Classification Based on Supervised Clustering with Application to Juvenile Idiopathic Arthritis

by

Yuanyu Yang

B.Sc., China Agricultural University, 2010

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY

Summer 2013

Approval

Name: **Yuanyu Yang**

Degree: **Master of Science**

Title of Project: ***Classification Based on Supervised Clustering with Application to Juvenile Idiopathic Arthritis***

Examining Committee: **Chair:** Tim Swartz
Professor

Thomas M. Loughin
Senior Supervisor
Professor

Carl James Schwarz
Supervisor
Professor

Jaime Guzman
Supervisor
Assistant Professor

Qian Zhou
Examiner
Assistant Professor

Date Defended/Approved: August 16, 2013

Partial Copyright Licence

Abstract

Juvenile Idiopathic Arthritis (JIA) is the most common rheumatic disease of childhood. Our objective is to predict the results of remission so that those children who are likely to experience poor remission outcomes could benefit from early aggressive treatment. Many classification techniques could provide either a binary prediction or an estimated probability of remission. However, parents would like to know more specifically about the remission outcomes of children similar to their own. In this project, we propose a supervised clustering method that provides this information.

Inspired by the basic idea of supervised principal component analysis, we perform supervision by selecting and/or weighting explanatory variables differently depending on their associations with the class response. Our supervised clustering method is applied to JIA data and to data simulated with known properties. Our method is shown to be competitive with an existing supervised clustering method, classification trees and random forests in terms of out-of-sample misclassification rates.

Keywords: Variable weighting; logistic regression; categorical; classification tree; random forest; SRIDHCR

Acknowledgements

First and foremost, I would like to express my deepest gratitude and appreciation to my senior supervisor, Dr. Thomas M. Loughin for his helpful guidance, deep understanding and been a good mentor throughout my graduate studies. He is a funny man in life but very professional in academic. This project would not have been possible without his support and encouragement. I would also like to thank the members of my examining members, Carl Schwarz, Jaime Guzman, and Qian Zhou for patient review and useful feedback for my project.

In addition, I would like to acknowledge all the professors in the department of Statistics and Actuarial Science for helping me obtain solid knowledge in statistics. I would also like to thank all my friends and fellow graduate students. With your supports and friendship, the study in SFU became more enjoyable.

Last but not least, I would like to thank my dearest family for their love, understanding and encouragement.

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Overview of Juvenile Idiopathic Arthritis

Juvenile Idiopathic Arthritis (JIA) is the most common rheumatic disease of childhood (Hashkes and Laxer 2005). JIA is a chronic disease characterized by persistent joint inflammation and causes much disability. "Juvenile" points out that the symptoms appear before the patient is aged 16 years, and "Idiopathic" means the causes of disease are unknown. JIA affects about 80-90 per 100,000 children. It consists of several subtypes, and the chances of remission vary widely depending on the subtype.

Treatment options for children with JIA have increased dramatically in recent decades, but along with these options come different risks and side effects. Parents need to decide whether to take a conservative, less risky approach to treatment, or be more aggressive, but with greater risk. Unfortunately, most parents have little specific knowledge of the expected course of the disease, treatment response and risk of side effects to their child. For parents to make treatment decisions better on newly diagnosed children, it would be helpful to provide them with their child's predicted chance of remission so that those children who are likely to experience poor remission outcome could benefit from early aggressive treatment.

1.2. Objective

In this research, several Canadian Institutions of Health Research (CIHR)—funded JIA cohorts, such as the Research in Arthritis in Canadian Children (ReACCh) and the Biologically-Based Outcome Predictors in JIA (BBOP), collected data on disease status and demographic information on recently diagnosed patients. This project will focus on the disease remission, since it is the top priority outcome chosen by patients'

families. Our objective is predicting whether a child will experience poor or good outcomes, as measured according to whether the JIA is in remission. This is a classification question.

There are many classification techniques that can provide predicted outcomes for classification problems (Hastie, Tibshirani, and Friedman 2009). However, these typically provide either a binary prediction or an estimated probability of remission. It may be easier for parents to understand statements like, "your child is similar to _ other children, _ % of whom experienced remission when treated with _." A classification tree (Breiman, 1984) is the well-established classification technique that creates predictions by grouping subjects in this manner. However, classification trees are notoriously unstable and can produce a totally different tree with just a small change in data.

Alternatively, we could treat this problem as a clustering question, in which a child belongs to a cluster of similar children based on his/her explanatory variables. Then we could estimate the probability of remission in this cluster. Unfortunately, since clustering is unsupervised, explanatory variables that contribute most to forming clusters may not be highly related to the outcome.

To solve this problem, we develop a supervised approach that makes use of a measured outcome to "guide" clustering. The new supervised clustering technique we are developing is similar to supervised principal component analysis (Bair et al. 2006). We select those variables that have relative high associations with the outcome, and apply a clustering algorithm to these selected variables. Optionally, we can weight variables' influence on the clustering according to the strength of their association with the outcome. We expect our supervised clustering method will have competitive performance for predicting outcomes compared to classification trees, and also provide more stable results.

1.3. Outline

This project is organized as follows. In Chapter 2, we review some basic concepts of classification and clustering, and introduce an existing supervised clustering algorithm. Chapter 3 mainly discusses the framework of our supervised clustering

method and how to choose tuning parameters. In Chapter 4, we compare our method with the existing supervised clustering method, a classification tree, and a random forest on the JIA data. We also try these methods on other data sets with multiple class response (i.e., not binary). In Chapter 5, simulation studies based on the JIA data are presented and discussed. Chapter 6 concludes the whole project and suggests future work.

2. Review of Classification and Clustering

In machine learning, there are two types of problems distinguished by whether there is an output variable in the data: supervised learning and unsupervised learning (Hastie, Tibshirani, and Friedman 2009). A supervised learning algorithm analyzes training data and produces a function of explanatory variables to predict the output variable. The type of output variable leads to two distinct problems: regression is used for predicting numerical output, while classification is used for predicting categorical output. In unsupervised learning, there is no target variable that we are trying to predict. Instead, we try to explore the hidden structure of explanatory variables, such as searching for clusters or reducing dimensionality (e.g., principal component analysis). In this chapter, the basic concepts and common methods of classification and clustering are described. Moreover, a "supervised clustering" method is introduced that uses an output variable to help inform the clustering process.

2.1. Classification

In a classification problem, the population is divided into c groups, called "classes", and the goal is to identify any observation's class based on some explanatory variables. A sample is taken, consisting of observations (also called "items") whose class memberships are known. A rule or function that implements classification is fit to the data and is known as a classifier. There are many attributes that are desired in a would-be classifier, such as accuracy, speed, and comprehensibility. Accuracy is generally the most essential concern at least in small or moderately sized data sets. The most widely used measure of accuracy is misclassification rate:

$$\text{misclassification rate} = \frac{\text{the number of misclassified observations}}{\text{the total number of observations}},$$

where an observation is “misclassified” if its predicted class does not match its known class.

There are several well-established classification procedures, such as linear discriminant analysis, quadratic discriminant analysis, and logistic regression, which are still used as staple solutions for classification (Hastie, Tibshirani, and Friedman 2009). All of these methods assume that either the form of the underlying joint density of explanatory variables is known, or the exact relationship between the response and explanatory variables is known except for the values of some parameters. However, in real problems, this assumption does not always hold. In particular, real applications often consist of a mix of numerical and categorical variables. Therefore, some nonparametric (also called distribution-free) classification procedures have been introduced, which can be used without assuming the form of joint density of explanatory variables. Examples include K-nearest neighbour, naïve Bayes, neural networks, and classification trees (Hastie, Tibshirani, and Friedman 2009). Among these methods, the classification tree is commonly used, especially when we need to explain the results to non-statisticians, due to the easy interpretability of its structure.

