Estimating the number of clusters using Cross Validation

Wei Fu [*]
Department of IOMS, New York University
and
Patrick O. Perry
Department of IOMS, New York University

October 12, 2015

Abstract

Many clustering methods, including k-means, require the user to specify the number of clusters as an input parameter. A variety of methods have been devised to choose the number of clusters automatically, but they often rely on strong modeling assumptions. We propose a data-driven approach to estimate the number of clusters based on a novel form of cross-validation. This differs from ordinary cross-validation, because clustering is fundamentally an unsupervised learning problem. Simulation and real data analysis results show that our proposed method outperforms existing methods, especially in high-dimensional settings with heavy-tailed data.

Keywords: clustering, unsupervised learning, data-driven method

[*] The authors gratefully acknowledge

1 Introduction

As a main task of exploratory data analysis, clustering organizes unlabeled observations into groups such that observations in same group are more similar compare to those in different group. Clustering is an important topic in unsupervised learning because it can reveal the internal structure of data through grouping, segment the data through partitioning and summarize data for other purposes such as dimension reduction. It has being widely used in various fields such as psychology, biology, statistics and machine learning including pattern recognition, image segmentation etc (Jain et al., 1999).

After being proposed more than 50 years, k-means remains one of the most popular and widely used clustering algorithms (Jain, 2010). Like many other clustering methods, k-means requires an input parameter k, the number of clusters, to be specified by the user. Automatically and quantitatively deciding such parameter is important and yet unsolved problem (Fujita et al., 2014). Various methods have been proposed to tackle this difficulty. One ad hoc approach is to explore the relationship between W_k (within-cluster dispersion) and the number of cluster k for a certain clustering method such as k-means. Since W_k decreases as k increases, one usually find the "elbow" of curve obtain by plotting W_k versus k as the appropriate number of clusters. The example on the top row of Figure 1 demonstrates such approach for data with $k = 4$, where the "elbow" point indeed reveals the true number of clusters. This is based on the idea that under partitioning data set has more impact than over partitioning data set in terms of W_k. However, locating the "elbow" point is somewhat subjective and sometimes is not appropriate to select the optimal k. The second example on the bottom row of Figure 1 shows a situation where there is no clear choice of the "elbow" point – both $k = 2$ and $k = 3$ can be viewed as the "elbow" point. What's more, the true $k = 4$ can never be selected as the optimal k using such approach in this case since it can hardly be viewed as the "elbow" of the curve.

Recently, there are several new proposals to find the k automatically. Gap statistics (Tibshirani et al., 2001) estimates k by comparing the change in within-cluster dispersion with that expected under an appropriate reference null distribution. Specifically, the graph of $\log(W_k)$ is compared with its expectation under an appropriate null reference distribution of the data. The value of k associated with the largest gap between $\log(W_k)$ and the

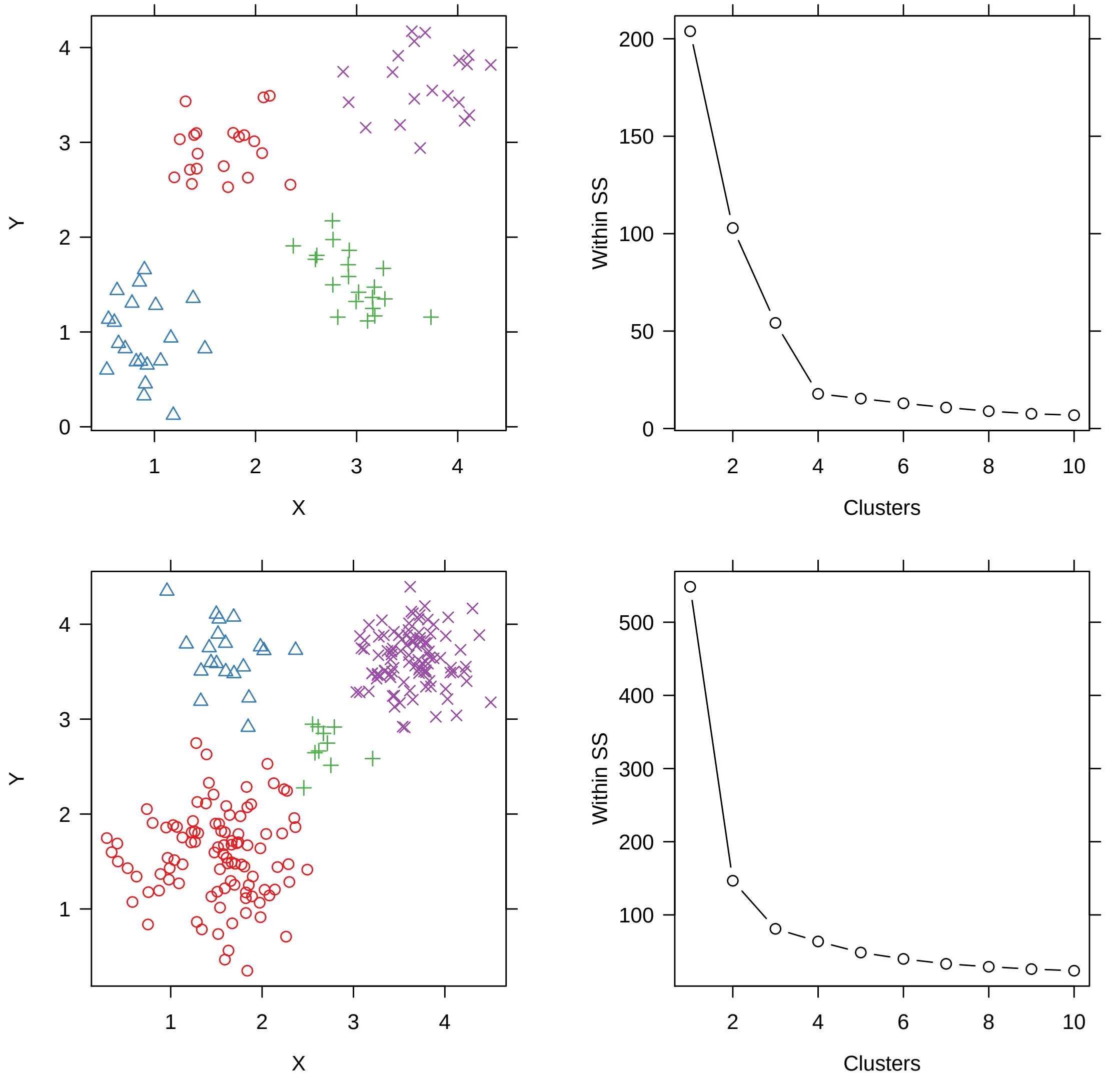

Figure 1: Left panels show the (X, Y) data points; right panels show the corresponding values of the within-cluster sum of squares W_k plotted against the number of clusters, k.

reference curve is selected as optimal k. Sugar and James (2003) proposed an approach which finds the number of clusters based on distortion, a quantity that measures the average distance, per dimension. It's backed by a rigorous theoretical justification based on information-theoretic ideas. Fraley and Raftery (2002)'s Model-based method employs the EM algorithm to estimate the parameters in Gaussian mixture model, and select the best model (k) using BIC criterion. Stability-based criterion is also proposed to locate the best k by some authors such as Ben-Hur et al. (2001), Wang (2010) and Fang and Wang (2012). Chiang and Mirkin (2010) provides a nice review of existing methods for finding the right k in published literature.

Most existing methods are either model based method requires strong modeling assumptions or otherwise lack of clear interpretation and theoretical justification. Although many view selecting the number of clusters as a model selection problem, very few approaches this problem from the prediction point of view. Select model with smallest prediction error via cross-validation is one of the simplest and most widely used model selection techniques in supervised learning. The lack of true class (label) in data set makes the adoption of cross-validation into unsupervised leaning problem difficult.

