2015

Cluster Analysis with Batch Effect

Yifan Tang
University of South Carolina - Columbia

CLUSTER ANALYSIS WITH BATCH EFFECT

by

Yifan Tang

Bachelor of Science
Xiamen University, 2013

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Public Health in

Biostatistics

The Norman J. Arnold School of Public Health

University of South Carolina

2015

Accepted by:

Alexander C. McLain, Director of Thesis

Jiajia Zhang, Reader

James Hussey, Reader

Lacy Ford, Senior Vice Provost and Dean of Graduate Studies

Abstract

Clustering, as a fundamental process in data science, is frequently used in preliminary data analysis. Batch effects are a powerful source of variation that can come from many sources in data collection, and influence data. We propose a method to simultaneously remove batch effects and perform cluster analysis. We see a batch effect as a fixed value added on to each batch, and do not make assumptions about the distribution of batch effects. We represent the data using a Gaussian mixture model, and use the EM algorithm to estimate the cluster means, the cluster covariance matrices, and the batch effects, and give predictions on which cluster each observation belongs to via their posterior probability. We also give two tests to identify the presence of batch effects in the data. Gap statistics are used to determine the number of clusters that should be used.

We compare our method via simulation studies with a standard K-means method and K-means with the batch effects removed prior to analysis. Out simulations studies our method has better prediction results than both of these approaches. Our method does not assume the batch effects following any particular distribution, and works on data that have larger batch effects, as well as an interaction between clusters and batches.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Classification and clustering are fundamental processes in data science. Setting up classes of subjects according to similarity in multivariate data is the first step to understand and develop models to explain and predict the data. Classification and clustering are two terms in machine learning that are used in supervised and unsupervised learning, respectively. In supervised leaning the outcome (classes) are known. In unsupervised learning, we don't have any information about the underlying classes. In this thesis we will consider the latter situation where the natural classes are unknown.

Batch effects occur when we are gathering data from different sources, different time periods, or trying to use results from different labs. They have been observed from the earliest microarray experiment (Lander, 1999) and they are also inevitable when new data are added to existing data, or in a meta-analysis of multiple studies (Rhodes et al., 2004). Batch effects are a type of correlated measurement error that can be a powerful source of variation in experiments (Leek et al., 2010). They occur because measurements taken at the same time, or in the same "batch", are similarly affected by lab conditions, reagent lots, personnel differences, etc. Batch effects occur due to

quantitative differences across conditions, which are uncorrelated with the variables of interest.

For example, in the Upstate Kids Study (G. M. Buck Louis et al., 2014), 65 pro and anti-inflammatory cytokines were measured on 3944 newborns from Human Obesity Panel (R&D Systems) by Luminex. The Luminex platform measures the cytokines in batches of 36. For each batch, samples with known concentrations, called standards, are included and measured. The readings from the standard samples are then used to calibrate the readings of the newborn samples through the use of five-parameter logistic model. This batch-by-batch calibration creates a "batch effect", a correlated measurement error from all measurements in the same batch. If ignored, the batch effect will result in incorrect clustering due to the correlated measurement error instead of biological similarities.

There are four types of methods to adjust for batch effects. The first one is normalization. In order to adjust for the biases caused by non-biological effects, researchers developed normalization methods (Schadt et al., 2001; Tseng et al., 2001; Yang et al., 2002). Data are normalized such that observations in each batch have mean 0 and variance 1.

The second type of method is based on a singular-value decomposition (SVD), adjusting data by identifying the eigenvector referred to batch effects (Alter, 2000). Benito et al.(2004) use distance weighted discrimination (DWD) to adjust for batch effects by finding a hyperplane where the batch effects are most significant when projecting data on this plane, and then removing the batch effects. This type of method is

complicated and requires a large batch sample.

The third method is a model based location/scale (L/S) adjustment. In this method researchers assume a model for the location (mean) and/or scale (variance) of the data within batches and then adjust the parameters to meet model assumptions.

The fourth method is an empirical Bayes method (Johnson et al., 2007). In this method, the L/S model parameters that represent the batch effects are estimated by assuming prior distributions of batch effects. The EB estimates for batch effect parameters are given by conditional posterior means. This method provides robust adjustments for the batch effect on each observation. However when there is interaction between batches and biological factors, the batch effect removal process could remove some of the biological variation.

Methods for performing cluster analysis include K-means clustering, Gaussian mixture, hierarchical clustering, and many others. We will discuss Gaussian mixture and K-means clustering in Chapter 2 and Chapter 5, respectively.

In this thesis, we are considering a situation where the data are gathered from different batches and there are systematic measurement error for each batch. Systematic means the error is consistent in each batch. We develop a method to simultaneously cluster and remove the batch effects.

In Chapter 2, we represent the data using a linear mixed model and after we treat the batch effects as fixed values, we can represent the data using a Gaussian mixture model. In the Gaussian mixture model, we can calculate the conditional probabilities

to do the cluster analysis. In Chapter 3, the EM algorithm, the method for estimating the parameters in Gaussian mixture model, is introduced. We also give introduction of the Gap statistics to determine the correct number of clusters. In next chapter, we set up two tests to identify batch effects. In Chapter 5, K-means clustering and a two-stage procedure to remove batch effect are introduced. In Chapter 6, we compare our method via simulation studies with a standard K-means method and K-means with the batch effects removed prior to analysis. In Chapter 7 are the conclusions of our method and simulation studies.

Chapter 2

Statistical background

In this chapter, we introduce linear mixed models and Gaussian mixture models as statistical background to show how our data can be represented as a Gaussian mixture model.

2.1 Linear mixed model

In a linear mixed effect model the data are represented as

$$\boldsymbol{y}_{it(k)} = \boldsymbol{\theta}_k + \widetilde{\boldsymbol{\beta}}_t + \boldsymbol{r}_{it(k)},$$

where $\boldsymbol{y}_{it(k)}$ represents the ith observation in the tth batch belonging to the kth cluster, and similarly for the residual $\boldsymbol{r}_{it(k)}$. $\boldsymbol{\theta}_k$ is the fixed effect, $\widetilde{\boldsymbol{\beta}}_t \sim N(0, \boldsymbol{\Sigma_t})$ and $\boldsymbol{r}_{it(k)} \sim N(0, \boldsymbol{\Sigma_k})$, $\{\boldsymbol{y}_{it(k)}, \boldsymbol{\theta}_k, \widetilde{\boldsymbol{\beta}}_t, \boldsymbol{r}_{it(k)}\}$ are all $1 \times p$ matrices and $\boldsymbol{\Sigma_k}$ is $p \times p$ covariance matrix.

All observations can be represented in the following matrix form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{Z}\widetilde{\boldsymbol{\beta}} + \boldsymbol{r},$$

where $\boldsymbol{Y}$ ($N \times p$ matrix) is the observation matrix, p is the number of biomarkers, $\boldsymbol{X}$ ($N \times K$ matrix) is the matrix indicating cluster membership, K is the number of clusters, $\boldsymbol{\theta}$ ($K \times p$ matrix) is the matrix of the means of the each cluster, $\boldsymbol{Z}$ ($N \times T$ matrix) is the design matrix, indicating which batch the observations belong to, $\widetilde{\boldsymbol{\beta}}$ ($T \times p$ matrix) is the random effects for the batches, T is the number of batches and $\boldsymbol{r}$ ($N \times p$ matrix) contains the residuals.