A popular tree-based method is classification and regression trees (CART), which was introduced by Breiman (1984). Recursive partitioning is the key to finding a decision tree in CART. It is a recursive process of splitting each subset of data, called a “node”, into two offspring nodes. The starting point of a classification tree is called the root node, consisting of all the observations in the data. A split is determined by a condition on the value of a single variable that improves misclassification rate the most. Observations whose variable satisfies the condition are grouped into one offspring node, while the remaining observations are grouped into the other offspring node. A node that is split into two offspring nodes is called a nonterminal node. When a node is no longer split, it is called a terminal node. Each observation falls into a particular terminal node in the end. Theoretically, we can keep splitting a tree until each terminal node contains a single observation. But usually, a lower limit on the number of observations in a node is applied to stop splitting and prevent over-fitting. Sometimes, the tree is still too large even when the stopping rule is applied. Then some branches of the tree are pruned according to how much they can improve the prediction of classes per added node. After pruning, the observations in each class are counted at every terminal node, and

the most common class in the node is assigned to be the predicted value for that node. As an illustration of a tree structure, we apply classification tree to Pima Indians Diabetes data set (http://astro.temple.edu/ ~alan/pima.indians.diabetes3.txt). The graph of the classification tree is in figure 2.1.

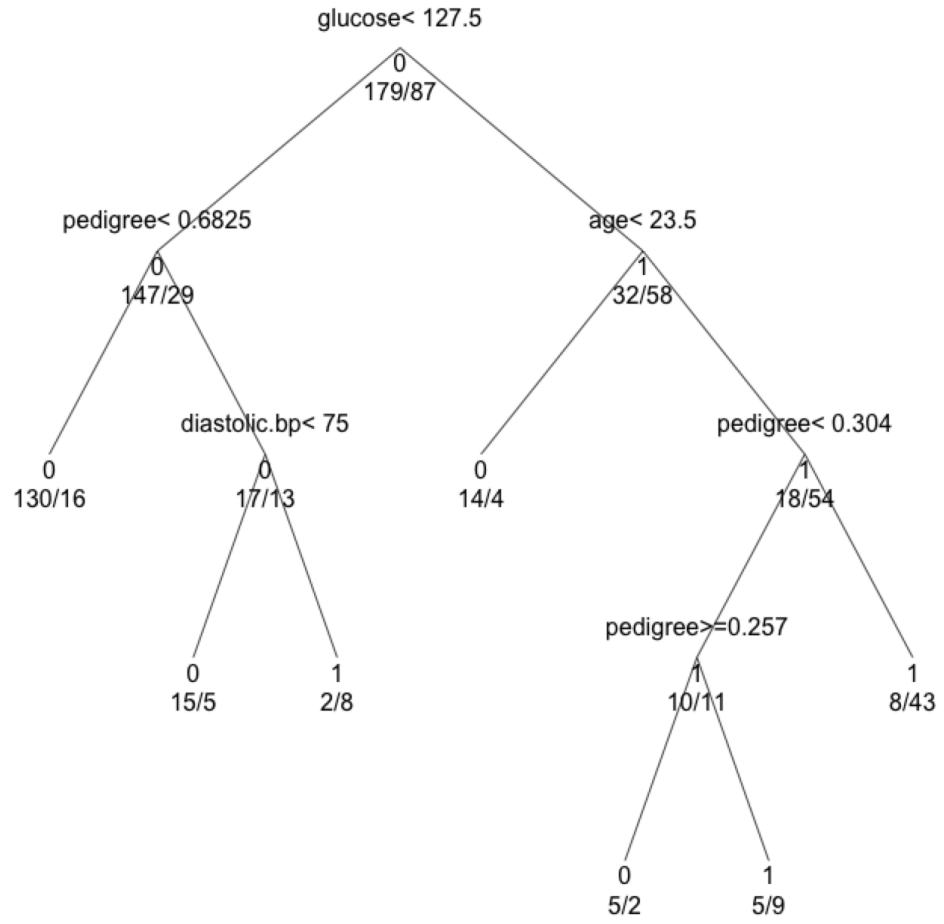

Figure 2.1 ***Classification tree for predicting diabetes status of 266 Pima Indians: 0 for normal and 1 for diabetes. At each node, there is an associated question on an explanatory variable. An observation is assigned to the left branch if the answer is "yes", to the right branch if the answer is "no". Finally, the majority class is assigned as the class label of each terminal node. For example, the left-most terminal node consists of subjects with glucose < 127.5 and pedigree < 0.6825. There were 130 subjects in class "0", 16 in class "1", so the node is assigned to class "0"***

There are several advantages of classification trees. First, they usually have good comprehensibility. If a tree is not too large, the decision tree graph gives us a visual explanation of the results. It is very easy to interpret, even for non-statisticians. Also, it is applicable for both continuous and categorical explanatory variables. These features make it an appealing candidate technique for our JIA problem since there are many categorical variables in the JIA data and the goal is to produce a classifier that parents can understand.

Even though classification trees provide us with easily interpreted decision tree structures, there are still some drawbacks of this method. First, even a small change in the data can at times cause large changes in the tree. For example, adding or deleting one observation can change the variables or locations of splits on a node, including the root node, thereby affecting all subsequent splits involving those data. Also, sometimes a tree might get too large even with some pruning. A complex tree structure is no longer easy to understand.

Random forest (Hastie, Tibshirani, and Friedman 2009) is a useful algorithm to reduce the instability of a classification tree by constructing a large number of trees and combining their predictions. Suppose there are n items in the data and each of them contains p explanatory variables. Before constructing a classification tree, n items are randomly sampled from the data with replacement. This "resample" becomes the data set on which the tree is grown. Any unsampled items, called out-of-bag (OOB) data, are used to test how well the tree does with classifying. Instead of selecting a best split from all p explanatory variables at each node, a random forest grows trees by randomly selecting $m < p$ explanatory variables as candidate variables at each node. After a classification tree is built, items in the OOB data are assigned to predicted classes based on the tree. This process is repeated a large number of times. In each repetition, a different set of OOB data are predicted by the tree grown on a different resample. Thus, a given item gets predicted numerous times by trees in different repetitions. In the end, each item is assigned to its most frequently predicted class. While a random forest often decreases the variability of a classification tree considerably, it cannot give us an easily interpreted tree structure anymore.

2.2. Clustering

2.2.1. General concepts

The task of clustering is to gather items into different groups called “clusters”, such that items in the same cluster are more similar to each other than to those in other clusters. The goal of clustering looks similar to that of classification, but there is a major difference. Clustering is unsupervised—we do not have a measured response that identifies the group to which an observation belongs. Instead, methods of clustering items depend upon how similar the items are to each other. Similar items are assigned to the same cluster, while remaining items form additional clusters. There is no predetermined number of clusters into which a population should be partitioned.

The criterion of measuring how far apart two items are is called dissimilarity. Let $x_i = (x_{i1}, \dots, x_{ip})^T$ and $x_j = (x_{j1}, \dots, x_{jp})^T$ be sets of p explanatory variables on items i and j, respectively. The dissimilarity between x_i and x_j, $d(x_i, x_j)$, usually satisfies the following three properties:

1. $d(x_i, x_j) \geq 0$
2. $d(x_i, x_i) = 0$
3. $d(x_i, x_j) = d(x_j, x_i)$

There are several ways to measure dissimilarity. The most commonly used is Euclidean distance:

$$d(x_i, x_j) = \left[\sum_{k=1}^{p} (x_{ik} - x_{jk})^2\right]^{1/2}.$$

In this paper, we use Euclidean distance as the default measurement for dissimilarity.