One exception is Tibshirani and Walther (2005), which selects the optimal k by prediction strength. The strategy is to first cluster the test data and training data into k clusters respectively. Then, for each pair of observations that are assigned to the same test cluster, algorithm determines whether they are also assigned to the same cluster based on the training centers. The intuition here is, if $k = k_0$, the true number of clusters, then the k training set clusters will be similar to the k test clusters, and hence will predict them well. However, a specifically defined prediction error measure is used in such procedure, which is quite different from the one commonly used in cross-validation procedure in supervised learning. Wang (2010) also uses cross-validation to select the optimal k. Instead of selecting k which minimizes prediction error, such method picks k which minimizes the specifically defined clustering instability. Note that these methods minimize measures that are specifically defined, which makes these methods hard to interpret or compare to other methods through analogy because the underlying measures are unique and not well understood.

The prediction error measure in our proposed method is exactly the same as in supervised learning. Our method is a complete data-driven approach which doesn't rely on any model assumption. Through novel form of partitioning data set, we effectively transferred an unsupervised learning problem into a supervised leaning problem, which is one of the kind. By doing so, we are able to employing the cross-validation procedure in clustering a similar way as in supervised learning problem, so that much of the intuition from supervised learning carries over. Hence, it's easy for reader to understand the intuition behind our proposed method. Simulation and real data application shows the superior performance of our proposed method compare with existing methods in high-dimension settings and heavy-tailed data. Since the embedded cross-validation procedure is well understood, it also makes our method potentially easily to be extended in future study.

2 Cross-validation for selecting the number of clusters

Cross-validation is commonly used for model selection in supervised learning problems. In these settings, the data comes in the form of N predictor-response pairs, $(X_1, Y_1), \ldots, (X_N, Y_N)$, with $X_i \in \mathbb{R}^p$ and $Y_i \in \mathbb{R}^q$. The data can be represented as a matrix with N rows and $p+q$ columns. We partition the data into K hold-out "test" subsets, with K typically chosen to be 5 or 10. For each "fold" r in the range $1, \ldots, K$, we permute the rows of the data matrix to get $\mathfrak{X}$, a matrix with the rth test subset as its trailing rows. We partition $\mathfrak{X}$ as

$$\mathfrak{X} = \begin{bmatrix} X_{\text{train}} & Y_{\text{train}} \\ X_{\text{test}} & Y_{\text{test}} \end{bmatrix}.$$

We use the training rows $[X_{\text{train}} \; Y_{\text{train}}]$ to fit a regression model $\hat{Y} = \hat{Y}(X)$, and then evaluate the performance of this model on the test set, computing the cross-validation error $\|Y_{\text{test}} - \hat{Y}(X_{\text{test}})\|^2$ or some variant thereof. We choose the model with the smallest cross-validation error, averaged over all K folds.

In unsupervised learning problems like factor analysis and clustering, the features of the observations are not naturally partitioned into "predictors" and "responses", so we cannot directly apply the cross-validation procedure described above. For factor analysis, there are at least two versions of cross-validation. Wold (1978) proposed a "speckled" holdout, where in each fold we leave out a subset of the elements of the data matrix. Wold's procedure works well empirically, but does not have any theoretical support, and it requires a factor analysis procedure that can handle missing data. Owen and Perry (2009) proposed a scheme called "bi-cross-validation" wherein each fold designates a subset of the data matrix columns to be response and a subset of the rows to be test data. This generalized a procedure due to Gabriel (2002), who proposed holding out a single column and a single row at each fold. Owen and Perry proved that this procedure is self-consistent, in the sense that it performs the correct model selection in the absence of noise, and Perry (2009) provided more theoretical support.

In this report, we extend the Wold and Gabriel methods to the clustering problem, specifically to choose an appropriate number of clusters for a dataset. We prove that the Gabriel method is self-consistent, and we analyze some of its properties in the presence of noise. We compare these methods to state-of-the-art algorithms, and show that both are competitive.

We now give the details of how to implement the Gabriel cross-validation to locate the optimal cluster number k. The Wold cross-validation algorithm is described in Appendix A.

2.1 Gabriel CV algorithm

We are given a data matrix with N rows and P columns. In each fold of cross-validation, we permute the rows and columns of the data matrix and then partition the rows and columns as $N = n + m$ and $P = p + q$ for non-negative integers n, m, p, and q. We treat the first p columns as "predictors" and the last q columns as "responses"; similarly, we treat the first n rows as "training" and the last m rows as "test". In block form, the permuted data matrix is

$$\mathfrak{X} = \begin{bmatrix} X_{\text{train}} & Y_{\text{train}} \\ X_{\text{test}} & Y_{\text{test}} \end{bmatrix},$$

where $X_{\text{train}} \in \mathbb{R}^{n \times p}$, $Y_{\text{train}} \in \mathbb{R}^{n \times q}$, $X_{\text{test}} \in \mathbb{R}^{m \times p}$, and $Y_{\text{test}} \in \mathbb{R}^{m \times q}$.

Given such a partition of $\mathfrak{X}$, we perform four steps for each value of k, the number of clusters:

1. **Cluster:** Cluster $Y_1, \dots, Y_n$, the rows of Y_{train}, yielding the assignment rule $\hat{G}^Y : \mathbb{R}^q \to \{1, \dots, k\}$ and the cluster means $\bar{\mu}_1^Y, \dots, \bar{\mu}_k^Y$. Set $\hat{G}_i^Y = \hat{G}^Y(Y_i)$ to be the assigned cluster for row i.

2. **Classify:** Take $X_1, \dots, X_n$, the rows of X_{train} to be predictors, and take $\hat{G}_1^Y, \dots, \hat{G}_n^Y$ to be corresponding class labels. Use the pairs $\{(X_i, \hat{G}_i^Y)\}_{i=1}^n$ to train a classifier $\hat{G}^X : \mathbb{R}^p \to \{1, \dots, k\}$.

3. **Predict:** Apply the classifier to $X_{n+1}, \dots, X_{n+m}$, the rows of X_{test}, yielding predicted classes $\hat{G}_i^X = \hat{G}^X(X_i)$ for $i = n+1, \dots, n+m$. For each value of i in this range, compute predicted response $\hat{Y}_i = \bar{\mu}^Y(\hat{G}_i^X)$, where $\bar{\mu}^Y(g) = \bar{\mu}_g^Y$.

4. **Evaluate:** Compute the cross-validation error

$$\mathrm{CV}(k) = \frac{1}{m} \sum_{i=n+1}^{n+m} \|Y_i - \hat{Y}_i\|^2,$$

   where $Y_{n+1}, \dots, Y_{n+m}$ are the rows of Y_{test}.

In principle, we could use any clustering and classification methods in steps 1 and 2. In this report, we use k-means as the clustering algorithm. For the classification step, we compute the mean value of X for each class; we assign an observation to class g if that class has the closest mean (randomly breaking ties between classes). The classification step is equivalent to linear discriminant analysis with equal class priors and identity noise covariance matrix.

To choose the folds, we randomly partition the rows and columns into K and L subsets, respectively. Each fold is indexed by a pair (r, s) of integers, with $r \in \{1, \dots, K\}$ and $s \in \{1, \dots, L\}$. Fold (r, s) treats the rth row subset as “test”, and the sth column subset as “response”. We typically take $K = 5$ and $L = 2$. For the number of clusters, we select the value of k that minimizes the average of $\mathrm{CV}(k)$ over all $K \times L$ folds (choosing the smallest value of k in the event of a tie).