Here N is the total number of observations, where $N = \sum_{t=1}^{T} n_t$ and n_t is the number of observations in the tth batch.

The estimation procedure in our case is different from the standard mixed modeling. In the standard mixed models, we know $\{\boldsymbol{X}, \boldsymbol{Z}\}$ and want to estimate $\boldsymbol{\theta}$, $\boldsymbol{\Sigma}_t$ and $\boldsymbol{\Sigma}_k$. However, in our case, $\boldsymbol{X}$, $\boldsymbol{\theta}$, $\boldsymbol{\Sigma}_t$ and $\boldsymbol{\Sigma}_k$ are all unknown.

We assume in the observed data, $\boldsymbol{\beta}_t$ ($1 \times p$ matrix) is generated from its distribution only one time and this value, the "batch effect", is added to all the fixed effects in each batch. Since $\boldsymbol{\theta}_k$ and $\boldsymbol{r}_{it(k)}$ are both cluster specific, we can combine these two parts together, which gives us K Gaussian distributions. Each cluster has its own mean and covariance matrix. For each distribution, observations are generated and then batch effects are added according to their batch. Given $\boldsymbol{\beta}_t$ we can express the data as

$$\boldsymbol{y}_{it(k)} \sim N(\boldsymbol{\theta}_k + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_k).$$

Thus, our data can be expressed as observations generated from different Gaussian distributions. Thus arises a Gaussian mixture model, which is similar to our problem, and where methods exist to implement cluster analysis.

2.2 Gaussian mixture model

The Gaussian mixture model was created when researchers noticed some complex distributions couldn't be described as a single distribution. A linear combination of many Gaussian distributions is created to represent more complex distributions (McLachlan and Basford, 1988; McLachlan and Peel, 2000). We therefore consider a superposition of K Gaussian densities of the form

$$p(\boldsymbol{y}) = \sum_{k=1}^{K} \pi_k N\left(\boldsymbol{y}|\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k\right),$$

$N\left(\boldsymbol{y}|\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k\right)$, a multivariate Gaussian distribution, is called the kth component, with the mean $\boldsymbol{\theta}_k$, variance $\boldsymbol{\Sigma}_k$, and π_k is the prior probability of picking the kth component with $\sum_{k=1}^{K} \pi_k = 1$. This can also be viewed as the probability that an arbitrary observation belongs to the kth component (cluster). Figure 2.1 shows an example of a bivariate Gaussian mixture distribution, with 3 bivariate Gaussian distributions and $\pi_k = 1/3$. The left plot marks clusters using different colors.

Here π_k are called the mixing coefficients, and can be formulated as

$$\pi_k = p(z_k = 1),$$

where $z_k \in \{0, 1\}$ and $\sum_{k=1}^{K} z_k = 1$. Here $z_k = 1$ represents the indicator that an observation belongs to the kth cluster. For each observation we have $\boldsymbol{z} = (z_1, z_2, \cdots, z_k)$.

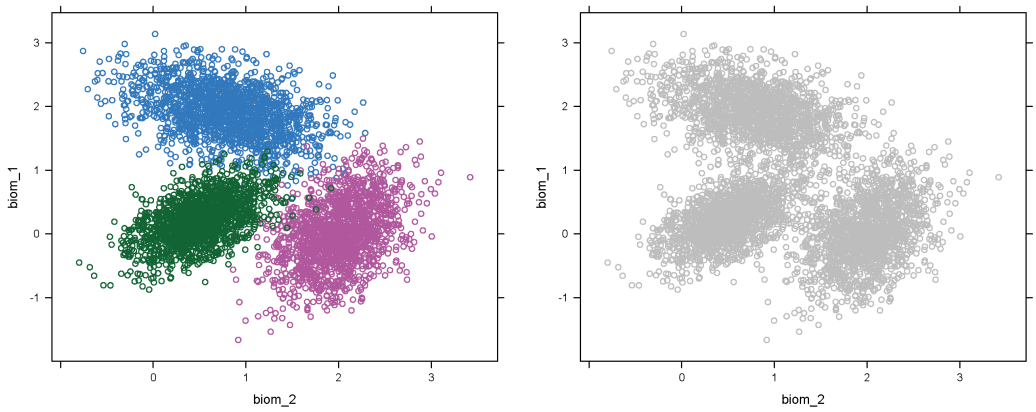

Figure 2.1: Simulation of Gaussian mixture model (scatter plot, K=3)

The distribution of $\boldsymbol{z}$ can be written as

$$p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}.$$

The conditional distribution of $\boldsymbol{y}$ given z_k is

$$p(\boldsymbol{y}|z_k = 1) = N\left(\boldsymbol{y}|\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k\right).$$

Thus,

$$p(\boldsymbol{y}|\boldsymbol{z}) = \prod_{k=1}^{K} N\left(\boldsymbol{y}|\boldsymbol{\theta_k}, \boldsymbol{\Sigma_k}\right)^{z_k}.$$

We can obtain the marginal distribution $p(\boldsymbol{y})$ by summing the joint distribution $p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z})$ over the possible values of $\boldsymbol{z}$

$$p(\boldsymbol{y}) = \sum_{z} p(\boldsymbol{y}|\boldsymbol{z})p(\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k N\left(\boldsymbol{y}|\boldsymbol{\theta_k}, \boldsymbol{\Sigma_k}\right).$$

Notice that we get the Gaussian mixture distribution as shown earlier. We represent

the marginal distribution in the form of a sum of the mixing distributions, which tells us for each observation there exists a latent variable $\boldsymbol{z}$ (we know this is actually the indicator of natural cluster assignment). Now we have equivalent representations of the Gaussian mixture model. To estimate the probability that a given observation belongs in cluster k, we use the posterior probability $p(z_k|\boldsymbol{y})$, denote by $\gamma(z_k)$, which can be obtained using Bayes' theorem as

$$\begin{aligned}\gamma(z_k) = p(z_k = 1|\boldsymbol{y}) &= \frac{p(z_k = 1)p(\boldsymbol{y}|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(\boldsymbol{y}|z_j = 1)} \\ &= \frac{\pi_k N\left(\boldsymbol{y}|\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j N\left(\boldsymbol{y}|\boldsymbol{\theta}_j, \boldsymbol{\Sigma}_j\right)}.\end{aligned}$$

Here, $\gamma(z_k)$ is the conditional probability given $\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k$ and the information of $\boldsymbol{y}$. Another way of interpreting $\gamma(z_k)$ is the responsibility of the kth component takes for explaining the observation $\boldsymbol{y}$.