Intuitively, explanatory variables with relatively large variation have greater contribution to the distance measurement than those with less variability, thus they greatly influence the clustering. In order to let every explanatory variable make equal contribution to the distance measurement, we should first standardize all the variables to have equal variance (i.e., 1). Generally they are also centered to have mean zero although this is not necessary.

There are two basic types of clustering algorithms: hierarchical and non-hierarchical.

2.2.2. Hierarchical clustering

Strategies for hierarchical clustering consist of two types: agglomerative and divisive. Agglomerative clustering algorithms start by treating each item as its own cluster. Two clusters are combined into one larger cluster in successive steps until all items are in one cluster. Divisive clustering algorithms do the opposite, performing like a tree. At beginning, they treat all items as members of a single cluster; then recursively split one of the existing clusters into two new clusters until each item is its own cluster.

In agglomerative algorithms, merging occurs between the two clusters with the smallest between-cluster dissimilarity, whereas in divisive algorithms, splitting occurs between the two clusters with the largest between-cluster dissimilarity. There are several possible ways to measure distance between clusters that contain more than one observation, such as single linkage, complete linkage, and average linkage. There are also methods for selecting an appropriate number of clusters. See Izenman (2008) for details.

2.2.3. Non-hierarchical clustering

Non-hierarchical clustering methods, also called partitioning methods, predetermine the number of clusters, K, and find some "optimal" partitioning of the data into K clusters. There are several popular non-hierarchical clustering methods, such as K-means, K-medoids, and PAM. Next, we discuss the most popular method, K-means, which will also be used in our supervised clustering algorithm in Chapter 3.

The framework of the K-means algorithm is as follows:

1. Determine a K, and choose K items in the data to be the initial "centers" for the algorithm.
2. Assign each item to a cluster according to which of the centers is nearest (distance is usually computed using Euclidean dissimilarity).

3. For each cluster created in Step 2, compute its current p-dimensional mean point (called a centroid).
4. Re-assign each item to the cluster whose centroid is nearest, and recalculate the centroids for each cluster.
5. Repeat Step 4 until no more items change clusters.

The final assignment to clusters may vary with different initial points. Therefore the K-means algorithm is run several times, and the final assignment chosen is the one with minimum sum of within-cluster distance:

$$SWD = \sum_{k=1}^{K} \sum_{i:C(i)=k} d(x_i, \bar{x}_k)$$

where $\bar{x}_k$ is the centroid of cluster k, and $C(i)$ is the cluster assignment for x_i.

2.3. Existing supervised clustering methods

The objective of classification is to predict the class of new observations, while clustering assigns observations into groups based on the underlying structure of explanatory variables. They have different objectives, but both classification and clustering assign data into groups. We may wonder how well a clustering method would work if we applied it to a classification problem. Since there is now a class response helping to determine the clustering, the clustering algorithm becomes supervised. The objective of supervised clustering is to produce clusters so that most of observations within a cluster are from the same class. Eick et al. (2004) developed some algorithms for supervised clustering, among which "single representative insertion/deletion steepest decent hill climbing with randomized restart" (SRIDHCR) was recommended on the basis of prediction accuracy and algorithm efficiency on several data sets. Thus, we will use SRIDHCR as an alternative method to compare with our proposed supervised clustering method. The detailed structure of SRIDHCR is explained next.

In the dataset, the number of observations is n, and the number of classes in the outcome measure is c. Most often, $c = 2$. Suppose we group the observations into K clusters, and C is a clustering solution mapping n observations into K clusters (i.e., assigning a number from $1,\ldots,K$ to the n observations). The class with the highest

frequency in a cluster is called the "majority class". All other classes in that cluster are "minority classes". Observations in one of the minority classes are called "minority examples". For any cluster, the majority class is the predicted class for all items in that cluster.

In supervised clustering, the following two quantities are critical:

- Class impurity, measured by the percentage of minority examples in the different clusters of a clustering (note that this is also the misclassification rate defined earlier);
- Number of clusters, K. Although we can obtain a pure clustering (i.e., no impurity) by assigning observations into n clusters $(K{=}n)$, this is meaningless in practical problems because it gives us no information about the similarities among observations and leads artificially to perfectly pure clusters. In general, we like to keep the number of clusters low to enhance interpretation of the clusters.

In particular, Eick et al. (2004) use the following fitness function measured on a clustering:

$$q(C) = Impurity(C) + \beta * Penalty(K)$$

where $Impurity(C) = \{\#of\ Minority\ Examples\ in\ clustering\ C\}/n,$

$$Penalty(K) = \begin{cases} \sqrt{\frac{K-c}{n}}, K \geq c \\ 0, K < c \end{cases},$$

and $\beta(0 < \beta \leq 2.0)$ determines the impact of the penalty for the number of clusters. No rigorous justification was given in the paper regarding choice of the penalty coefficient β.

Representative-based clustering aims to find a set of K representatives (i.e., items) that best characterize clusters in a dataset. Clusters are created by assigning each object to the closest representative in a manner similar to K-means clustering. Representative-based supervised clustering algorithms seek to accomplish the following goal: find a set of representatives such that the clustering C obtained by using these representatives minimizes $q(C)$.

In particular, the "single representative insertion/deletion steepest decent hill climbing with randomized restart" (SRIDHCR) algorithm is as follows:

1. Randomly select a number of observations from the dataset as the initial set of representatives, called REP.
2. Create an initial clustering, C_0, by assigning all remaining observations to the cluster of their closest representative within REP, and compute the fitness function $q(C_0)$.
3. For observation $i = 1, \dots, n$, obtain a new set of representatives, REP_i, by either adding observation i to REP if $i \notin REP$, or removing observation i from REP if $i \in REP$. For each REP_i, create the clustering C_i and compute the fitness function $q(C_i)$. If $q(C_0) < q(C_i)$ for all i, the algorithm terminates. Otherwise it continues to Step 4.
4. For i': $q(C_{i'}) = min_i\{q(C_i)\}$, set $C_0 = C_{i'}$, $REP = REP_{i'}$. Then repeat Step 3.

Like K-means clustering, this process can be sensitive to the choice of initial points in REP. Usually, we run this process several times, and report the best clustering with minimum value of $q(C_0)$.

3. Our Supervised Clustering Method

When developing a supervised clustering method, a challenging issue is how to use the target class response to supervise the clustering algorithm. In the representative-based supervised clustering algorithms described in Section 2.3, supervision is imposed using the impurity of a clustering to guide the choice of clustering directly. Inspired by the basic idea of supervised principal component analysis (Bair et al. 2006), we propose a new supervised clustering method in which supervision is performed by weighting the explanatory variables differently, depending on their univariate associations with the class response. In this chapter, supervised principal component analysis is introduced and the framework of our supervised clustering method is described in detail.

3.1. Supervised Principal Component Analysis

In regression modeling, we sometimes face high-dimensional data in which the number of explanatory variables is much more than the number of observations. When there are many highly correlated variables in the data, then the actual information contained in the data can often largely be explained in a lower-dimensional subspace. Principal component analysis (PCA) is a popular dimension-reducing technique that seeks to project the data onto a lower-dimensional subspace indexed by "principal components" without losing important information. Often, most of the variability of data can be accounted for by the first few principal components. Following a PCA, regression can be carried out using the first few principal components as the explanatory variables. However, this does not always work well.