3 Self-Consistency of Gabriel CV method

This section gives the self-consistency proof of the proposed Gabriel method. Specifically, we will show that under appropriate conditions, in the absence of noise, the Gabriel cross-validation procedure finds the optimal number of clusters.

Because k-means algorithm is essential to the method, we review the procedure here. Given a set of observations $\{x_1, \ldots, x_n\}$, and a specified the number of clusters k, the goal of the k-means procedure is to find a set of k or cluster centers $A = \{a_1, \ldots, a_k\}$ minimizing the within cluster dispersion

$$W(A) = \sum_{i=1}^{n} \min_{a \in A} \|x_i - a\|^2.$$

This implicitly defines a cluster assignment rule

$$g(x) = \underset{g \in \{1, \ldots, k\}}{\arg\min} \|x - a_g\|^2,$$

with ties broken arbitrarily. We will assume that the k-means procedure finds an optimal solution, A, but we will not assume that this solution is unique.

It will suffice to analyze a single fold of the cross-validation procedure. As in in section 2.1 we assume that the P variables of the data set have been partitioned into p predictor variables represented in vector X and q response variables represented in vector Y. The N observations have been divided into two sets: n train observations and m test observations. The following theorem gives conditions for Gabriel CV to recover the true number of clusters in the absence of noise.

Theorem 1. *Let $\{(X_i, Y_i)\}_{i=1}^{n+m}$ be the data from a single fold of Gabriel cross-validation. For any k, let $CV(k)$ be the cross-validation error for this fold, computed as described in Section 2.1. We will assume that there are K true centers $\mu(1), \ldots, \mu(K)$, with the gth cluster center partitioned as $\mu(g) = \big(\mu^X(g), \mu^Y(g)\big)$ for $g = 1, \ldots, K$. Suppose that*

(i) Each observation i has a true cluster $G_i \in \{1, \ldots, K\}$. There is no noise, so that $X_i = \mu^X(G_i)$ and $Y_i = \mu^Y(G_i)$ for $i = 1, \ldots, n + m$.

(ii) The vectors $\mu^X(1), \dots, \mu^X(K)$ are all distinct.

(iii) The vectors $\mu^Y(1), \dots, \mu^Y(K)$ are all distinct.

(iv) The training set contains at least one member of each cluster: for all g in the range $1, \dots, K$, there exists at least one i in the range $1, \dots, n$ such that $G_i = g$.

(v) The test set contains at least one member of each cluster: for all g in the range $1, \dots, K$, there exists at least one i in the range $n+1, \dots, n+m$ such that $G_i = g$.

Then $CV(k) < CV(K)$ for $k < K$, and $CV(k) = CV(K)$ for $k > K$, so that Gabriel CV correctly chooses $k = K$.

This theorem is implied by the following two lemmas.

Lemma 1. *Suppose that the assumptions of Theorem 1 are in force. If $k < K$, then* $\mathrm{CV}(k) > 0$.

Proof. By definition,

$$\mathrm{CV}(k) = \sum_{i=n+1}^{n+m} \|Y_i - \bar{\mu}^Y(\hat{G}_i^X)\|^2,$$

where $\bar{\mu}^Y(g)$ is the center of cluster g returned from applying k-means to $Y_1, \dots, Y_n$. Assumptions (i) and (v), imply that as i ranges over the test set $n+1, \dots, n+m$, the response Y_i ranges over all distinct values in $\{\mu^Y(1), \dots, \mu^Y(K)\}$. Assumption (iii) implies that there are exactly K such distinct values. However, there are only k distinct values of $\bar{\mu}^Y(g)$. Thus, at least one summand $\|Y_i - \bar{\mu}^Y(\hat{G}_i^X)\|^2$ is nonzero. Therefore, $\mathrm{CV}(k) > 0$. □

Lemma 2. *Suppose that the assumptions of Theorem 1 are in force. If $k \geq K$, then* $\mathrm{CV}(k) = 0$.

Proof. From assumptions (i), (iii), and (iv), we know the cluster centers gotten from applying k-means to $Y_1, \dots, Y_n$ must include $\mu^Y(1), \dots, \mu^Y(K)$. Without loss of generality, suppose that $\bar{\mu}^Y(g) = \mu^Y(g)$ for $g = 1, \dots, K$. This implies that $\hat{G}_i^Y = G_i$ for $i = 1, \dots, n$. Thus, employing assumption (i) again, we get that $\bar{\mu}^X(g) = \mu^X(g)$ for $g = 1, \dots, K$.

Since assumption (ii) ensures that $\mu^X(1), \dots, \mu^X(K)$ are all distinct, we must have that $\hat{G}_i^X = G_i$ for all $i = 1, \dots, m+n$. In particular, this implies that $\bar{\mu}^Y(\hat{G}_i^X) = Y_i$ for $i = 1, \dots, m+n$, so that $\mathrm{CV}(k) = 0$. □

4 Analysis of Gabriel Cross-Validation with Gaussian Noise

4.1 Single Cluster

Now we we analyze the asymptotic performance of Gabriel Cross-Validation, in the case of Gaussian noise. Our main result is that with single-cluster Gaussian data, if the predictor and response columns of $\mathfrak{X}$ are weakly correlated or independent, then the method will correctly prefer $k = 1$ to $k = 2$ clusters. We first state the result in the case where $\mathfrak{X}$ has two columns, and later generalize this result to higher dimensions.

Proposition 1. *Suppose that $\{(X_i, Y_i)\}_{i=1}^{n+m}$ is data from a single fold of Gabriel cross-validation, where each (X, Y) pair in $\mathbb{R}^2$ is an independent draw from a mean-zero multivariate normal distribution with unit marginal variances and correlation ρ. In this case, the data are drawn from a single cluster; the true number of clusters is* 1*. If $|\rho| < 0.5$, then* $\mathrm{CV}(1) < \mathrm{CV}(2)$ *with probability tending to one as m and n increase.*

Proof. Given $(X_1, Y_1), \dots, (X_n, Y_n)$, we first apply k-means to $\{Y_i\}_{i=1}^{n}$. With $k = 1$, the single-cluster centroid will be equal to $\bar{Y}_n$, the sample mean of the $Y_1, \dots, Y_n$, approximately equal to $\mathrm{E}(Y) = 0$, with error of size $O_p(n^{-1/2})$. The cross-validation error will be

$$\mathrm{CV}(1) = \frac{1}{m} \sum_{i=n+1}^{n+m} \|Y_i - \bar{Y}_n\|^2 = 1 + O_p(m^{-1/2}) + O_p(n^{-1/2}).$$

Now we will consider the $k = 2$ case. If n is large enough, then Pollard (1981) showed that the centroids $\bar{\mu}_1^Y$ and $\bar{\mu}_2^Y$ will be close to $\mathrm{E}(Y \mid Y > 0) = \sqrt{2/\pi}$ and $\mathrm{E}(Y \mid Y < 0) = -\sqrt{2/\pi}$. We have used Lemma 3 (Appendix B) to compute the expectations. Further, Pollard (1982) showed that the errors will be of size $O_p(n^{-1/2})$.

If $\rho > 0$ and n is large enough, then classification rule learned from $\{(X_i, \hat{G}_i^Y)\}_{i=1}^{n}$ variables will be determined according to whether $X > 0$; if $\rho < 0$ then the decision is according to whether $X < 0$. More specifically, the decision boundary will be at $0 + O_p(n^{-1/2})$.