Chapter 3

Clustering with batch effects

To generalize the Gaussian mixture model to our problem, the probability density function for an individual observation is

$$p(\boldsymbol{y}_{it}) = \sum_{k=1}^{K} \pi_k N\left(\boldsymbol{y}_{it} | \boldsymbol{\theta}_k + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_k\right),$$

where π_k has the exact definition as we mentioned earlier, the prior probability of $z_{it}^k = 1$.

Assume there is a latent indicator $z_{it}^k \quad (k = 1, 2, \ldots, K)$ with $z_{it}^k = 1$ if $\boldsymbol{y}_{it}$ belongs to the kth natural cluster. Thus, the marginal probability of $\boldsymbol{y}_{it}$ is the Gaussian mixture distribution.

Then we define γ_{it}^k for our data, which will play a role in the cluster assignments, as

$$\begin{aligned}\gamma_{it}^k = p(z_{ik} = 1 | \boldsymbol{y}_{it}) &= \frac{p(z_{ik} = 1) p(\boldsymbol{y} | z_{ik} = 1)}{\sum_{j=1}^{K} p(z_{ij} = 1) p(\boldsymbol{y} | z_{ij} = 1)} \\ &= \frac{\pi_k N\left(\boldsymbol{y} | \boldsymbol{\theta}_k + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j N\left(\boldsymbol{y} | \boldsymbol{\theta}_j + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_j\right)}.\end{aligned}$$

Here, γ_{it}^k is the posterior probability, given $\boldsymbol{\theta}_k, \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_k$ and $\boldsymbol{y}_{it}$, that the observation belongs to the kth cluster.

3.1 EM algorithm

The method for finding the maximum likelihood solution for models that contain latent variables is called the expectation-maximization algorithm or EM algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997). To implement the EM algorithm in our problem, the first step is to write down the log-likelihood of a Gaussian mixture model

$$\log p(\boldsymbol{y}|\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\Sigma}) = \sum_{t=1}^{T} \sum_{i=1}^{n_t} \ln \left\{ \sum_{k=1}^{K} \pi_k N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_k + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_k\right) \right\}.$$

When we take the derivative on the above log-likelihood with respect to $\boldsymbol{\theta}_k$ (for the moment, we treat $\boldsymbol{\Sigma}_k, \boldsymbol{\beta}_t$ and π_k as fixed), we get

$$\frac{\partial \log p}{\partial \boldsymbol{\theta}_k} = \sum_{t=1}^{T} \sum_{i=1}^{n_t} \frac{\pi_k N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_k + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K} \pi_j N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_j + \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_j\right)} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{y}_{it} - \boldsymbol{\theta}_k - \boldsymbol{\beta}_t).$$

Notice that the first part behind the summation sign is γ_{it}^k. Then, we set the above equation to 0, and get

$$\widehat{\boldsymbol{\theta}}_k = \frac{\sum_{t=1}^{T} \sum_{i=1}^{n_t} \gamma_{it}^k (\boldsymbol{y}_{it} - \boldsymbol{\beta}_t)}{\sum_{t=1}^{T} \sum_{i=1}^{n_t} \gamma_{it}^k},$$

a weighted average of the observations removing the corresponding batch effect, where the observations that have higher probability of coming from the kth cluster have larger weights.

Following the same steps we take the derivative of the log-likelihood with respect to $\boldsymbol{\beta}_t$ (now treating $\boldsymbol{\Sigma}_k, \boldsymbol{\theta}_k$ and π_k as fixed) and get

$$\widehat{\boldsymbol{\beta}}_t = \frac{\sum_{i=1}^{n_t} \sum_{k=1}^{K} \gamma_{it}^k (\boldsymbol{y}_{it} - \boldsymbol{\theta}_k)}{\sum_{i=1}^{n_t} \sum_{k=1}^{K} \gamma_{it}^k}.$$

Here, $\boldsymbol{\beta}_t$ is estimated through a weighted average of the observed data removing the mean value (cluster mean) in the corresponding batch. Observations that have higher probability coming from the kth cluster have larger weights.

Similarly, we take the derivative of the log-likelihood with respect to $\boldsymbol{\Sigma}_k$ (treating $\boldsymbol{\theta}_k, \boldsymbol{\beta}_t$ and π_k as fixed), and then set it to 0 to get

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{\sum_{t=1}^{T} \sum_{i=1}^{n_t} \gamma_{it}^k (\boldsymbol{y}_{it} - \boldsymbol{\theta}_k - \boldsymbol{\beta}_t)(\boldsymbol{y}_{it} - \boldsymbol{\theta}_k - \boldsymbol{\beta}_t)^T}{\sum_{t=1}^{T} \sum_{i=1}^{n_t} \gamma_{it}^k}.$$

Similarly, this is a weighted average of the sample variances. Note the observations that have higher probability coming from the kth cluster have larger weights.

Next we take the derivative of the log-likelihood with respect to π_k (treating $\boldsymbol{\theta}_k, \boldsymbol{\beta}_t$ and $\boldsymbol{\Sigma}_k$ as fixed). Recall that π_k has the restriction $\sum_{k=1}^{K} \pi_k = 1$, thus we need to account for this restriction when looking for the maximum likelihood solution. We use Lagrange multipliers, and take the derivative of the following function

$$\log p(\boldsymbol{y}|\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right).$$

Doing this we get

$$\sum_{t=1}^{T}\sum_{i=1}^{n_t}\frac{N\left(\boldsymbol{y}_i|\boldsymbol{\theta}_k+\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K}\pi_j N\left(\boldsymbol{y}_i|\boldsymbol{\theta}_j+\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_j\right)}+\lambda\overset{set}{=\!=}0.$$

Multiplying both sides by π_k and sum on k, we get $\lambda=-N$. Plugging in $\lambda=-N$ and multiplying the above equation by π_k, we then recognize the first part behind the summation sign as $\gamma(z_{nk})$. Thus

$$\widehat{\pi}_k=\frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\gamma(z_{ik})}{N},$$

where $N=\sum_{t=1}^{T}n_t$.

Now we can form the EM algorithm:

1. Initialize π_k, $\boldsymbol{\theta}_k$, $\boldsymbol{\beta}_t$ and $\boldsymbol{\Sigma}_k$. We use $\frac{1}{K}$ for all π_k, the estimated cluster means from K-means for $\boldsymbol{\theta}_k$, $\mathbf{0}$ for $\boldsymbol{\beta}_t$, and $\lambda\mathbf{I}_p$ for $\boldsymbol{\Sigma}_k$, where $\lambda=\frac{1}{KP}tr(V)$ and V is the sample covariance matrix.