Since PCA is based only on the internal structure of the explanatory variables, there is no guarantee that the principal components with largest variation are also good predictors of the response variable. If they are not, then the regression based on the

first few principal components does not predict responses well. In order to solve this problem, Bair et al. (2006) presented a revised version of PCA, called supervised principal component analysis (SPCA). Here is the basic structure of SPCA:

1. Compute univariate linear regression coefficients for the response on each variable separately.
2. Select those variables whose absolute value of regression coefficient is greater than a threshold θ, and use these variables to form a new reduced data matrix.
3. Perform PCA on the new reduced data matrix.
4. Use the first few principal components in a regression model to predict the response.

Since SPCA identifies variables with high correlations with the response prior to performing PCA, the first few principal components are more likely to have strong correlation with the response variable than without screening. Therefore, SPCA often predicts the response better than PCA does.

Conventional clustering algorithms gather items into clusters only according to explanatory variables, which is similar to how PCA is normally conducted. The items within each cluster have more similar characteristics to one another than to items in other clusters. However, when there is a class response, these similar items within each cluster may not be in the same class because clustering algorithms do not use the information about the class response. In our new supervised clustering method, we use the class response to supervise the clustering algorithm, in a process similar to SPCA: choose those explanatory variables highly associated to the class response to conduct a conventional clustering algorithm.

3.2. Proposed Supervised Clustering Method

SPCA works only when all explanatory variables are continuous. In our algorithm, we improve the first step of the SPCA algorithm so that it can be applied to data with both continuous and categorical explanatory variables, both of which are present in the JIA data.

The framework of our proposed supervised clustering method is as follows:

1. Fit a logistic regression of the class response on each explanatory variable separately, and use likelihood ratio tests to measure the associations between each explanatory variable and the response.
2. Select those explanatory variables with p-value from the likelihood ratio test smaller than a threshold ρ, forming new data set X^*.
3. Manipulate X^* so that they are in appropriate form for a clustering algorithm. Create dummy variables to replace categorical variables, and (optionally) weight variables differently according to their associations with response.
4. Perform K-means algorithm and assign class labels based on the majority class within each cluster.

More detailed information on each step is described in this section.

3.2.1. Association between explanatory variables and response

Logistic regression is a popular regression analysis tool used for predicting the outcome of a categorical dependent variable according to one or more explanatory variables. Usually, "logistic regression" refers specifically to the problem in which the dependent variable is binary, such as our JIA problem. For a binary response *Y* and vector of explanatory variables $X = (x_1, x_2, \dots, x_p)$, let $\pi(x) = P(Y = 1|X = x)$. The logistic regression model is

$$logit(\pi(x)) = log\left[\frac{\pi(x)}{1-\pi(x)}\right] = \alpha + x\beta$$

where α is the intercept and β is a $p\times1$ vector of regression coefficients. Equivalently,

$$\pi(x) = \frac{exp(\alpha + x\beta)}{1 + exp(\alpha + x\beta)}$$

Categorical explanatory variables need to be converted into numerical values prior to fitting the logistic regression model. For instance, suppose v is a categorical variable with m levels. Then we create m-1 dummy variables

$$x_{ij} = \begin{cases} 1, if\ v_i = jth\ level \\ 0, otherwise \end{cases}, i = 1, \dots, n;\ j = 1, \dots, m-1$$

Logistic regression can be generalized to deal with a multicategory dependent variable. This is called multinomial logistic regression. Let *Y* be a categorical response with *J* categories, and $\pi_k(x) = P(Y = k|X = x)$ with $\sum_{k=1}^{J} \pi_k = 1$. Logistic models pair each response category with a baseline category, often the most common one or the first or the last one. If the last category is chosen as the baseline, then the *J-1* logistic regression models are

$$log\left[\frac{\pi_k(x)}{\pi_J(x)}\right] = \alpha_k + x\beta_k, \qquad k = 1, \dots, J-1$$

where α_k and β_k are intercept and vector of regression coefficients corresponding to response *k*, respectively. Equivalently,

$$\pi_k(x) = \frac{exp(\alpha_k + x\beta_k)}{1 + \sum_{l=1}^{J-1} exp(\alpha_l + x\beta_l)}, k = 1, \dots, J-1$$

and

$$\pi_J(x) = \frac{1}{1 + \sum_{l=1}^{J-1} exp(\alpha_l + x\beta_l)}$$

To test the significance of each variable's association with the response in a proposed model, we conduct likelihood ratio tests. For each variable, we could set H_0: a model with only intercept(s) versus H_A: proposed model with intercept(s) and slope(s). In other words, it tests whether there is a significant improvement in fit by including an explanatory variable or set of explanatory variables in the model. The test statistic is defined as

$$D = -2\left[\ell_{\mathrm{n}} - \ell_{\mathrm{p}}\right]$$

where ℓ_p and ℓ_n are log-likelihood of the proposed and null model, respectively. The test statistic D approximately follows a chi-square distribution with degrees of freedom equal to the number of parameters of proposed model minus the number in the model under H_0.

In our supervised clustering method, we fit a logistic regression model with only one explanatory variable each time, and use the likelihood ratio test of significance for the explanatory variable. (Note that we test a set of dummy variables corresponding to a single categorical variable together in one step.) The stronger the association between an explanatory variable and the response, the smaller the p-value of its likelihood ratio test will be. As in the SPCA, we select those explanatory variables with p-value smaller than a threshold ρ, forming X^* from all selected variables. Notice that choosing $\rho = 1$ includes all explanatory variables in X^*. Otherwise, ρ is a tuning parameter that we need to specify.

3.2.2. Transformation of categorical variables

After obtaining X^*, we want to run a clustering algorithm on it. However, there is a serious issue we need to consider: how to cluster with categorical variables? As mentioned in Section 2.2, all the clustering algorithms assign items into clusters based on some distance measurement. There is no natural definition of distance inherent in a categorical variable. Therefore, categorical variables need to be recoded so that they can also make the same contribution to a distance measurement as the numerical variables do.

Categorical variables can be mainly classified into two types: nominal and ordinal. Considering nominal variables first, the Euclidean distance between any two units with different categories should be the same, while the distance between units with the same category should be zero. To achieve this goal, we create one dummy variable for each level of the categorical variable. In this way, there are only two possible values of distance: one for matched categories, and the other for unmatched categories. For example, a nominal variable with three categories (A, B, C) is recoded by three dummy variables in Table 3.1. If one unit has level A and another has level B, the squared

Euclidean distance between these two units is 2. If one unit has level A and another has level C, the squared Euclidean distance is also 2.

Table 3.1 Recoding nominal variable with dummy variables

Nominal variable	Dummy 1	Dummy 2	Dummy 3
A	1	0	0
B	0	1	0
C	0	0	1

For an ordinal variable, we assume the distances between adjacent categories are the same. Then the distances between nonadjacent categories are larger than between adjacent categories, and there are m different distances, including 0, where m is the number of categories. We transform an ordinal variable with m levels into m-1 dummy variables. Table 3.2 presents an example of an ordinal variable with 4 levels. For example, the squared Euclidean distance between Excellent and Good is 1, while the squared Euclidean distance between Excellent and Poor is 3.