In the $\rho > 0$ case, the cross-validation error will be

$$\begin{aligned}\mathrm{CV}(2) &= \frac{1}{m} \sum_{i=n+1}^{n+m} \|(Y_i - \bar{\mu}_1^Y) 1\{\hat{G}_i^X = 1\}\|^2 + \|(Y_i - \bar{\mu}_2^Y) 1\{\hat{G}_i^X = 2\}\|^2 \\ &= \mathrm{E}[(Y - a)^2 1\{X > 0\}] + \mathrm{E}[(Y + a)^2 1\{X < 0\}] + O_p(m^{-1/2}) + O_p(n^{-1/2}),\end{aligned}$$

where $a = \sqrt{2/\pi}$. From the joint normality of X and Y, it follows that $Y \mid X$ is normal with mean ρX and variance $(1 - \rho^2)$, so that $\mathrm{E}[(Y - a)^2 \mid X] = (\rho X - a)^2 + (1 - \rho^2)$. Applying Lemma 3, we get that for large m and n, the Gabriel cross-validation error is close to $1 + a^2(1 - 2\rho)$.

In the $\rho < 0$ case, a similar calculation shows that CV(2) is close to $1 + a^2(1 + 2\rho)$. In particular, if $|\rho| < 0.5$, then with probability tending to 1 and m and n increase, the asymptotic cross-validation error for $k = 1$ will be smaller than for $k = 2$. □

We confirm this result with a simulation. We perform 10 replicates. In each replicate, we generate 20000 observations from a mean-zero bivariate normal distribution with unit marginal variances and correlation ρ. We perform a single 2×2 fold of Gabriel cross-validation and report the cross-validation mean squared error for the number of clusters k ranging from 1 to 5. Figure 2 shows the cross-validation errors for all 10 replicates. The simulation demonstrates that in the Gabriel cross-validation criterion chooses the correct answer $k = 1$ whenever $\rho < 0.5$; the criterion chooses $k = 2$ clusters whenever $|\rho| > 0.5$.

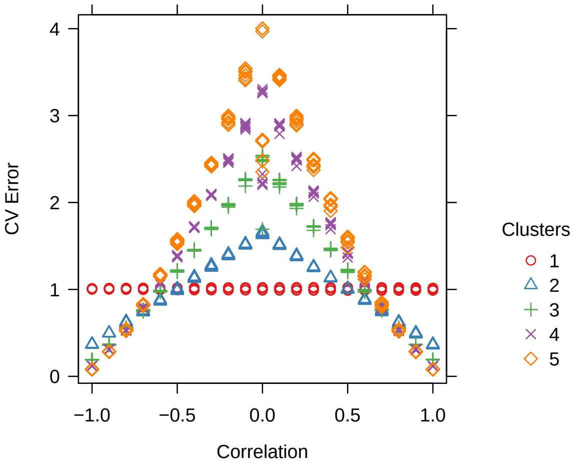

Figure 2: Cross-validation error on 10 replicates, with the number of clusters k ranging from 1 to 5. Data is generated from two-dimensional multivariate normal distribution with correlation ρ. The Gabriel cross-validation criterion chooses the correct answer $k = 1$ whenever $\rho < 0.5$; the criterion chooses $k = 2$ clusters whenever $|\rho| > 0.5$.

Proposition 1 gives a very simple condition for the Gabriel method to correctly pick $k = 1$ with single cluster in 2 dimensions. The following proposition generalizes such condition for data in arbitrary dimension.

Proposition 2. *Suppose that $\{(X_i, Y_i)\}_{i=1}^{n+m}$ is data from a single fold of Gabriel cross-validation, where each (X, Y) pair in $\mathbb{R}^{p+q}$ is an independent draw from a mean-zero multivariate normal distribution with covariance matrix $\Sigma_{XY} = \left(\begin{smallmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{smallmatrix}\right)$, with Σ_{YY} has leading eigenvalue λ_1 and corresponding eigenvector u_1. In this case, the data are drawn from a single cluster; the true number of clusters is 1. If $\frac{\sqrt{\lambda_1}}{2} > \frac{u_1^T \Sigma_{YX} \Sigma_{XY} u_1}{\sqrt{u_1^T \Sigma_{YX} \Sigma_{XX} \Sigma_{XY} u_1}}$, then* $\mathrm{CV}(1) < \mathrm{CV}(2)$ *with probability tending to one as m and n increase.*

Proof. Let X and Y be jointly multivariate normal distributed with mean $\mathbf{0}$ and covariance matrix Σ_{XY}, i.e.

$$(X, Y) \sim \mathcal{N}(\mathbf{0}, \Sigma_{XY})$$

where $\Sigma_{XY} = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix}$.

Let $\Sigma_{YY} = U \Lambda U^T$ be the eigendecomposition of Σ_{YY}, with leading eigenvalue λ_1 and corresponding eigenvector u_1. Then the centroid of k-means applying on $(y_1, .., y_n)$ is on the first PC of Y,

$$E(u_1^TY|u_1^TY > 0) = \bar{\mu}_1^Y = \sqrt{2\lambda_1/\pi}u_1$$

and

$$E(u_1^TY|u_1^TY < 0) = \bar{\mu}_2^Y = -\sqrt{2\lambda_1/\pi}u_1$$

where $u_1^TY \sim \mathcal{N}(0, \lambda_1)$.

To compute $\bar{\mu}_1^X = E(X|u_1^TY > 0)$, we need to know the conditional distribution $X|u_1^TY$. Since (X, Y) has multivariate normal distribution, (X, u_1^TY) also has a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix

$$\Sigma_{X,u_1^TY} = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY}u_1 \\ u_1^T\Sigma_{YX} & \lambda_1 \end{bmatrix}$$

The conditional distribution $X|u_1^TY$ is hence normal with mean

$$\mu_{X|u_1^TY} = \Sigma_{XY}u_1\lambda_1^{-1}u_1^TY$$

Therefore,

$$\begin{aligned}
\bar{\mu}_1^X &= E(X \mid u_1^TY > 0) \\
&= E\left(E[X \mid u_1^TY] \mid u_1^TY > 0\right) \\
&= E\left(\Sigma_{XY}u_1\lambda_1^{-1}u_1^TY \mid u_1^TY > 0\right) \\
&= \lambda_1^{-1}\Sigma_{XY}u_1E(u_1^TY \mid u_1^TY > 0) \\
&= \lambda_1^{-1}\Sigma_{XY}u_1\sqrt{2\lambda_1/\pi} \\
&= \sqrt{2/\lambda_1\pi}\Sigma_{XY}u_1
\end{aligned}$$

Similar calculation yields $\bar{\mu}_2^X = -\sqrt{2/\lambda_1\pi}\Sigma_{XY}u_1$. The decision rule to classify any observed value of X to $\bar{\mu}_1^X$ is therefore

$$(\bar{\mu}_1^X)^TX > 0 \quad \text{or} \quad u_1^T\Sigma_{YX}X > 0$$

Since $u_1^T \Sigma_{YX} X$ is a linear combination of X, it also has normal distribution

$$\mathcal{N}\left(0, u_1^T \Sigma_{YX} \Sigma_{XX} \Sigma_{XY} u_1\right)$$

And $(Y, u_1^T \Sigma_{YX} X)$ also have multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix

$$\begin{bmatrix} \Sigma_{YY} & \Sigma_{YX}\Sigma_{XY}u_1 \\ u_1^T \Sigma_{YX}\Sigma_{XY} & u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1 \end{bmatrix}$$

The conditional distribution of $Y|u_1^T \Sigma_{YX} X$ is also multivariate normal with mean

$$\mu_{Y|u_1^T \Sigma_{YX} X} = \Sigma_{YX}\Sigma_{XY}u_1(u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1)^{-1}u_1^T \Sigma_{YX} X$$

The Y center for $u_1^T \Sigma_{YX} X > 0$ is

$$\begin{aligned} \hat{\mu}_1^Y &= E(Y|u_1^T \Sigma_{YX} X > 0) \\ &= \Sigma_{YX}\Sigma_{XY}u_1(u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1)^{-1}E(u_1^T \Sigma_{YX} X \mid u_1^T \Sigma_{YX} X > 0) \end{aligned}$$