2. **E step**: Evaluate the probabilities using the current parameter values

$$\gamma_{it}^{k(new)}=\frac{\pi_k N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_k+\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K}\pi_j N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_j+\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_j\right)}$$

3. **M step**: Re-estimate the parameter values using the renewed probabilities

$$\begin{aligned}
\boldsymbol{\beta}_t^{new} &= \frac{\sum_{i=1}^{n_t}\sum_{k=1}^{K}\gamma_{it}^k(\boldsymbol{y}_{it}-\boldsymbol{\theta}_k)}{\sum_{i=1}^{n_t}\sum_{k=1}^{K}\gamma_{it}^k} \\
\boldsymbol{\theta}_k^{new} &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\gamma_{it}^k(\boldsymbol{y}_{it}-\boldsymbol{\beta}_t^{new})}{\sum_{t=1}^{T}\sum_{n=1}^{n_t}\gamma_{it}^k} \\
\boldsymbol{\Sigma}_k^{new} &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\gamma_{it}^k(\boldsymbol{y}_{it}-\boldsymbol{\theta}_k^{new}-\boldsymbol{\beta}_t^{new})(\boldsymbol{y}_{it}-\boldsymbol{\theta}_k^{new}-\boldsymbol{\beta}_t^{new})^T}{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\gamma_{it}^k} \\
\pi_k^{new} &= \frac{\sum_{t=1}^{T}\sum_{i=1}^{n_t}\gamma_{it}^k}{N}
\end{aligned}$$

4. Evaluate the log-likelihood using the renewed parameters

$$\log p(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\Sigma}) = \sum_{i=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k N\left(\boldsymbol{y}_i|\boldsymbol{\theta}_k,\boldsymbol{\Sigma}_k\right)\right\}$$

Then check the convergence criterion:

$$|\log p(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\Sigma})_{new} - \log p(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\Sigma})_{old}| < 1*10^{-5},$$

where $\log p(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\Sigma})_{new}$ is the result we get from the current iteration step, and $\log p(\boldsymbol{y}|\boldsymbol{\pi},\boldsymbol{\theta},\boldsymbol{\Sigma})_{old}$ is the result we get from previous iteration step. If the convergence criterion is meet, go to step 5, otherwise go back to step 2.

5. Predicting clusters. First we update the probabilities

$$\gamma_{it}^{k(new)} = \frac{\pi_k N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_k+\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K}\pi_j N\left(\boldsymbol{y}_{it}|\boldsymbol{\theta}_j+\boldsymbol{\beta}_t,\boldsymbol{\Sigma}_j\right)}$$

Then we assign each observation $\boldsymbol{y}_{it}$ to a cluster using γ_{it}^k, with $cluster_{it} = k^*$, where

k^* satisfies $k^* = \underset{k}{argmax}\{\gamma_{it}^k\}$ for observation i in batch t.

Notice this algorithm makes no assumption on the distributions of the batch effects. Now we have given the method to perform cluster analysis on data with batch effects, we will discuss how to determine the correct number of clusters.

3.2 Gap statistics

Let W_K denote the within cluster variation. W_K is a weighted average of within cluster distances D_k. They are defined as follows

$$\begin{aligned} W_K &= \sum_{k=1}^{K} \frac{1}{2n_k} D_k, \\ D_k &= \sum_{i \in C_k} \sum_{j \in C_k} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \\ &= 2n_k \sum_{i \in C_k} \|\boldsymbol{y}_i - \bar{\boldsymbol{y}}\|^2. \end{aligned}$$

Thus

$$W_K = \sum_{k=1}^{K} \sum_{i \in C_k} \|\boldsymbol{y}_i - \bar{\boldsymbol{y}}\|^2,$$

where n_k is the number of elements in the kth cluster, and C_k represents the kth cluster. W_K can be seen as a sum across clusters of within cluster variation.

W_K is calculated for each number of clusters K (usually start from 2). If K is less than the true number of clusters, some between cluster variation will be calculated into within cluster variation. Naturally, W_K decrease with K. However, we observe

that W_K decreases rapidly at the number of natural clusters. Thus,

$$\{W_K - W_{K-1} \mid K = K^*\} \gg \{W_{K-1} - W_{K-2} \mid K = K^*\},$$

where K^* is the natural number of clusters.

The recently proposed Gap statistic (Tibshirani et al., 2001b) compares a curve of $\log(W_K)$ instead of W_K to a curve obtained from data uniformly distributed over a rectangle containing the data. They use a group of simulated data to calculate $log(W_{Kd}^{'})$ as our reference (d is the dth group of simulations). The simulations are done by the Monte Carlo Algorithm. The Gap Statistic is defined below:

$$Gap(K) = \frac{1}{D}\sum_{d=1}^{D} log(W_{Kd}^{'}) - log(W_K).$$

We want to find the smallest K^* with the largest the gap. The following criterion for choosing the "best" K^* will be utilized.

$$K^* = \underset{K}{argmin}\{K \mid Gap(K) \geq Gap(K+1) - S_{K+1}^{'}\},$$

$$S_K^{'} = S_K\sqrt{1 + 1/D},$$

where $S_K^{'}$ is an adjusted standard deviation of S_K, and S_K is the standard deviation of $\{log(W_{Kd}^{'})\}_{d=1,\ldots,D}$ (D usually takes 20). The advantage of using this choosing criteria is that we only need to do cluster analysis from K=1 to $K^* + 1$ to find the correct number of clusters, which is quite efficient. Further this algorithm can

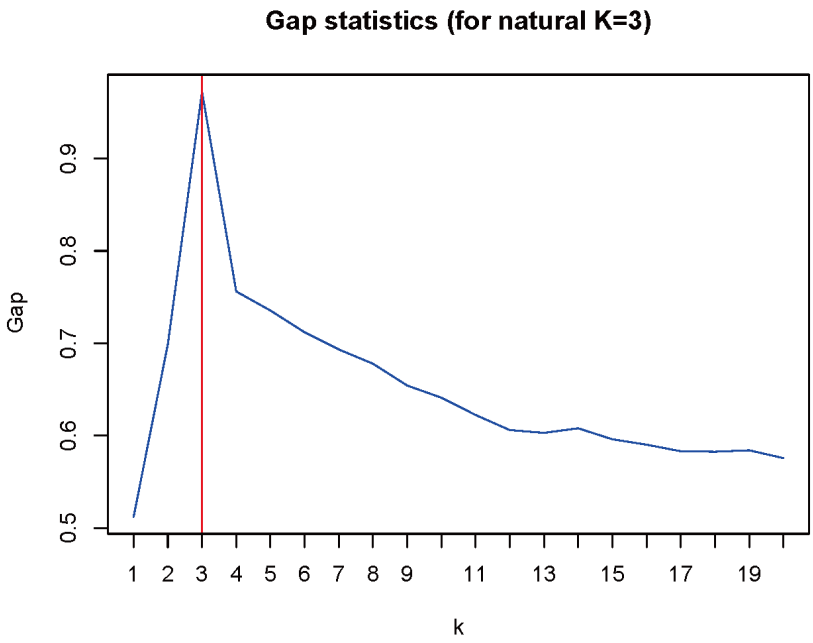

Figure 3.1: Gap statistics (with $K^* = 3$, test for K=1 to 20)

estimate determine K^* automatically.

Figure 3.1 shows gap statistics for simulated data sets (without batch effect) with true $K = 3$. We use K-means for cluster analysis. The Gap statistics for each K are mean value of 200 simulations.