Table 3.2 Recoding ordinal variable with dummy variables

Evaluation	Dummy 1	Dummy 2	Dummy 3
Excellent	1	1	1
Good	0	1	1
Fair	0	0	1
Poor	0	0	0

3.2.3. Variable weighting

Like conventional clustering algorithms, we should first standardize all the variables in some way. For numerical variables, we rescale variables so that they have mean equal to zero and standard deviation equal to one. Suppose x_{1j} and x_{2j} are two independent values randomly chosen from the standardized numerical variable j. Using Euclidean distance, the contribution of a numerical variable j to the distance can be measured by $(x_{1j} - x_{2j})^2$. Now, calculate the expected value of $(x_{1j} - x_{2j})^2$:

$$E(x_{1j} - x_{2j})^2 = Var(x_{1j} - x_{2j}) + [E(x_{1j} - x_{2j})]^2 = Var(x_{1j}) + Var(x_{2j}) = 2$$

Next, consider categorical variables. There seems to be no straightforward method to standardize the recoded dummy variables. Suppose there is a nominal variable v with m categories, and we recode it with m dummy variables $z_1, \dots, z_m$. If we randomly select two observations v_1 and v_2 from v, and we let the corresponding dummy variables be $(z_{11}, \dots, z_{1m})'$ and $(z_{21}, \dots, z_{2m})'$, then the contribution of the nominal variable to the distance can be measured by $\sum_{k=1}^{m}(z_{1k} - z_{2k})^2$. We want the expected distance between two randomly selected values of a categorical variable to be the same as that of the standardized numerical variable. Now, calculate the expected value of $\sum_{k=1}^{m}(z_{1k} - z_{2k})^2$:

$$E(\sum_{k=1}^{m}(z_{1k} - z_{2k})^2) = P(y_1 \neq y_2) * 2 = 2 * (1 - \sum_{k=1}^{m} {\pi_k}^2),$$

where π_k is the probability of kth category. In order to make $E(\sum_{k=1}^{m}(z_{1k} - z_{2k})^2) = E(x_{1j} - x_{2j})^2$, we need to standardize dummy variables by dividing by $\sqrt{1 - \sum_{k=1}^{m} {\pi_k}^2}$. In practice, we need to use sample proportions instead of π_k. As an example, the dummy variables in Table 3.1 are now standardized in Table 3.3

Table 3.3 Recoding nominal variable with standardized dummy variables

Nominal variable	Dummy 1	Dummy 2	Dummy 3
A	$\frac{1}{\sqrt{1-({\pi_A}^2+{\pi_B}^2+{\pi_C}^2)}}$	0	0
B	0	$\frac{1}{\sqrt{1-({\pi_A}^2+{\pi_B}^2+{\pi_C}^2)}}$	0
C	0	0	$\frac{1}{\sqrt{1-({\pi_A}^2+{\pi_B}^2+{\pi_C}^2)}}$

For ordinal variables, we can use the same approach to standardizing as we do for a nominal variable.

After standardizing all the explanatory variables, we could run a clustering algorithm on selected variables with equal weight. However, since each explanatory variable has different relationship with the response, their influences on forming clusters should not be the same. Even when we select only variables with small p-values, their

associations with the response may still be quite different. For instance, two explanatory variables with p-value 0.05 and 10^{-5} may have very different capacity for predicting class response. Therefore, we want to rescale each explanatory variable according to its predictive potential prior to clustering. This involves multiplying each selected, standardized variable by a constant depending on that variable's association with the response. The constant $-log(p_j)$, where p_j is the p-value for variable j, seems to be a reasonable choice for the following reasons (see Figure 3.1):

- $-log(p)$ is a decreasing function of the p-value. The smaller the p-value is, larger the $-log(p)$ is.
- The range of variances produced by $-log(p)$ seems reasonable in that there are clear differences between important and less-important predictors, yet no extremely large standard deviations are produced.
- The value of $-log(p)$ changes gently when p-value is greater than 0.2, so that such variables do not differ greatly in weight. The value of $-log(p)$ changes sharply when p-value is near zero, so that small change of p-values near zero result in substantially different influence on the clustering.

We notice that $-log(p)$ provides relatively much smaller multipliers to explanatory variables with large p-values than to those with small p-values. Thus, weighting by the $-log(p)$ is kind of like a screening for highly associated explanatory variables. In addition, selecting explanatory variables according to some threshold ρ is arbitrary. A variable with a p-value just above the threshold would be excluded, while one with a p-value just below the threshold would be selected. In fact, these two explanatory variables may have no big difference in their predictive ability. If we use $-log(p)$ weights without preselecting, this situation can be avoided. Therefore, weighting variables directly without preselecting is an alternative procedure to the approach to variable selection used originally in Bair et al. (2006). We therefore have four variants of our procedure: with or without variable selection, and with or without weighting. In the simulation study in Chapter 5, we will compare several different weighting and selection combinations.

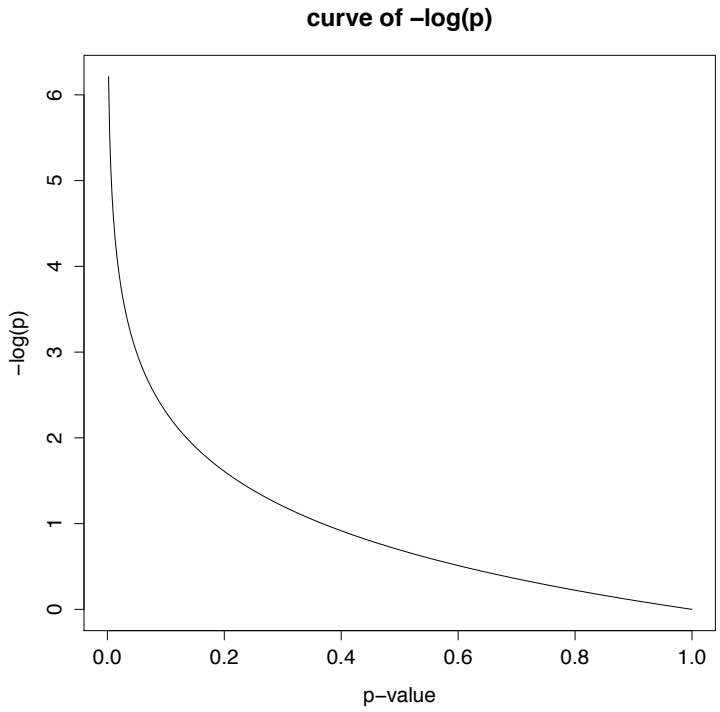

Figure 3.1 Curve of $-log(p)$ ***for*** $p \in (0, 1)$

3.2.4. Perform K-means algorithm

After variable screening, categorical variable transformation, and variable weighting, we get a new explanatory variable set X^{**} that can be used within conventional clustering algorithms. In our supervised clustering method, we choose to perform K-means algorithm on X^{**}, although a different clustering method could have been considered. Before we conduct K-means algorithm, the number of clusters K needs to be determined. This K is a tuning parameter that influences the performance of our method directly. In next section, we will discuss how to find the optimal K.

We run the K-means algorithm several times, and choose the clustering assignment that minimizes the sum of within-cluster distance. Then for each cluster, the majority class of the cluster is assigned to be the predicted class for all items in that cluster.

3.3. Tuning parameters

In our method, there are two tuning parameters that need to be determined: the threshold p-value ρ and the number of clusters K. We use two different methods to seek the optimal tuning parameters.

3.3.1. Fitness function

The basic objective of our supervised clustering method is grouping observations with the same class into the same clusters, and meanwhile keeping the number of clusters not too large. The fitness function $q(C)$ used in SRIDHCR not only considers the performance of predicting class response but also provides a penalty on the number of clusters. It seems also appropriate to measure the performance of our supervised clustering method. Therefore, we would like to find optimal values of ρ and K that minimize $q(C)$. More specifically, we predetermine several different combinations of ρ and K (e.g., $\rho = 0.01, 0.05, 0.1, 0.2, 1.0; K = 1, \dots, 30$), and perform our supervised clustering method on each combination separately. In the end, we choose the corresponding ρ and K that have the minimum value of $q(C)$.