Note that $u_1^T \Sigma_{YX} X$ has normal distribution $\mathcal{N}\left(0, u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1\right)$, so

$$E(u_1^T \Sigma_{YX} X \mid u_1^T \Sigma_{YX} X > 0) = \sqrt{2/\pi} \cdot \sqrt{u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1}$$

Therefore, we have the Y center for $u_1^T \Sigma_{YX} X > 0$ be

$$\begin{aligned} \hat{\mu}_1^Y &= \sqrt{2/\pi} \cdot \sqrt{u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1} \;\; \Sigma_{YX}\Sigma_{XY}u_1(u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1)^{-1} \\ &= \frac{\sqrt{2/\pi}}{\sqrt{u_1^T \Sigma_{YX}\Sigma_{XX}\Sigma_{XY}u_1}} \Sigma_{YX}\Sigma_{XY}u_1 \end{aligned}$$

Recall that $\bar{\mu}_1^Y = \sqrt{2\lambda_1/\pi}u_1$, to judge if $CV(2) > CV(1)$, one only need to compare the distance between $\hat{\mu}_1^Y$ and $\bar{\mu}_1^Y$ with distance between $\hat{\mu}_1^Y$ and grand mean 0. By variance and bias decomposition of prediction MSE, when variance is the same, only bias influence the MSE.

After some linear algebra manipulation, we get $||\hat{\mu}_1^Y - \bar{\mu}_1^Y||^2 > ||\hat{\mu}_1^Y||^2$ or $CV(2) > CV(1)$ iff

$$\frac{\sqrt{\lambda_1}}{2} > \frac{u_1^T \Sigma_{YX} \Sigma_{XY} u_1}{\sqrt{u_1^T \Sigma_{YX} \Sigma_{XX} \Sigma_{XY} u_1}}$$

□

Although the condition given by Proposition 2 is succinct in such that it's straight forward to see how the structure of covariance matrix Σ_{XY} affects the performance of Gabriel CV method, it's not simple and easy to interpret such condition as the one in Proposition 1. In specific, each block of matrix Σ_{XY} influences the condition while the combined effect determines whether Gabriel CV method chooses $k = 1$ or $k = 2$.

In practice, check such condition is tricky. One need to know the covariance matrix of each cluster to check if the condition holds. However, in order to estimate the covariance matrix we need to know the cluster center of each cluster, which in turn need to know the number of clusters in the data. A possible solution is following

1. For parameter k starts at $k = 1$ do the following
   (i) Estimate k cluster centers by applying k-means with parameter k
   (ii) Estimate the covariance matrix $\hat{\Sigma}$ for each cluster
   (iii) Check the condition for each $\hat{\Sigma}$
   (iv) If the the condition doesn't hold, rotate that cluster through rotation matrix R^T, where $\hat{\Sigma} = RLR^T$
   (v) Apply previously proposed Gabriel CV method on the rotated data, get estimated number k^*
2. If $k^* = k$, stop the algorithm and return k; otherwise repeat step 1 for parameter $k + 1$

from which we can see the condition checking step can not be separated from the proposed Gabriel CV method. Another empirically observed solution is to leave out most columns for clustering, i.e. increase the dimension of Σ_{YY}. Such method is observed reducing the impact of correlation on the performance of the proposed method. Although the theory for such solution is available for special cases, its general version hasn't been developed.

4.2 Two Clusters

We will now analyze a simple two-cluster setting, and derive conditions for Gabriel cross-validation to correctly prefer $k = 2$ clusters to $k = 1$.

Proposition 3. *Suppose that $\{(X_i, Y_i)\}_{i=1}^{n+m}$ is data from a single fold of Gabriel cross-validation, where each (X, Y) pair in $\mathbb{R}^2$ is an independent draw from an equiprobable mixture of two multivariate normal distributions with identity covariance. Suppose that the first mixture component has mean (μ^X, μ^Y), and the second has mean $(-\mu^X, -\mu^Y)$, where $\mu^X > 0$ and $\mu^Y > 0$. If $1+2\Phi(\mu^Y)+\frac{2\varphi(\mu^Y)}{\mu^Y} < 4\Phi(\mu^X)$, then $\mathrm{CV}(2) < \mathrm{CV}(1)$ with probability tending to one as m and n increase.*

Proof. There are two clusters G_1 and G_2, where observations from G_1 are distributed as

$$\mathcal{N}\left(\begin{pmatrix}\mu^X\\ \mu^Y\end{pmatrix}, \mathbf{I}\right)$$

and observations from G_2 are distributed as

$$\mathcal{N}\left(\begin{pmatrix}-\mu^X\\ -\mu^Y\end{pmatrix}, \mathbf{I}\right)$$

where $\mu_1^X > 0$ and $\mu_1^Y > 0$. Let G_i be the true cluster where observation i is generated from, by assumption

$$P(G_i = G_1) = P(G_i = G_2) = 1/2$$

After applying k-means on $\{Y_i\}_{i=1}^n$ with $k = 2$, if n is large enough, we have the estimated centroids $\bar{\mu}_1^Y$ and $\bar{\mu}_2^Y$ be close to $E(Y \mid Y > 0)$ and $\mathrm{E}(Y \mid Y < 0)$, with errors will be of size $O_p(n^{-1/2})$. Here

$$\begin{aligned} E(Y \mid Y > 0) &= E(Y_1 \mid Y_1 > 0) \cdot P(Y_1 > 0) + E(Y_2 \mid Y_2 > 0) \cdot P(Y_2 > 0) \\ &= 2\varphi(\mu^Y) + 2\mu^Y\Phi(\mu^Y) - \mu^Y \qquad (1) \end{aligned}$$

where $Y_1 \sim N(\mu^Y, 1)$ and $Y_2 \sim N(-\mu^Y, 1)$, and we used Lemma 3 (Appendix B). Similarly, we have

$$\begin{aligned} E(Y \mid Y < 0) &= E(Y_1 \mid Y_1 < 0) \cdot P(Y_1 < 0) + E(Y_2 \mid Y_2 < 0) \cdot P(Y_2 < 0) \\ &= -2\varphi(\mu^Y) - 2\mu^Y \Phi(\mu^Y) + \mu^Y \end{aligned} \tag{2}$$

where $\varphi()$ and $\Phi()$ are the standard normal probability and cumulative distribution function respectively.

Same as in single cluster case, the classification rule learned from $\{(X_i, \hat{G}_i^Y)\}_{i=1}^n$ variables will be determined according to whether $X > 0$, with the decision boundary be at $0 + O_p(n^{-1/2})$. By symmetry, the CV error for points from G_1 is same as the points from G_2. Because $P(G = G_1) = P(G = G_2) = 1/2$, the CV error can be calculated solely from G_2, that is

$$\begin{aligned} \text{CV}(2) &= \frac{1}{m} \sum_{i=n+1}^{n+m} \|(Y_i - \bar{\mu}_1^Y) 1\{\hat{G}_i^X = 1\}\|^2 + \|(Y_i - \bar{\mu}_2^Y) 1\{\hat{G}_i^X = 2\}\|^2, \quad Y \sim N(-\mu^Y, 1) \\ &= \text{E}[(Y - a)^2 1\{X > 0\}] + \text{E}[(Y + a)^2 1\{X < 0\}] + O_p(m^{-1/2}) + O_p(n^{-1/2}) \end{aligned}$$

With

$$\begin{aligned} \text{E}[(Y - a)^2 1\{X > 0\}] + \text{E}[(Y + a)^2 1\{X < 0\}] =& P(\hat{G}_i^X = 1) \cdot E[(Y - a)^2] + P(\hat{G}_i^X = 2) \cdot E[(Y + a)^2] \\ =& [1 - \Phi(\mu^X)][var(Y) + (-\mu^Y - a)^2] + \\ & \Phi(\mu^X)[var(Y) + (-\mu^Y + a)^2] \\ =& [1 - \Phi(\mu^X)][1 + (\mu^Y + a)^2] + \Phi(\mu^X)[1 + (\mu^Y - a)^2] \\ =& 1 + (\mu^Y + a)^2 - 4a\Phi(\mu^X)\mu^Y \end{aligned}$$

where a is given by (1).