In our simulations, the choosing criterion determine $K^* = 3$ for 100 % probability in the same 200 simulations. While for data sets with batch effect, the choosing criterion determine $K_* = 3$ with 32 % probability in another 200 simulations. We use our method instead to perform cluster analysis, and it turns out gap statistics are able to determine the true number of clusters every time in simulations.

Chapter 4

Test for identify batch effect

4.1 Multinomial test

In this section, we discuss a test for independence between batches and clusters using a multinomial test (or called a chi-squared test for independence). We perform standard K-means clustering on the data. After clustering, the number of observations n_{tk} assigned to the kth cluster in the tth batch is known. These numbers can be presented in Table 4.1, where X represents batch and Y represents the cluster.

Let $n_{t+} = \sum_{k=1}^{K} n_{tk}$, $n_{+K} = \sum_{t=1}^{T} n_{tk}$ and $N = \sum_{k=1}^{K} \sum_{t=1}^{T} n_{tk}$, where n_{tk} is the number of observations being assigned to the kth cluster in the tth batch, n_{t+} is the total number of observations in the tth batch, n_{+K} is the total number of observations being assigned to the kth cluster, and N is the total sample size. We suppose $\pi_{tk} = P(X = t, Y = k)$ and $(n_{11}, n_{12}, \ldots, n_{TK}) \sim Multinomial\left\{(\pi_{11}, \pi_{12}, \ldots, \pi_{TK}), N\right\}$. We call this a saturated model.

Assuming independence between batches and clusters (under our null hypothesis), we have

$$P(X = t, Y = k) = P(X = t)P(Y = k).$$

Table 4.1: Two-way contingency table of n_{tk}'s

	$Y=1$	$Y=2$	$\dots$	$Y=K$	Totals
$X=1$	n_{11}	n_{12}	$\dots$	n_{1K}	n_{1+}
$X=2$	n_{21}	n_{12}	$\dots$	n_{1K}	n_{2+}
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
$X=T$	n_{T1}	n_{T2}	$\dots$	n_{TK}	n_{T+}
Totals	n_{+1}	n_{+2}	$\dots$	n_{+K}	N

Denote $P(X=t)$ by α_t and $P(Y=k)$ by β_k so $\pi_{tk}=\alpha_t\beta_k$, under independence.

Here, α_t and β_k can be estimated through their sample frequencies

$$\widehat{\alpha}_t=\frac{n_{t+}}{N},\quad and\quad \widehat{\beta}_k=\frac{n_{+k}}{N}.$$

Under independence, the expected value of n_{tk} is

$$\widehat{\mu}_{tk}=N\widehat{\mu}_{tk}=N\widehat{\alpha}_t\widehat{\beta}_k=\frac{n_{t+}n_{+k}}{N}.$$

This gives the following chi-square test statistics of independence

$$\chi^2=\sum_{k=1}^{K}\sum_{t=1}^{T}\frac{(n_{tk}-\widehat{\mu}_{tk})^2}{\widehat{\mu}_{tk}},$$

where under the null hypothesis

$$\chi^2\sim\chi^2_{(K-1)(T-1)}.$$

The degrees of freedom of the test statistics equals the difference between the degrees of freedom in the saturated model and the model under independence. The batch memberships are fixed in both models, but the number of clusters are random. Thus the saturated model has $(K-1)T$ degrees of freedom, and the independence model has $K-1$ degrees of freedom.

If we reject the null, we conclude there is some association between clusters and batches, and that the adjustment for batch effects is needed.

4.2 MANOVA test

As we mentioned before, we see batch effects as a value add to all observations in each batch. We want to test whether this value is $\mathbf{0}$. Since the batch effect effects all observations in each batch the same way, it effects the centers of all the observations the same way. We calculate the center of the observations in each batch using the sample mean,

$$\boldsymbol{C}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \boldsymbol{y}_{it}.$$

Each observation follows a Gaussian distribution with mean $\boldsymbol{\theta}_k + \boldsymbol{\beta}_t$ and covariance matrix $\boldsymbol{\Sigma}_k$, thus $\boldsymbol{C}_t$ also follows a Gaussian distribution. Under the null hypothesis $\boldsymbol{\beta}_t = \mathbf{0}$, and assuming there's no interaction between batches and clusters, the proportion of each cluster in each batch should be consistent with prior probability π_k.

Thus the mean and covariance of $\boldsymbol{C}_t$ can be calculated as

$$\boldsymbol{C}_t \sim N(\sum_{1}^{K} \pi_k \boldsymbol{\theta}_k + \boldsymbol{\beta}_t, \frac{1}{n_t} \sum_{1}^{K} \pi_k \boldsymbol{\Sigma}_k).$$

Denote $\sum_1^K \pi_k \boldsymbol{\theta}_k$ by $\boldsymbol{\mu}$, the overall mean, and denote $\frac{1}{n_t} \sum_1^K \pi_k \boldsymbol{\Sigma}_k$ by $\boldsymbol{\Sigma}$, the overall variance. Note both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is independent of t. Thus, testing $\boldsymbol{\beta}_t = \mathbf{0}$ is a multivariate analysis of variance (MANOVA) test, where the batch is the only factor in the test. Notice this method is more powerful, but only effective when dealing with additive batch effect (which is our assumption in the thesis).

If we reject the null, meaning the values of batch effect are large and cannot be neglected, the adjustment for batch effect is needed.

Chapter 5

Alternative methods

In this chapter, we introduce the K-means clustering method and a procedure for removing the batch effect. We will compare the clustering results from our methods with results from K-means, and K-means with the batch effect removed prior to the analysis using the two-stage procedure described earlier.

5.1 K-means cluster analysis

K-means clustering, as a method of vector quantization, was invented in signal processing and is popular for cluster analysis in data mining. The purpose of K-means clustering is to partition the observations into K clusters so that each observation is closest to the "center" of the cluster it belongs to.

The algorithm for K-means is given below:

1. Initial "centroids"

Give the K initial points (do not have to be observations) as the centroids (center of each cluster).

2. Assignment step

Calculate the distances from each observation to all K centroids $D_{ik} = \|\boldsymbol{y}_i - \boldsymbol{c}_k\|^2$ (where $\boldsymbol{c}_k$ is the centroid of kth cluster), then assign $\boldsymbol{y}_i$ to the k^*th cluster such that

$$k^* = \underset{k}{argmin}\{D_{ik}\}.$$

3. Update step

Update the centroids using

$$\boldsymbol{c}_k = \frac{1}{N_k} \sum_{\boldsymbol{y}_i \in k} \boldsymbol{y}_i,$$

where N_k is the number of observations being assigned to the kth cluster.

4. Check convergence

If the criterion for convergence is meet (clustering result dose not change any more), output the clustering results, otherwise go back to step 2.

We use R build in function 'kmeans' to implement this algorithm.

5.2 Two-stage procedure for removing batch effect

A new method for removing batch effects was proposed by Giordan (2013). The model is based on the extension of the empirical Bayes method proposed by Johnson et al.(2007) with a different method for estimating the parameters. This method is designed for both supervised analysis (meaning we have other covariates of interest in the data) and unsupervised analysis (which is our case, batch is the only covariate we consider). We give the algorithm for removing batch effect in unsupervised analysis

only.