However, there is a drawback to this fitness function. As we mentioned in Section 2.3, no rigorous justification was given regarding choice of the penalty coefficient β. We need also predetermine the value of β. Small β tends to suggest a large K, while a relative large β prefers a small number of clusters due to the heavy penalty on K. Thus, the choice of tuning parameters can be subjective. A more objective method is needed.

3.3.2. Cross-Validation

Cross-validation is one of the most popular methods for estimating prediction error, and hence is often used for selecting tuning parameters (Hastie, Tibshirani, and Friedman 2009). In an r-fold cross-validation, the data set is randomly partitioned into r roughly equal subsets (e.g., 10-fold cross-validation is most commonly used). For each subset, a model is fit to the remaining $r - 1$ subsets and the fitted model is used to predict the outcomes for the excluded subset. After repeating this procedure on each

subset, every observation obtains a predicted value. Then the overall prediction error can be computed comparing these predictions to the original data.

Since cross-validation is randomized, the results are variable. In order to obtain more reliable prediction error, we need to repeat the cross-validation procedure a few times and calculate the average prediction error. Theoretically, the model with minimum average error should be chosen as the best model. However, there are many other models whose average errors are very close to the minimum one. Thus, a "1-SE" rule is often used with cross-validation, in which we choose the most parsimonious model whose average error is no more than one standard error above the error of the best model.

In our problem, the cross-validation procedure is as follows:

1. Randomly partition the data set into r equal subsets (we use $r = 10$).
2. Perform our supervised clustering method on $r - 1$ subsets, and find the majority class in each cluster. Then determine the cluster into which each observation of the remaining subset falls. The predicted class response for each observation is the majority class of its cluster.
3. For each single subset, repeat Step 2 to obtain predicted class response. After obtaining predicted class response on every observation, compute the misclassification rate for the full sample.
4. Repeat above steps several times for each different value of ρ and K. Then compute the average misclassification rates and their standard errors. Select the combination of ρ and K with minimum average value of misclassification rate. We apply the "1-SE" rule when we select K. As an example, Figure 3.2 shows how "1-SE" rule works.

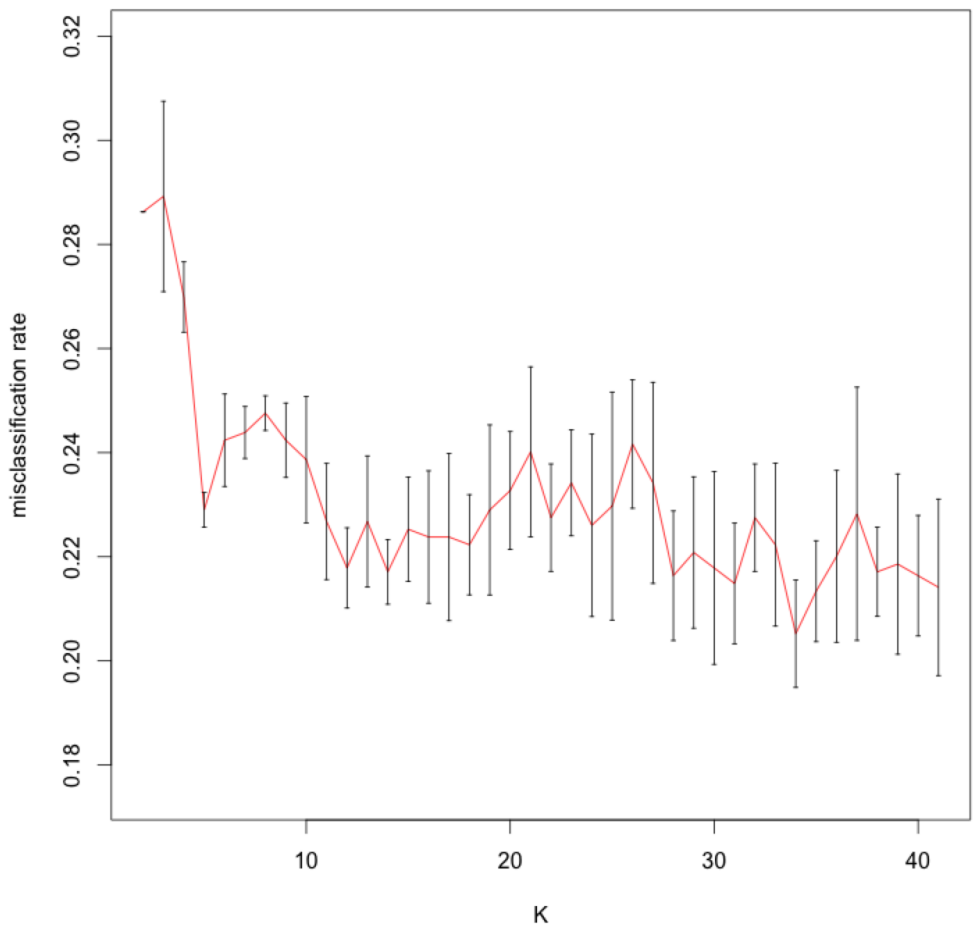

Figure 3.2 ***Misclassification rates of cross-validation ± 1 standard error: misclassification rate reaches the minimum value at K=34; Using the "1=SE" rule, K=31 would be chosen***

4. Examples to Compare Methods

In this chapter, our supervised clustering method is applied to the JIA data and compared with SRIDHCR, a classification tree, and a random forest. Since we use logistic models in the variable selection step in our method, we will also apply a stepwise logistic regression approach to the JIA data and compare it with our method. In addition, these methods are also applied to other data sets with multiple class response.

4.1. Description of JIA data

Our JIA data consists of 269 children who suffered from Juvenile Idiopathic Arthritis, supplied by Dr. Jaime Guzman of BC Children's Hospital. The data consists of 17 explanatory variables and 1 response. Table 4.1 provides more detailed descriptions of each variable.

Since parents may care mainly about whether their child can somehow achieve remission, we redefine a two-class response: no remission as one class and all the other three categories (all with some remission) as the other class. Among all these variables, there are two variables with missing data: ESR_RES and S_ASIAN_ANY. We need to handle these missing values before applying our supervised clustering method and other alternative methods on this data set. The S_ASIAN_ANY variable has 95% missing data, so it contributes very little information. We therefore eliminate this variable. On the other hand, ESR_RES has only 30 missing values. Because the missing rate is low, we used simple regression imputation to replace the missing data with surrogate values (Paul, 2002). The regression imputation procedure is as follows: treat ESR_RES as response and the other 15 explanatory variables (excluding S_ASIAN_ANY) as predictors, and fit a linear model using all complete cases. Then predict missing data based on the fitted linear model and replace the missing values with predicted values. This is performed prior to using any analytical methods.