When $k = 1$, the result is straight forward since the estimated centroid will approximately equal to 0, with error of size $O_p(n^{-1/2})$. The cross-validation error will be

$$\text{CV}(1) = \frac{1}{m} \sum_{i=n+1}^{n+m} \|Y_i - \bar{Y}_n\|^2 = 1 + (\mu^Y)^2 + O_p(m^{-1/2}) + O_p(n^{-1/2}).$$

So if we have $1 + 2\Phi(\mu^Y) + \frac{2\varphi(\mu^Y)}{\mu^Y} < 4\Phi(\mu^X)$, we have CV(2) < CV(1) □

We confirm this result with a simulation. We perform 10 replicates for each pair of (μ^X, μ^Y), where both μ^X and μ^Y take value on grid of $[0, 3]$ with step 0.1. In each replicate, we generate 20000 observations from two multivariate normal distributions with identity covariance, where one has mean (μ^X, μ^Y) and the other one has mean $(-\mu^X, -\mu^Y)$. We perform a single 2×2 fold of Gabriel cross-validation and report the times (out of 10 replicates) when $k = 2$ is selected by the algorithm in stead of $k = 1$. Figure 3 shows the frequency $k = 2$ is selected by the algorithm for each pair of (μ^X, μ^Y). The dark spot means high number (close to 10) is selected by the algorithm, which means algorithm very likely will pick $k = 2$ over $k = 1$ for the corresponding (μ^X, μ^Y). While light spot means algorithm prefer $k = 1$ for the corresponding value of (μ^X, μ^Y). We can see the simulation result perfectly align with the theoretical curve (the black line), which separates the $k = 2$ zone from the $k = 1$ zone. It demonstrates that the Gabriel cross-validation works exactly as it suppose to under such setting. The position of dark spots shows that when the two clusters are reasonably apart (not overlapping too heavily) in both dimensions, the Gabriel cross-validation is asymptotically consistent.

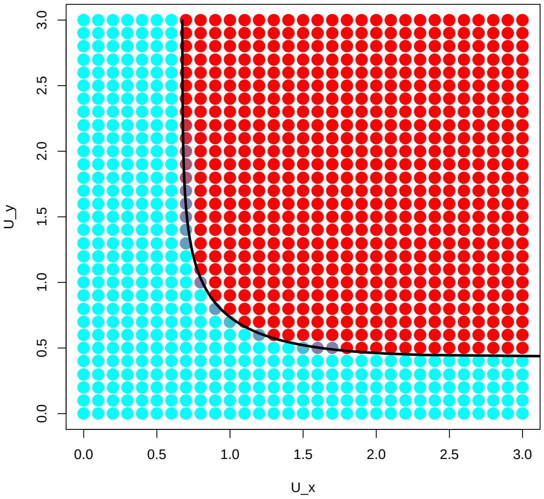

Figure 3: Number of times $k = 2$ is selected out of 10 replicates for each pair of (μ^X, μ^Y). The heat map shows the frequency $k = 2$ is selected by the algorithm, with light means low number (of $k = 2$) is selected and dark color means high number is selected. The black line is the theoretical curve, above which the algorithm suppose to pick $k = 2$ and below which algorithm select $k = 1$.

5 Simulation

In this section, simulation is performed to evaluate the performance of our proposed methods in locating the "correct" number of clusters. We compare with a basket of existing methods including Gap statistics (Tibshirani et al., 2001), Gaussian mixture model-based clustering (Fraley and Raftery, 2002), CH-index (Caliński and Harabasz, 1974), Hartigan statistics (Hartigan, 1975), Jump method (Sugar and James, 2003), Prediction strength (Tibshirani and Walther, 2005), Bootstrap stability (Fang and Wang, 2012) in following simulation settings. We select $p = q$ and $5-$fold cross-validation in row ($m = \frac{1}{4}n$) as default parameter setting for our proposed Gabriel method. Note that set $p = q$ corresponding to $2-$fold cross-validation in column.

1. Single cluster in 10 dimensions — 200 observations, each observation uniformly distributed over $[0, 1]$ in each dimension.

2. Two clusters in 4 dimensions — 50 i.i.d observations are generated from both mul-

tivariate normal $\mathcal{N}(\mu_1, 0.5\Sigma)$ and multivariate normal $\mathcal{N}(\mu_2, 1.5\Sigma)$, where cluster center $\mu_1 = (1,0,0,1)$ and $\mu_2 = (1,3.5,3.5,1)$ respectively. Σ has $AR(1)$ structure with $\rho = -0.2$ and $\sigma = 1$.

3. Four clusters in 100 dimensions — Each cluster has 100 or 150 i.i.d standard normal observations, with cluster centers randomly generated from multivariate normal distribution $\mathcal{N}(\mathbf{0}, 0.65^2\mathbf{I})$

4. Ten clusters in 100 dimensions — Each cluster has 50 or 100 i.i.d standard normal observations, with cluster centers randomly generated from multivariate normal distribution $\mathcal{N}(\mathbf{0}, 0.72^2\mathbf{I})$

5. Four log-normal clusters in 16 dimensions — For each cluster, 30 or 60 i.i.d centered log-normal observations are generated from $ln\mathcal{N}(0, 0.5^2)$. The cluster centers are randomly generated from multivariate normal distribution $\mathcal{N}(\mathbf{0}, 1.2^2\mathbf{I})$

6. Three exponential clusters in 20 dimensions with different variance — 40 observations in each cluster are generated from centered exponential distribution ($exp(\lambda) - 1/\lambda$ in each dimension), with $\boldsymbol{\mu}$ randomly generated from $\mathcal{N}(\mathbf{0}, 19\mathbf{I})$ and $\lambda = 1, 1/2, 1/5$ respectively.

Note that in setting 2–6, all clusters are well-separated, i.e. no overlapping. In fact, any simulated clusters with minimum distance less than 1 unit was discarded, so there is clear definition of true number of clusters. The parameters in setting 2–6 are chosen such that about half of the random realization were discarded. The idea is borrowed from Tibshirani et al. (2001).

Table 1 shows the distribution of k selected by each algorithm in each simulation setting. We run k from 1 to 15, and the best k within this range is selected by the algorithms. NA in the table means algorithm did not converge or otherwise failed.