1. Set

$$\mathbf{Y}_1 = \mathbf{Y} - \mathbf{Z}\widehat{\mathbf{B}},$$

where $\widehat{\mathbf{B}} = \mathbf{Z}^{\dagger}\mathbf{Y}$ and $\mathbf{Z}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\mathbf{Z}$ ($\mathbf{Z}^{\dagger} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$ here). The definition of $\mathbf{Y}$ and $\mathbf{Z}$ are the same as in Chapter 2.

2. Set

$$\mathbf{Y}_2 = \mathbf{Y}_1 \circ \hat{\triangle}^{-1},$$

where $\circ$ denote the Hadamard product and $\hat{\triangle}^{-1}$ is a matrix with $\hat{\triangle}^{-1}(i,j) = 1/\hat{\delta}_{ij}$, where

$$\hat{\delta}^2_{ij} = \sum_{t=1}^{T} \mathbf{Z}(i,t)\hat{\mathbf{D}}(t,j),$$

and $\hat{\mathbf{D}} = \mathbf{Z}^{\dagger}\hat{\mathbf{E}}^2$, with $\hat{\mathbf{E}}^2 = \hat{\mathbf{E}} \circ \hat{\mathbf{E}}$ and $\hat{\mathbf{E}} = \mathbf{Y} - \mathbf{Z}\widehat{\mathbf{B}}$

3. Set

$$\mathbf{Y}_3 = \mathbf{Y}_2 \circ \hat{\triangle}_2,$$

where $\hat{\triangle}_2(i,j) = \sqrt{\hat{\sigma}^2_j}$ and

$$\hat{\sigma}^2_j = \frac{1}{N}\sum_{i=1}^{N} \hat{\delta}^2_{ij}.$$

4. Set

$$\mathbf{Y}_4 = \mathbf{Y}_3 + \frac{1}{N}\mathbf{1}\mathbf{Z}\widehat{\mathbf{B}},$$

where $\mathbf{1}$ is an $N \times N$ matrix of ones.

$\mathbf{Y}_4$ is the estimated batch effect. It will be removed form the observations. Bagging

technique can be applied to get better estimation.

We use R package 'ber' (Giordan, M. 2013) to implement this algorithm.

Chapter 6

Simulations

In this chapter, we first give a simulated data set to show the steps of applying the methods we mentioned earlier to a data set. Then we give comparisons between our method and other two methods in different situations (200 simulations for each situation).

6.1 Clustering comparison

In this section, we simulated two-dimensional data with batch effects, and 3 natural clusters. For the baseline simulation, we have 5 batches and 50 observations for each batch. For each batch, the observations are randomly assigned to 3 clusters with even probability, i.e., $\pi_k = 1/3$ for all k. Each cluster has their own Gaussian distribution $\boldsymbol{y}_{it(k)}^{'} \sim N(\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k)$, where $\boldsymbol{y}_{it(k)}^{'}$ is the observation without any batch effect. The parameters $\{\boldsymbol{\theta}_k, \boldsymbol{\Sigma}_k\}$ for the 3 multivariate Gaussian distributions are:

$$N\left((2, 0.8), \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix}\right), N\left((-0.5, 2), \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}\right), N\left((0.2, 0.5), \begin{pmatrix} 0.25 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}\right).$$

Batch effects are generated from a multivariate mean zero Gaussian distribution,

$$N\left((0,0),\begin{pmatrix}0.5 & 0\\ 0 & 0.5\end{pmatrix}\right).$$

The variance was chosen to set the ICC (intraclass correlation, which represents the proportion of batch variation out of total variation) at a various levels (here ICC=0.625). We assume no correlation between the two dimensions. The generated value $\boldsymbol{\beta}_t$ for each batch is added to all the observations in batch t.

In Figure 6.1, we give plots of data with and without batch effects. We assume that the variance is larger for biomarker 2, and there is correlation between biomarker 1 and biomarker 2. The estimated ICC is about 0.625.

We can see from Figure 6.1 that the observations without batch effects are nicely separated, so a typical clustering algorithm like K-means should be able to give accurate results. After adding the batch effect, however, the edges of clusters become overlapped, which makes the prediction of observations on the edges difficult.

We use Gap statistics to determine the number of clusters. Since our method estimates $\boldsymbol{\beta}_t$, we can directly remove the batch effect from observations. Then we calculate the $Gap(K)$ using batch effect removed data. We present the results of $Gap(K)$ and $S_K^{'}$ in Table 6.1. According to the choosing criterion, we use K=3 clusters in our analysis.

Next we perform the test to identify any possible batch effect. For the multinomial

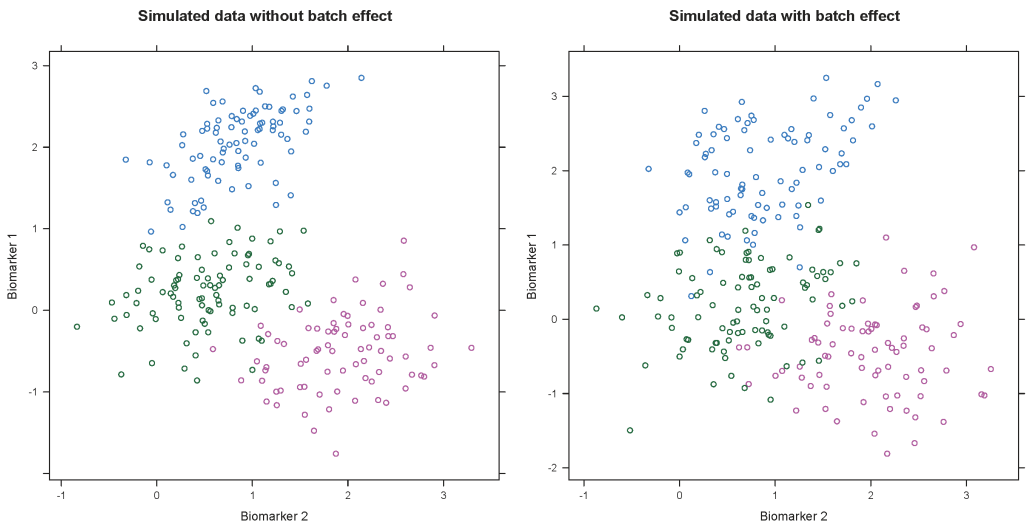

Figure 6.1: Comparison of observations with and without batch effect

Table 6.1: Gap statistics using choosing criterion

K	2	3	4
$Gap(K)$	0.7597289	1.049799	0.7188229
$S_K^{'}$	0.09433268	0.04744324	0.04263864

test, the contingency table is shown in Table 6.2. The p-value we get is 0.2524, not significant at the $\alpha = 0.05$ level.

For MANOVA test, the p-value for batch is $1.73 * 10^{-8}$, so we conclude there is batch effects, and we should apply our method to the data.