Table 4.1 Description of JIA data

Variable Name	Type (levels)	Description
ESR_RES	Numeric	Erythrocite Sedimentation Rate (ESR) is a blood test that raises in response to inflammation
DIAGNOSIS	Categorical (10)	Subtype of juvenile arthritis according to International League of Associations for Rheumatology
PGADA	Numeric	Physician Global Assessment of Disease Activity (PGADA) as marked in a 10 cm line
ANA_RES	Categorical (4)	Antinuclear Antibody (ANA) text is done in blood and used for classification and prognosis
SEX	Categorical (2)	Patient's sex
WRIST_ANY	Categorical (2)	Swelling of the wrist joint at time of enrolment
HIP_ANY	Categorical (2)	Swelling of the hip joint at time of enrolment
ANKLE_ANY	Categorical (2)	Swelling of the ankle joint at time of enrolment
JOINTS_ SYMMETRIC	Categorical (2)	Swelling of the same joint in both sides of the body at enrolment
ARTHRITIS	Numeric	Parent's overall opinion of how much arthritis is affecting the child as marked in a 10 cm line
FIRST_N_ANY	Categorical (2)	First Nations ancestry
CH_KO_JA_ANY	Categorical (2)	Chinese, Korean or Japanese ancestry
S_ASIAN_ANY	Categorical (2)	South Asian ancestry
BLACK_ANY	Categorical (2)	Black ancestry
FHX_RH_DIS	Categorical (2)	History of rheumatic disease in a relative
CHAQ_VISIT1_ CHAQ_score	Numeric	Child Health Assessment Questionnaire (CHAQ) score at enrolment, with higher number meaning more functional difficulties due to arthritis
JAQQ_VISIT1_JAQ Q_score	Numeric	Juvenile Arthritis Quality of Life Questionnaire (JAQQ) score at enrolment, with higher number meaning poor quality of life
Response	Categorical (4)	Four categories: no remission; remission in six months; remission in twelve months with medication; remission in twelve months with no medication

4.2. Implement our supervised clustering method on JIA data

As the first step of our supervised clustering method, the associations between explanatory variables and response are measured by the p-values of the likelihood ratio tests (see Table 4.2).

Table 4.2 Associations between explanatory variables and response in JIA data (p-values of likelihood ratio test)

Variable	ANKLE_ ANY	WRIST_ ANY	JOINTS _SYMM ETRIC	DIAGN OSIS	ESR_R ES	SEX	ANA_R ES	JAQQ_s core
P-value	0.005	0.007	0.031	0.035	0.057	0.116	0.139	0.156
Variable	HIP_AN Y	PGADA	CHAQ_ score	CH_KO _JA_AN Y	BLACK_ ANY	ARTHRI TIS	FIRST_ N_ANY	FHX_R H_DIS
P-value	0.173	0.239	0.284	0.474	0.601	0.638	0.703	0.981

ANKLE_ANY and WRIST_ANY are most highly associated to the response, while FHX_RH_DIS does not seem to have any strong relationship to the response. To select explanatory variables, we use thresholds $\rho = 0.05, 0.1, 0.2$. The number of variables selected at each threshold is 4, 5, and 9, respectively. $\rho = 1$ is used to include all the explanatory variables. For each threshold, we run the algorithm both with $-log(p)$ weighting and without. For each combination of threshold and weighting, the other tuning parameter, the number of clusters K, needs to be determined. In order to maintain some interpretability of clustering, we choose K under 40. We start with the fitness function method to select tuning parameters ρ and K.

At first, we predetermine the penalty coefficient of the fitness function, $\beta = 0.1$, and run each combination of threshold, weighting, and K. Figure 4.1 shows the curves of the fitness function, the impurity and the penalty term ($\beta * Penalty(K)$). In the right-top graph ($\rho = 0.05$ and unweighted), we just try K fewer than 10, because there are only 15 different distances between observations for the four selected unweighted variables. From all of the graphs, we can see that penalty term keeps increasing and impurity has an overall decreasing trend, as the number of clusters K increases. Due to

the small predetermined value of β, the increase of the penalty term is generally less than the decrease of impurity. Thus, the fitness also has a decreasing tendency as the number of clusters increases. Under different thresholds, fitness functions reach minimum at relatively large K (between 32 and 37), and minimum values of fitness functions are very close. Moreover, for the same threshold, weighting or not weighting variables gives very similar results.

Next, we predetermine a larger value of the penalty efficient, $\beta = 0.3$. Figure 4.2 shows that as the number of clusters increases, the penalty term increases and impurity has an overall decreasing trend, which is just the same as Figure 4.1. However, we notice that the fitness functions reach minimum value at K=2, and then have an increasing trend. This is because the increase of penalty term is larger than the improvement of impurity. We list the optimal number of clusters, the minimum value of fitness function, and corresponding impurity and penalty term in Table 4.3, so as to select the optimal tuning parameter ρ and K, and analyze the influence of different values of β.

From Table 4.3, we can see that the smallest value of fitness function is 0.370 for $\beta = 0.1$. Therefore, using the fitness function method with $\beta = 0.1$, the optimal tuning parameters are: $\rho = 0.2$ and $K = 36$, unweighted. Meanwhile, we notice that the minimum values of fitness function with different ρ are very close, which indicates the threshold ρ does not have great influence on the value of fitness function. To simplify the tuning parameter selection procedure, we might ignore the choice of ρ, and just use $\rho = 1$. The feasibility of this behaviour will be examined in the simulation study in Chapter 5. On the other hand, when $\beta = 0.3$, the optimal tuning parameters are: $K = 2$ and $\rho = 0.05$ or 0.1, unweighted. As β increases from 0.1 to 0.3, the penalty for increasing the number of clusters increases as well. Consequently, the fitness function method will choose a smaller number of clusters as the optimal tuning parameter. The results of fitness function method heavily depend on the choice of β, which makes it a very subjective method.