Table 1: Simulation Results

	Estimated Number of Clusters															
Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	NA
Setting 1																
Gap	**96**	·	·	·	·	·	·	1	·	·	·	·	·	1	2	·
Gaussian-Mix	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
CH*	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Hartigan*	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Jump	**0**	·	·	·	·	·	·	·	·	·	·	·	4	25	71	·
Pred. Strength	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Stability*	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Gabriel CV	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Wold CV	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Setting 2																
Gap	·	**74**	24	·	2	·	·	·	·	·	·	·	·	·	·	·
Gaussian-Mix	·	**88**	2	·	·	·	·	·	·	·	·	·	·	·	·	10
CH	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Hartigan	·	**0**	34	26	11	7	3	9	1	4	1	1	1	·	2	·
Jump	·	**1**	·	·	·	·	·	·	·	1	·	3	8	34	53	·
Pred. Strength	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Stability	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·	·	·
Gabriel CV	·	**86**	10	4	·	·	·	·	·	·	·	·	·	·	·	·
Wold CV	·	**67**	24	7	1	1	·	·	·	·	·	·	·	·	·	·
Setting 3																
Gap	·	·	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·
Gaussian-Mix	78	6	2	**1**	4	2	2	·	·	2	·	1	·	·	2	·
CH	·	16	24	**60**	·	·	·	·	·	·	·	·	·	·	·	·
Hartigan	·	·	9	**80**	8	3	·	·	·	·	·	·	·	·	·	·
Jump	·	·	·	**0**	·	·	·	·	·	·	·	·	·	9	91	·
Pred. Strength	47	·	·	**53**	·	·	·	·	·	·	·	·	·	·	·	·
Stability	·	·	·	**35**	56	9	·	·	·	·	·	·	·	·	·	·
Gabriel CV	·	·	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·
Wold CV	·	·	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·

Continued on next page

Table 1 – *Continued from previous page*

	Estimated Number of Clusters															
Method	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	NA
Setting 4																
Gap	·	·	·	·	·	·	·	·	·	**100**	·	·	·	·	·	·
Gaussian-Mix	44	5	10	9	3	5	5	4	5	**3**	3	2	1	1	·	·
CH	·	38	16	12	7	10	7	4	2	**4**	·	·	·	·	·	·
Hartigan	·	·	1	·	5	8	5	19	19	**16**	13	5	9	·	·	·
Jump	·	·	·	·	·	·	·	·	·	**100**	·	·	·	·	·	·
Pred. Strength	100	·	·	·	·	·	·	·	·	**0**	·	·	·	·	·	·
Stability	·	·	·	·	·	·	·	·	·	**0**	9	16	37	23	15	·
Gabriel CV	·	·	·	·	·	·	·	·	·	**100**	·	·	·	·	·	·
Wold CV	·	·	·	·	·	·	·	·	·	**100**	·	·	·	·	·	·
Setting 5																
Gap	·	·	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·
Gaussian-Mix	·	·	·	**61**	31	7	1	·	·	·	·	·	·	·	·	·
CH	·	3	15	**59**	19	4	·	·	·	·	·	·	·	·	·	·
Hartigan	·	·	17	**63**	9	4	1	4	1	1	·	·	·	·	·	·
Jump	·	·	·	**1**	·	·	·	·	·	·	·	·	·	28	71	·
Pred. Strength	76	5	9	**10**	·	·	·	·	·	·	·	·	·	·	·	·
Stability	·	3	1	**23**	44	13	9	2	·	·	·	1	1	2	1	·
Gabriel CV	·	·	·	**100**	·	·	·	·	·	·	·	·	·	·	·	·
Wold CV	·	·	1	**96**	3	·	·	·	·	·	·	·	·	·	·	·
Setting 6																
Gap	·	·	**0**	·	·	·	·	·	1	1	4	7	11	34	42	·
Gaussian-Mix	·	·	**84**	15	·	·	·	·	·	·	·	·	·	·	·	1
CH	·	13	**73**	13	1	·	·	·	·	·	·	·	·	·	·	·
Hartigan	·	·	**91**	5	2	·	1	·	1	·	·	·	·	·	·	·
Jump	·	·	**0**	·	·	·	·	·	·	·	·	4	11	40	45	·
Pred. Strength	14	·	**86**	·	·	·	·	·	·	·	·	·	·	·	·	·
Stability	·	·	**18**	44	29	6	·	·	·	·	·	·	·	·	·	3
Gabriel CV	·	·	**99**	1	·	·	·	·	·	·	·	·	·	·	·	·
Wold CV	·	·	**90**	8	2	·	·	·	·	·	·	·	·	·	·	·

* CH, Hartigan, and Stability are excluded from Setting 1 because they cannot select $k = 1$.

All methods are used with their default parameter setting except for Gap statistics. We selected the k corresponds to the global maximum of gap statistics, rather than the default method which selects the smallest k such that the gap statistics is not more than 1 standard error away from the first local maximum. This is because among all the possible options, the global maximum criteria gives the best result in our simulation settings.

For CH-index, Hartigen and Bootstrap stability, the minimum k they can pick is 2. So they are not included for comparison since the true $k = 1$. It also demonstrates the difficulty that null scenario poses for some existing algorithms. For those could select $k = 1$, most of them work well under such setting except for Jump method. Setting $2 - 4$ have Gaussian distribution, where the Gaussian mixture model-based method suppose to perform the best. However, it only performs reasonably well in setting 2. In setting 3 and 4, it has a very poor performance even if the underlying distribution is indeed Gaussian mixture model. It highlights the difficulty that high dimension data could cause. Other methods (CH-index, Hartigan, Jump method, Prediction strength and Boostrap stability) also have problems in these two high-dimension settings as we could not find any one with reasonable performance in terms of finding the correct k in both cases. The only exception is Gap statistics, which works perfectly in both settings. Our proposed Gabriel method (also Wold CV method) clearly stands out from the basket of existing methods under the high-dimension setting 3 and 4, as we can see they both perform perfectly under such settings.

Setting 5 and 6 corresponding to the situation where data are heavy-tailed instead of normal distributed, coupled with unequal number of observations or unequal variance for different clusters. From Table 1, we can see none of the existing methods has good performance in both settings. In contrast, our proposed Gabriel method is robust and has superior performance compare with other methods. Because one can hardly tell the underlying distribution of data in practice, the resilience to non-Gaussian data gives the Gabriel method a clear edge.

6 Real data application

We also applied our proposed method to three real world data sets obtained from the University of California Irvine machine learning repository. The first and third data sets are selected because there are clear number of clusters in those two data sets. The second data set is used as a benchmark data set since it was widely used in literature.

The first one is congress voting data which consists of voting records of 98th United States Congress, 2nd session (Schlimmer, 1987). This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the *CQA* (Congressional Quarterly Almanac). For each vote, each Congressman either vote positively "yea" (voted for/paired for/announced for), negatively "nay" (voted against/paired against/announced against) or position unknown "?". We took out those records contain "?" before comparing each algorithm. It results in 232 remaining records, with 124 democrat and 108 republican.

The second data set is the well-known Wisconsin breast cancer data set (Mangasarian et al., 1990). After excluding the records with missing data, this data set consists records of 683 patients, each with measurements of nine attributes of their biopsy specimens. It is known that there exist at least two groups of patients: 444 patients with benign specimens and 239 patients with malignant specimens.

The third data set is the Sonar data returned from two targets – a metal cylinder and a rock with similar shape, which is first studied by Gorman and Sejnowski (1988) using a neural network. Both targets were impinged by pulse which was a wide-band linear FM chirp ($ka = 55.6$). Returns were collected at a range of 10 meters and obtained from the cylinder at aspect angles spanning 90° and from the rock at aspect angles spanning 180°. The data set contains 208 returns (111 cylinder returns and 97 rock returns), with each composed of 60 spectral samples, normalized to take on values between 0 and 1. So the data has 60 features with clearly 2 clusters.