Next, we show the comparison of the clustering results from our method, K-means and K-means with batch effect removed prior to analysis in Figure 6.2. We add the original cluster as reference. Notice in the clustering results, we assign "similar" observations to the same cluster, but the clusters themselves don't have orders. We focus on whether observations originally from the same cluster are assigned to the same cluster by the algorithm.

Table 6.2: Table of clusters and batches

	Y=1	Y=2	Y=3	Totals
X=1	10	15	25	50
X=2	18	15	17	50
X=3	15	17	18	50
X=4	19	13	18	50
X=5	15	22	13	50
Totals	77	82	91	250

In Figure 6.2, we display the results from a single simulation. Notice that our method and K-means with batch effect pre-removed have better prediction on observations on the edges of clusters, especially where two clusters partially overlapped.

We use the accurate rate to quantify the clustering results. The definition of the accurate rate is the portion the observations being correctly clustered.

$$AR = \frac{N_c}{N},$$

where N_c is the number of observations being correctly clustered, and N is the total sample size. An observation is determined to be correctly clustered if its clustering assignment is to the cluster that contains the most observations from its true cluster.

For this baseline data set, the ARs for our method, K-means, K-means with batch effect pre-removed are 0.944, 0.900 and 0.932, respectively.

We ran 200 simulations using these three methods on data sets without batch effects.

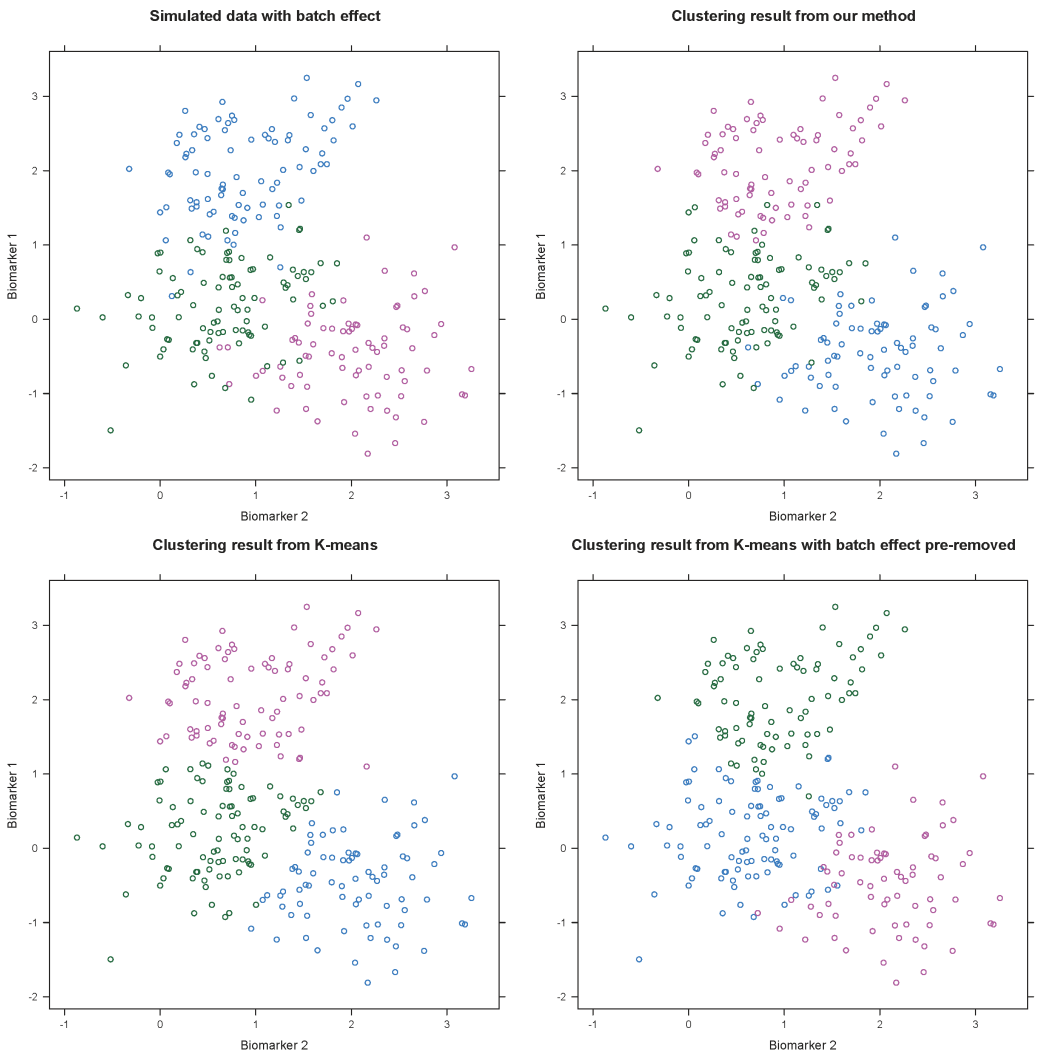

Figure 6.2: Comparison of results from three clustering methods, observations being assigned to the same cluster are in the same color

The results show there is no harm applying our method on data sets without batch effects (estimated $\boldsymbol{\beta}_t$ are close to $\mathbf{0}$). The ARs (followed by standard deviations of ARs in parenthesis) for our method, K-means, K-means with batch effect pre-removed are 0.952(0.017), 0.944(0.015) and 0.934(0.018), respectively. If our target is to cluster the data, we don't need to worry about the existence of batch effects; we can apply our method directly.

Table 6.3: ARs (followed by standard deviations of ARs in parenthesis) of clustering results with increase of ICC

ICC	0.625	0.667	0.727
AR of our method	0.948(0.022)	0.949(0.022)	0.947(0.029)
AR of K-means	0.853(0.053)	0.821(0.073)	0.737(0.096)
AR of K-means batch effect removed	0.934(0.019)	0.936(0.016)	0.934(0.019)

6.1.1 Clustering comparison by ICC

The higher the ICC the larger batch effect. As a result, we explore the influence of higher ICC on clustering results. Again we compare results from the three methods discussed earlier. We use the same simulation setting as the baseline data set, only altering the covariance matrix of the batch effect to set the ICC.

In Table 6.3, we represent the ARs. We can see that with an increase in ICC, K-means loses accuracy in its prediction, while the accuracies of our method and K-means with batch effect pre-removed remain (in fact they appear immune to the increase of ICC). This shows the necessity of adjusting for the batch effect, especially when the values of batch effects are relatively large.

6.1.2 Clustering comparison by sample size of each batch

We know that larger sample sizes in each batch lead to better estimates of the batch effect. In this section we want to test if the methods are consistent when there are fewer observations in each batch. We use the same simulation setting as the baseline data set, only altering the number of observations in each batch.

Table 6.4: ARs (followed by standard deviations of ARs in parenthesis) of clustering results with decrease of sample size

Sample size in each batch	50	30	20	10
AR of our method	0.948(0.022)	0.949(0.022)	0.925(0.049)	0.883(0.078)
AR of K-means	0.853(0.053)	0.821(0.073)	0.853(0.061)	0.838(0.076)
AR of K-means with batch effect removed	0.934(0.019)	0.936(0.016)	0.914(0.034)	0.877(0.068)

In Table 6.4, we present the ARs with the number of observations in each batch decreasing. All three methods are consistent when the sample size of each batch is greater than 30, no sign of accuracy rates decreasing appears. However, when the sample size of each batch is small (for example, 10), we can see obvious decrease in the AR for all three methods. We therefore recommend apply these methods to data sets with sample size at least 20 for each batch.

