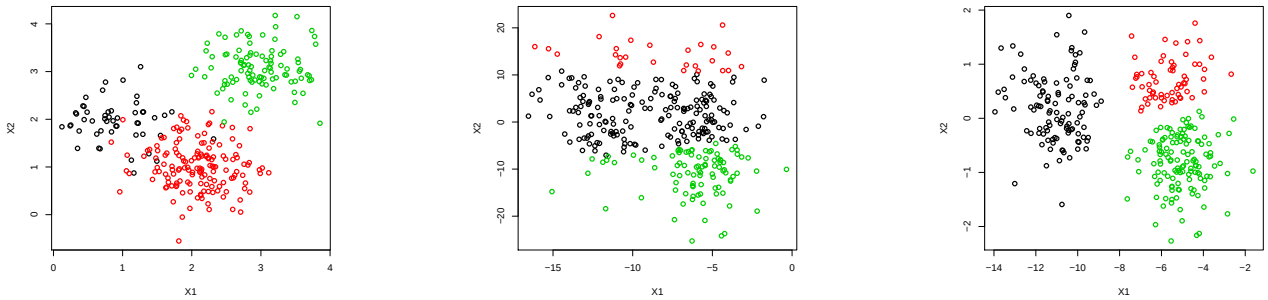
p	50	100	200	500	1000
GMM	**0.83** $\pm$ 0.04	**0.84** $\pm$ 0.04	**0.84** $\pm$ 0.03	**0.87** $\pm$ 0.03	**0.85** $\pm$ 0.05
PCA+GMM	0.64 $\pm$ 0.20	0.77 $\pm$ 0.19	0.66 $\pm$ 0.29	0.79 $\pm$ 0.22	0.69 $\pm$ 0.30
ARC	0.72 $\pm$ 0.12	0.82 $\pm$ 0.16	0.79 $\pm$ 0.14	0.73 $\pm$ 0.19	0.86 $\pm$ 0.13

With $(p-20)$ irrelevant dimensions

p	50	100	200	500	1000
GMM	0.76 $\pm$ 0.03	0.69 $\pm$ 0.13	0.70 $\pm$ 0.22	0.63 $\pm$ 0.11	0.53 $\pm$ 0.07
PCA+GMM	0.75 $\pm$ 0.23	0.48 $\pm$ 0.25	0.63 $\pm$ 0.21	0.48 $\pm$ 0.24	0.57 $\pm$ 0.21
ARC	**0.80** $\pm$ 0.11	**0.75** $\pm$ 0.11	**0.83** $\pm$ 0.10	**0.79** $\pm$ 0.09	**0.76** $\pm$ 0.09

Table 1: Clustering performance using adjusted random index with respect to the true labels.

In this regard, the reduced dimensional method is advantageous as it serves to discard the irrelevant dimensions. Nevertheless, clustering based on PCA-generated subspace almost never perform well. This can be explained by Figure 2b that the first two principle components show little separation. This problem is solved with the repulsive regularization. As shown in Figure 2c, the discovered low dimension space demonstrates clear and wide boundaries among clusters, leading to superior performance in ARI.



(a) The true low dimensions in data generation.

(b) The lower dimensions discovered by PCA.

(c) The lower dimensions discovered by ARC.

Figure 2: Scatterplots illustrating the learning of the low dimensional subspace using PCA and ARC. The ground truth, clustered labels under PCA+GMM and clustered labels under ARC are shown in colors.

4.3 Clustering Real Data: Handwritten Digits Clustering

We now test the model on real data with MNIST handwritten digits. The digits are centered in a 28x28 image. We normalize the pixel values by dividing 255.

Following the choice of digits (Lee and Choi, 2015) for clustering benchmark, we use the images of four digits: $0, 3, 7, 9$. To mimic the low sample size and high dimension setting, we randomly draw 50 samples for each digit. Each experiment is repeated 10 times for GMM applied on the data, PCA based GMM and ARC. The results are shown in Table **??**. The ARC clearly shows clear advantages in clustering the four digits over the other methods.

For completeness, we also test the clustering models with all 10 digits. Due to the similarity of the digits and noise in rotation, we observe a significant decrease in ARI. Nevertheless, ARC still dominates in the performance over the other methods.

As comparison, we include the test result using sparsity regularized clustering (SparCL,Witten and Tibshirani (2012)). Due to the nature of image data, the sparsity assumption is not quite suitable for this data, except for removing common blank background. As the result, its performance is almost the same as the simple GMM directly applied on the data.

Digits	$(0, 3, 7, 9)$	$0 - 9$
GMM	0.37 ± 0.04	0.10 ± 0.11
PCA+GMM	0.35 ± 0.14	0.24 ± 0.01
SparCL	0.36 ± 0.12	0.11 ± 0.13
ARC	$\mathbf{0.44} \pm 0.12$	$\mathbf{0.30} \pm 0.05$

Table 2: Adjusted rand index as benchmark for the different clustering algorithms applied on the MNIST handwritten digits data.

5 Discussion

With the large dimensions contaminated with irrelevant information, clustering with high dimensional data is challenging. Applying dimension reduction is an appealing direction and has recently gain more traction. Examples include Niu et al. (2011) combining dimension reduction with spectral clustering to separate data into multiple manifolds and Jing et al. (2013) using dictionary-learning structure to find the low-dimension subspace.