Table 2: Number of clusters selected by each algorithm

	Congress Voting	Breast Cancer	Sonar
CH-index	2	2	2
Hartigan	3	3	3
Jump	9	9	10
Prediction strength	2	2	1
Bootstrap stability	2	2	10
Gap	10	9	10
Gaussian-Mix	7	5	1
Gabriel	2	3	2
Wold	2	3	10

All the algorithms executed with their default parameter settings with k ranges from 1 to 10

Since most congressmen vote base on their parties' interest, 2 parties (Democratic and Republican) represent two clusters in this data set. So the optimal number should be two. Close inspection shows k-means with $k = 2$ separates the two parties very well with the lowest miss-classification error 10.43% ($k = 3$ has 14.78%). Note that CH-index and Bootstrap stability also return $k = 2$, but 2 is the lower bound those methods can select for k. So it's not clear they actually choose $k = 2$ or they hit the lower bound (they would pick $k = 1$ if allowed). For the breast cancer data, it's known to have at least 2 cluster based on whether it's benign specimens or not. But it doesn't mean the optimal k should be 2. Fujita et al. (2014) noticed that the malign group is quite heterogeneous and can be further clustered into at least two subgroups. Hence, the result $k = 3$ given by our proposed Gabriel method (as well as Wold method) make sense. For the Sonar data set which is relatively high-dimension, only our proposed Gabriel method and CH-index selects $k = 2$. Majority of the methods collapse to either select the maximum or select the minimum number, underline how difficult it was to pick the right k when dimension increases.

7 Discussion

In this paper, we proposed a novel approach to estimate the number of clusters. The intuition behind our proposed methods is to transfer the unsupervised learning problem into supervised learning problem via novel form of cross validation. Such approach is quite different from previous methods which utilize the within/between cluster dispersion or stability criterion for selecting the optimal k. Our method utilizes the connection between different dimensions (columns) of data through the uniqueness of each cluster center. We proved the self-consistency for our proposed Gabriel CV method as well as its asymptotic property with Gaussian noise, and showed the robustness of our method by simulation. Our method has very good performance in our limited simulation settings and real data application, and clearly the superior one when data is high dimension or is heavy-tailed.

Besides no modeling assumption is required, our proposed method is robust against data set with variance heterogeneity, unequal number of observations, non-Gaussian noise and high-dimension. Such robustness is important in practice because for any given data, it's hard to tell whether its clusters have different number of observations, is the variance equal for each cluster, or what underlying noise distribution it has. The weakness of our proposed method is that its theory assumes only week correlation between "predictor" columns and "response" columns. In practice, many data sets don't exhibit high correlations between columns, where our proposed method can be safely applied. In case the high correlation does exist, some procedure such as rotating the data set and/or leave out most columns for clustering may be used to reduce its effect as previously mentioned. However, the theory for those procedure have not been fully developed and it could be the future research topic.

Another situation where our proposed method cannot be directly used is when the clusters are non-convex. In such situation, k-means itself doesn't work well, for example two concentric circles share the same cluster center (Hastie et al., 2009). However, it's possible that our proposed Gabriel CV method can be used on the transformed data set. In the concentric circles case where spectral clustering is appropriate, our proposed method can be applied on the eigenvector subspace of the graph Laplacian matrix inside the spectral clustering algorithm to find the optimal k. This can also be the future research topic.

APPENDIX

A Wold CV estimation

- For each $k = 1, 2, ..., k_{max}$

  1. Randomly draw some entries in $\mathbf{X}$ missing, keep those hold-out values in vector V_{true}
  2. Impute the missing values with column mean or 0, denote the imputed data as $\mathbf{X}_{new}$
  3. Apply the iterative procedure below until converge or stopping criteria reached
     - Apply K-mean on data set $\mathbf{X}_{new}$ with parameter k
     - Substitute each observation in $\mathbf{X}_{new}$ by its nearest center, get new data $\mathbf{X}^c_{new}$ ($\mathbf{X}_{new}$ keep the same)
     - Replace (impute) those imputed values in $\mathbf{X}_{new}$ with the corresponding entries in $\mathbf{X}^c_{new}$
     - Calculate the difference between the old and newly imputed values, check whether or not they coincide (converge)
  4. Obtain the last imputed entry values of converged $\mathbf{X}_{new}$, denote it by $V_{converge}$
  5. Calculate the prediction error $Error_k = ||V_{true} - V_{converge}||^2$

- For each CV folder, repeat above procedure and obtain the $Error_k$ for each k
- Average $Error_k$ across all folders for each k, and then select the k corresponding to the minimum average $Error_k$

B Technical Lemmas

Lemma 3. *If Z is a standard normal random variable, then*

$$\mathrm{E}(Z \mid a < Z < b) = -\frac{\varphi(b) - \varphi(a)}{\Phi(b) - \Phi(a)}$$

and

$$\mathrm{E}\{(Z-\delta)^2 \mid a < Z < b\} = \delta^2 + 1 - \frac{(b-2\delta)\varphi(b) - (a-2\delta)\varphi(a)}{\Phi(b)-\Phi(a)}$$

for all constants a, b, and δ, where $\varphi(z)$ and $\Phi(z)$ are the standard normal probability density and cumulative distribution functions. These expressions are valid for $a = -\infty$ or $b = \infty$ by taking limits.

Proof. We will derive the expression for the second moment. Integrate to get

$$\begin{aligned}\mathrm{E}[(Z-\delta)^2 1\{Z < b\}] &= \int_{-\infty}^{b} (z-\delta)^2 \varphi(z)\, dz \\ &= (\delta^2+1)\Phi(b) - (b-2\delta)\varphi(b).\end{aligned}$$

Now,

$$\mathrm{E}\{(Z-\delta)^2 \mid a < Z < b\} = \frac{\mathrm{E}[(Z-\delta)^2 1\{Z < b\}] - \mathrm{E}[(Z-\delta)^2 1\{Z < a\}]}{\Phi(b)-\Phi(a)}.$$

□

Lemma 3 has some important special cases:

$$\begin{aligned}\mathrm{E}\{Z \mid Z > 0\} &= 2\varphi(0) = \sqrt{2/\pi}, \\ \mathrm{E}\{(Z-\delta)^2 \mid Z > 0\} &= \delta^2 + 1 - 4\delta\varphi(0), \\ \mathrm{E}\{(Z-\delta)^2 \mid Z < 0\} &= \delta^2 + 1 + 4\delta\varphi(0).\end{aligned}$$

References

Ben-Hur, A., Elisseeff, A., and Guyon, I. (2001). A stability based method for discovering structure in clustered data. In *Pacific symposium on biocomputing*, volume 7, pages 6–17.

Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Chiang, M. M.-T. and Mirkin, B. (2010). Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *Journal of classification*, 27(1):3–40.

Fang, Y. and Wang, J. (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis*, 56(3):468–477.

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

Fujita, A., Takahashi, D. Y., and Patriota, A. G. (2014). A non-parametric method to estimate the number of clusters. *Computational Statistics & Data Analysis*, 73:27–39.

Gabriel, K. R. (2002). Le biplot–outil d'exploration de données multidimensionelles. *Journal de la Société Francaise de Statistique*, 143:5–55.

Gorman, R. P. and Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural networks*, 1(1):75–89.

Hartigan, J. A. (1975). *Clustering Algorithms.* Wiley.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediciton.* Springer Series in Statistics. Springer, 2nd edition.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

Mangasarian, O. L., Setiono, R., and Wolberg, W. (1990). Pattern recognition via linear programming: Theory and application to medical diagnosis. *Large-scale numerical optimization*, pages 22–31.

Owen, A. B. and Perry, P. O. (2009). Bi-cross-validation of the svd and the nonnegative matrix factorization. *Ann. Appl. Stat.*, 3(2):564–594.

Perry, P. O. (2009). *Cross-Validation for Unsupervised Learning.* PhD thesis, Stanford University.

Pollard, D. (1981). Strong consistency of k-means clustering. *Ann. Stat.*, 9(1):135–140.

Pollard, D. (1982). A central limit theorem for k-means clustering. *Ann. Probab.*, 10(4):919–926.

Schlimmer, J. C. (1987). Concept acquisition through representational adjustment.

Sugar, C. A. and James, G. M. (2003). Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463).

Tibshirani, R. and Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika*, 97(4):893–904.

Wold, S. (1978). Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20:397–405.