6.1.3 Clustering comparison by the number of batches

As we discussed earlier, the batch effect can come from many sources, like different time periods, samples, labs, etc. There can be many batches in one data set, so we explore the influence of the number of batches on the clustering results. We use the same simulation setting as the baseline data set, and only altering the number of batches.

In Table 6.5 we present the ARs of the three methods by batch size. We can see the number of batches doesn't influence the ARs of these three methods. With the increase of total sample size, these three methods gain precision (smaller variance of

Table 6.5: ARs (followed by standard deviations of ARs in parenthesis) of clustering results with increase of batch size

Number of batches	5	10	20
AR of our method	0.948(0.022)	0.958(0.011)	0.959(0.007)
AR of K-means	0.853(0.053)	0.854(0.032)	0.855(0.021)
AR of K-means with batch effect removed	0.934(0.019)	0.936(0.011)	0.935(0.009)

ARs).

6.1.4 Clustering with an interaction between batches and clusters

In the previous results our method has resulted in modest increases the accuracy rate compared to K-means with batch effect removed prior to analysis. A situation where there is noticeable improvement is when there is an interaction between batches and clusters. By interaction we mean that in some batches the proportion of each cluster is not consistent with the proportion in the total sample. For example, in our case the proportion of each cluster is 1/3, if in some batches there are much more observations from cluster 1, there is an interaction. In this situation when we remove the batch effect from the data, we remove part of the effect of cluster 1 in those batches. This could lead to very biased results. This kind of interaction occurs in the real life for two main reasons: when the sample size of each batch is small or the clusters and batches are correlated some way, for example, some biomarkers we are measuring are correlated with the locations the subjects live in. This could result in an interaction.

In this part, our simulated data comes from 5 batches, and 50 observations per batch

(ICC=0.625). We assign observations to three clusters randomly in batch 1 and batch 2. In batch 3, we assign observations to cluster 1 and cluster 2 randomly. In batch 4, we assign observations to cluster 2 and cluster 3 randomly. In batch 5, we assign observations to cluster 1 and cluster 3 randomly. The total number of each cluster will remain approximately equal, while there is an interaction between clusters and batch 3, 4 and 5.

In this simulation the AR for our method is 0.935(0.043); the AR for K-means is 0.859(0.053); while the AR for K-means with batch effect removed prior to analysis is only 0.805(0.035). Our method is immune to an interaction like this.

Chapter 7

Conclusions

Our simulation studies suggest that our method has better prediction results than K-means, and K-means with batch effect removed prior to the analysis. K-means tends to produce lower accuracy rates when the batch effects have relatively large variances. K-means with batch effect removed prior to the analysis fails when there is an interaction between batches and clusters. Since we have set up tests to identify the batch effect, we apply this test to the data first. If any batch effect is detected, we can choose our method to do clustering, and the Gap statistics can be used to determine the number of clusters.

Our method does not require the batch effect following any particular distribution. Further it works on data that has small sample size, data that has higher batch effect, data that has an interaction between clusters and batches, and data that has many batches.

We only test our method on bivariate data; further simulation studies on multivariate data can be done to find the effectiveness of our method on higher dimensional data. Our method models only additive batch effects, the simulation studies use only

additive batch effect and we are not sure our method can be applied to other type of batch effects. In the future, we can introduce scalars into our model to model the scaled variance in each batch.

Bibliography

Bishop, C. M. et al. (2006), *Pattern recognition and machine learning*, vol. 4, springer New York.

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)*, 1–38.

Giordan, M. (2013), *ber: Batch Effects Removal*, r package version 4.0.

— (2014), "A two-stage procedure for the removal of batch effects in microarray studies," *Statistics in Biosciences*, 6, 73–84.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R. (2009), *The elements of statistical learning*, vol. 2, Springer.

Johnson, W. E., Li, C., and Rabinovic, A. (2007), "Adjusting batch effects in microarray expression data using empirical Bayes methods," *Biostatistics*, 8, 118–127.

Lander, E. S. (1999), "Array of hope," *Nature genetics*, 21, 3–4.

Leek, J. T., Scharpf, R. B., Bravo, H. C., Simcha, D., Langmead, B., Johnson, W. E., Geman, D., Baggerly, K., and Irizarry, R. A. (2010), "Tackling the widespread and critical impact of batch effects in high-throughput data," *Nature Reviews Genetics*, 11, 733–739.

Louis, B., Germaine, M., Hediger, M. L., Bell, E. M., Kus, C. A., Sundaram, R., McLain, A. C., Yeung, E., Hills, E. A., Thoma, M. E., et al. (2014), "Methodology for Establishing a Population-Based Birth Cohort Focusing on Couple Fertility and Children's Development, the Upstate KIDS Study," *Paediatric and perinatal epidemiology*, 28, 191–202.

McLachlan, G. J. and Basford, K. E. (1988), "Mixture models: Inference and applications to clustering," *Applied Statistics*.

McLachlan, G. J. and Krishnan, T. (1997), "Wiley series in probability and statistics," *The EM Algorithm and Extensions, Second Edition*, 361–369.

Peel, D. and McLachlan, G. J. (2000), "Robust mixture modelling using the t distribution," *Statistics and computing*, 10, 339–348.

Rhodes, D. R., Yu, J., Shanker, K., Deshpande, N., Varambally, R., Ghosh, D., Barrette, T., Pander, A., and Chinnaiyan, A. M. (2004), "ONCOMINE: a cancer microarray database and integrated data-mining platform," *Neoplasia*, 6, 1–6.

Schadt, E. E., Li, C., Ellis, B., and Wong, W. H. (2001), "Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data," *Journal of Cellular Biochemistry*, 84, 120–125.

Schutt, R. and O'Neil, C. (2013), *Doing data science: Straight talk from the frontline*, O'Reilly Media, Inc.

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, 411–423.

Tseng, S.-Y., Otsuji, M., Gorski, K., Huang, X., Slansky, J. E., Pai, S. I., Shalabi, A., Shin, T., Pardoll, D. M., and Tsuchiya, H. (2001), "B7-DC, a new dendritic cell molecule with potent costimulatory properties for T cells," *The Journal of experimental medicine*, 193, 839–846.

Van Ryzin, J. (2014), *Classification and Clustering: Proceedings of an Advanced Seminar Conducted by the Mathematics Research Center, the University of Wisconsin at Madison, May 3–5, 1976*, Elsevier.

Yang, P., Yan, H., Mao, S., Russo, R., Johnson, J., Saykally, R., Morris, N., Pham, J., He, R., and Choi, H.-J. (2002), "Controlled growth of ZnO nanowires and their optical properties," *Advanced Functional Materials*, 12, 323.