Since one faces the task of choosing top few dimensions, one critical issue remained, it to identify a subspace that it is "optimal" in some sense. In this article, we restrict the focus on the linear subspace, and seek the one where the clustered components have a clear separating boundary. This is a particular useful complement to model-based clustering, where the separation is originally missing in mixture density.

With the tuning-free regularization, it is simple to achieve this task in our model. As demonstrated in the examples, it achieves quite satisfactory performance in both simulation and real data applications. One extension to the current work would be studying the robustness of the error structure. Another one is to combine the regularization with kernel based spectral clustering hence extends the method to non-linear space.

References

Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

Liping Jing, Michael K Ng, and Tieyong Zeng. Dictionary learning-based subspace structure identi-

fication in spectral clustering. *IEEE transactions on neural networks and learning systems*, 24(8): 1188–1199, 2013.

Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., Hanover, MA, USA, 2012. ISBN 1601986289, 9781601986283.

Juho Lee and Seungjin Choi. Bayesian hierarchical clustering with exponential family: Small-variance asymptotics and reducibility. In *AISTATS*, 2015.

Jun S Liu, Junni L Zhang, Michael J Palumbo, and Charles E Lawrence. Bayesian clustering with variable and transformation selections. In *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, page 249. Oxford University Press, USA, 2003.

Donglin Niu, Jennifer G Dy, and Michael I Jordan. Dimensionality reduction for spectral clustering. In *AISTATS*, pages 552–560, 2011.

Francesca Petralia, Vinayak Rao, and David B Dunson. Repulsive mixtures. In *Advances in Neural Information Processing Systems*, pages 1889–1897, 2012.

Pablo Tamayo, Daniel Scanfeld, Benjamin L Ebert, Michael A Gillette, Charles WM Roberts, and Jill P Mesirov. Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences*, 104(14):5959–5964, 2007.

Daniela M Witten and Robert Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 2012.

6 Appendix

6.1 Proof of Theorem

Consider the finite mixture model with the center estimates as $\mu_k = \frac{\sum_i \mathbb{E} z_{k,i} y_i}{\sum_i \mathbb{E} z_{k,i}}$, and complete membership model with the center estimates as $\mu_k^* = \frac{\sum_i z_{k,i} y_i}{\sum_i z_{k,i}}$. We are interested in comparing the pairwise distance among the centers from the two models.

Let the pairwise distance be $||\mu_1 - \mu_2||$ between two centers in the finite mixture. As the $||\mu_1 - \mu_2|| = \sqrt{\sum_{l=1}^{p} ||\mu_{1,l} - \mu_{2,l}||^2}$, we focus on one sub-dimension $\mu_{1,l} - \mu_{2,l}$, without loss of generality, we assume $\mu_{1,l} > \mu_{2,l}$.

For any $y_{j,l} \geq \frac{\sum_{i \neq j} \mathbb{E}z_{1,i} y_{i,l}}{\sum_{i \neq j} Ez_{1,i}} \geq \frac{\sum_{i \neq j} \mathbb{E}z_{2,i} y_{i,l}}{\sum_{i \neq j} Ez_{2,i}}$ and $\mathbb{E}z_{1,j} \geq \mathbb{E}z_{2,j}$,

$$\begin{aligned}
\mu_{1,l} - \mu_{2,l} =& \frac{\sum_{i \neq j} \mathbb{E}z_{1,i} y_{i,l} + \mathbb{E}z_{1,j} y_{j,l}}{\sum_{i \neq j} \mathbb{E}z_{1,i} + \mathbb{E}z_{1,j}} - \frac{\sum_{i \neq j} \mathbb{E}z_{2,i} y_{i,l} + \mathbb{E}z_{2,j} y_{j,l}}{\sum_{i \neq j} \mathbb{E}z_{2,i} + \mathbb{E}z_{2,j}} \\
\leq& \frac{\sum_{i \neq j} \mathbb{E}z_{1,i} y_{i,l} + y_{j,l}}{\sum_{i \neq j} \mathbb{E}z_{1,i} + 1} - \frac{\sum_{i \neq j} \mathbb{E}z_{2,i} y_{i,l}}{\sum_{i \neq j} \mathbb{E}z_{2,i}}
\end{aligned}$$

For any $y_{j,l} \leq \frac{\sum_{i \neq j} \mathbb{E}z_{2,i} y_{i,l}}{\sum_{i \neq j} Ez_{2,i}} \leq \frac{\sum_{i \neq j} \mathbb{E}z_{1,i} y_{i,l}}{\sum_{i \neq j} Ez_{1,i}}$ and $\mathbb{E}z_{1,j} \leq \mathbb{E}z_{2,j}$,

$$\mu_{1,l} - \mu_{2,l} \leq \frac{\sum_{i \neq j} \mathbb{E}z_{1,i} y_{i,l}}{\sum_{i \neq j} \mathbb{E}z_{1,i}} - \frac{\sum_{i \neq j} \mathbb{E}z_{2,i} y_{i,l} + y_{j,l}}{\sum_{i \neq j} \mathbb{E}z_{2,i} + 1}$$

By induction, this converts all the $Ez_{k,i}$ to $z_{k,i}$ hence $\mu_{1,l} - \mu_{2,l} \leq \mu^*_{1,l} - \mu^*_{2,l}$.