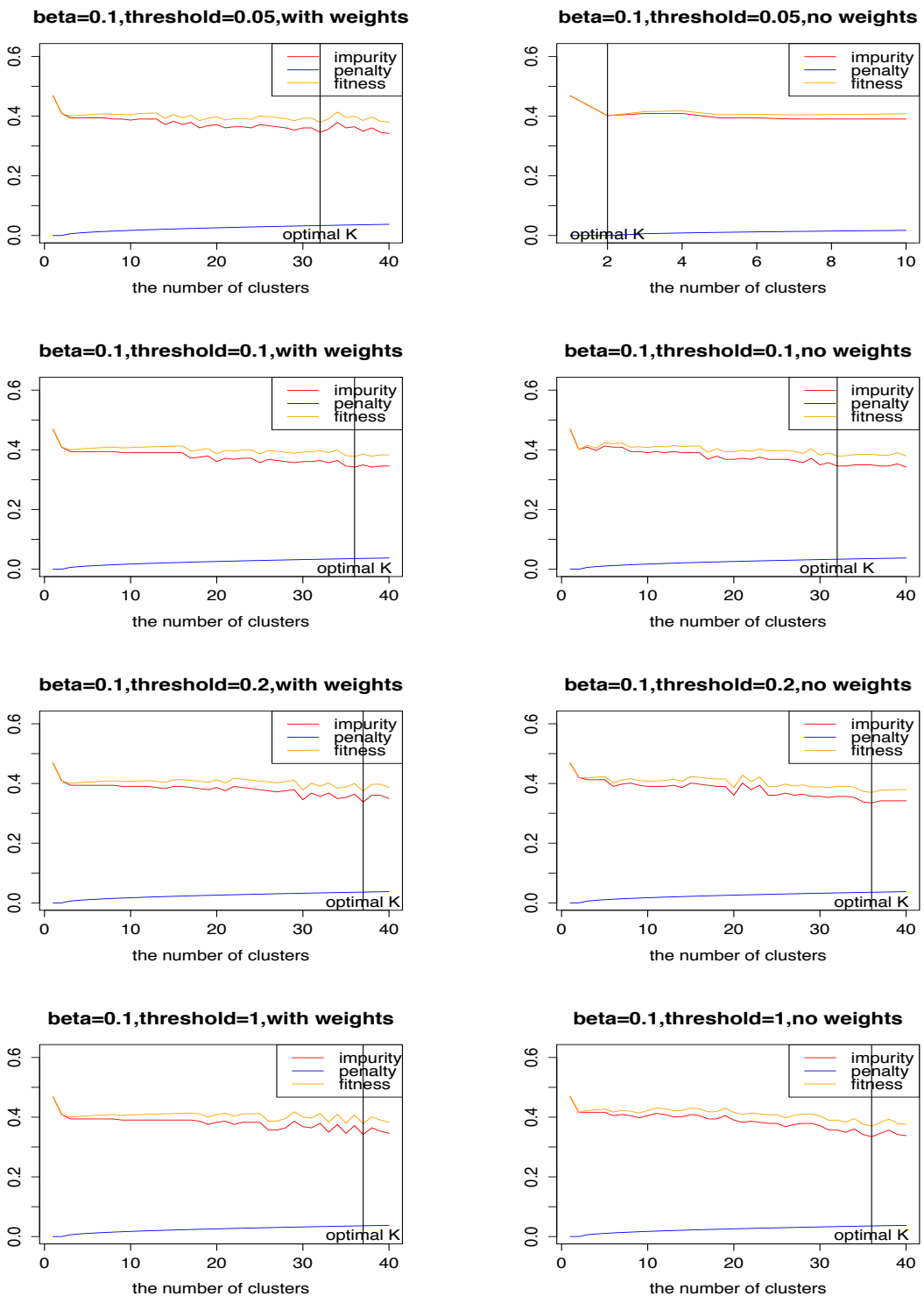

Figure 4.1 ***When $\beta = 0.1$, the figures of fitness function for different thresholds $\rho = 0.05, 0.1, 0.2$, and 1***

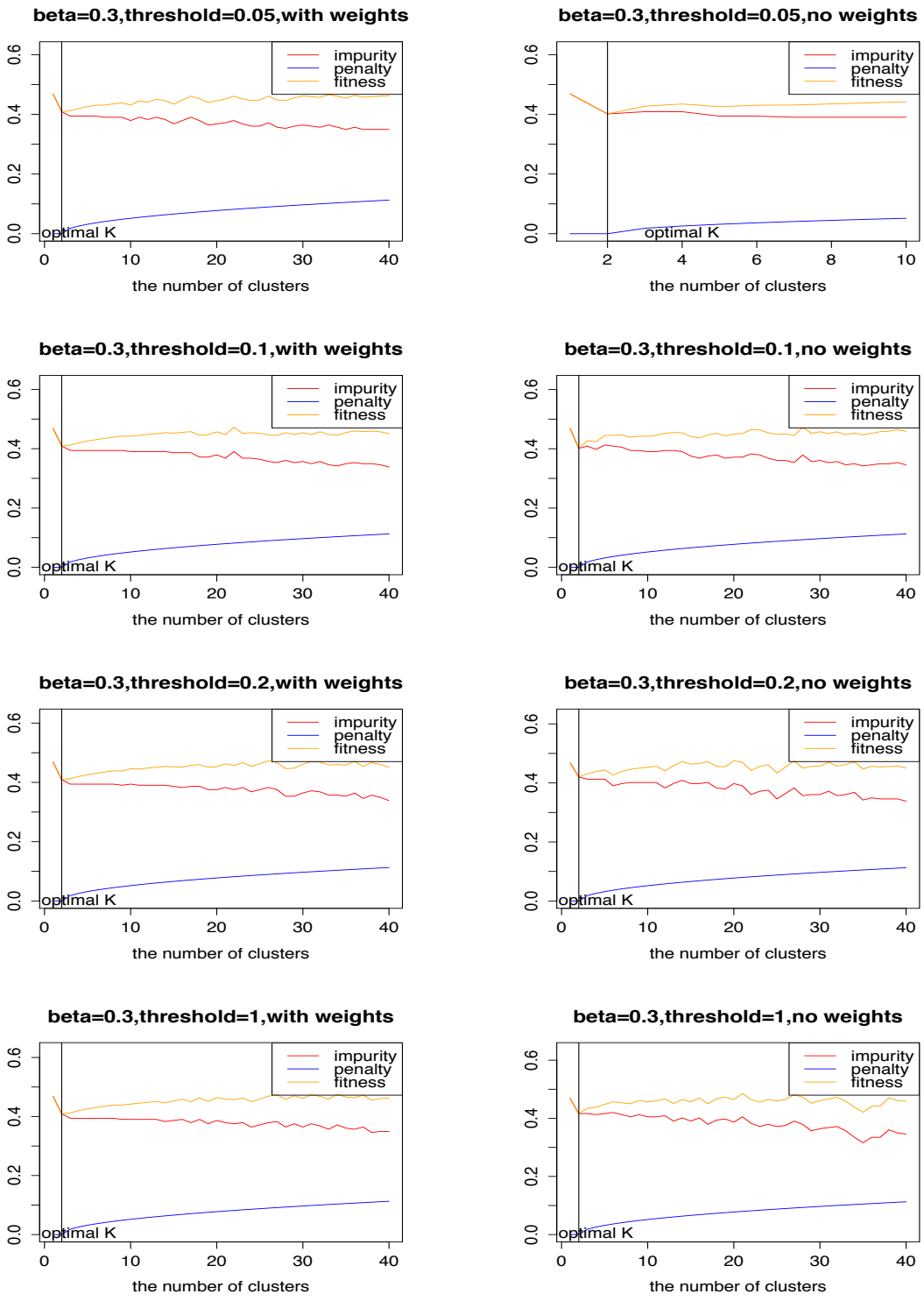

Figure 4.2 ***When $\beta = 0.3$, the figures of fitness function for different thresholds $\rho = 0.05, 0.1, 0.2$, and 1***

Table 4.3 Results of the fitness function method for selecting tuning parameters

ρ	$\beta = 0.1$				$\beta = 0.3$			
	Optimal K	$q(C)$	Impurity(*C*)	Penalty term	Optimal K	$q(C)$	Impurity(*C*)	Penalty term
0.05, weighted	32	0.379	0.346	0.033	2	0.409	0.409	0
0.05, unweighted	2	0.401	0.401	0	2	0.401	0.401	0
0.1, weighted	36	0.378	0.342	0.036	2	0.409	0.409	0
0.1, unweighted	32	0.379	0.346	0.033	2	0.401	0.401	0
0.2, weighted	37	0.374	0.338	0.036	2	0.409	0.409	0
0.2, unweighted	36	0.370	0.334	0.036	2	0.420	0.420	0
1, weighted	37	0.378	0.342	0.036	2	0.409	0.409	0
1, unweighted	36	0.371	0.335	0.036	2	0.416	0.416	0

Next, a less subjective method, cross-validation, is used to select the tuning parameters ρ and K. For the same combinations of ρ and weighting, and for each K (from 2 to 40), we perform our supervised clustering method using cross-validation five times, and then record the average misclassification rates and their standard errors. Using $\rho = 0.05$, and unweighted approach, there are only 15 distinct distances between observations. The number of distinct distances could be less in each cross-validated fold. Thus we try K fewer than 8 for this approach. Figure 4.3 lists the cross-validation results for different ρ and weighting. In these graphs, misclassification rates decrease substantially as K increases at the beginning. After that, as K increases, misclassification rates have increasing trends due to overfitting. Applying "1-SE" rule, the optimal K for different ρ are listed in Table 4.4. We notice that 4 weighted cases have the same optimal K and the same corresponding misclassification rate 0.394. As a result, the optimal tuning parameters are: $K = 3$ and any ρ, weighted. In addition, for the

same threshold ρ, weighting variables provides slightly lower misclassification rate than not weighting.

After selected the tuning parameters, we run K-means algorithm and assign predicted class to each cluster. The detailed results are not presented here.

Table 4.4 The results of cross-validation method

ρ	weighted(Y/N)	Optimal K	CV Misclassification Rate
0.05	Y	3	0.394
	N	3	0.403
0.1	Y	3	0.394
	N	2	0.401
0.2	Y	3	0.394
	N	3	0.413
1	Y	3	0.394
	N	3	0.419

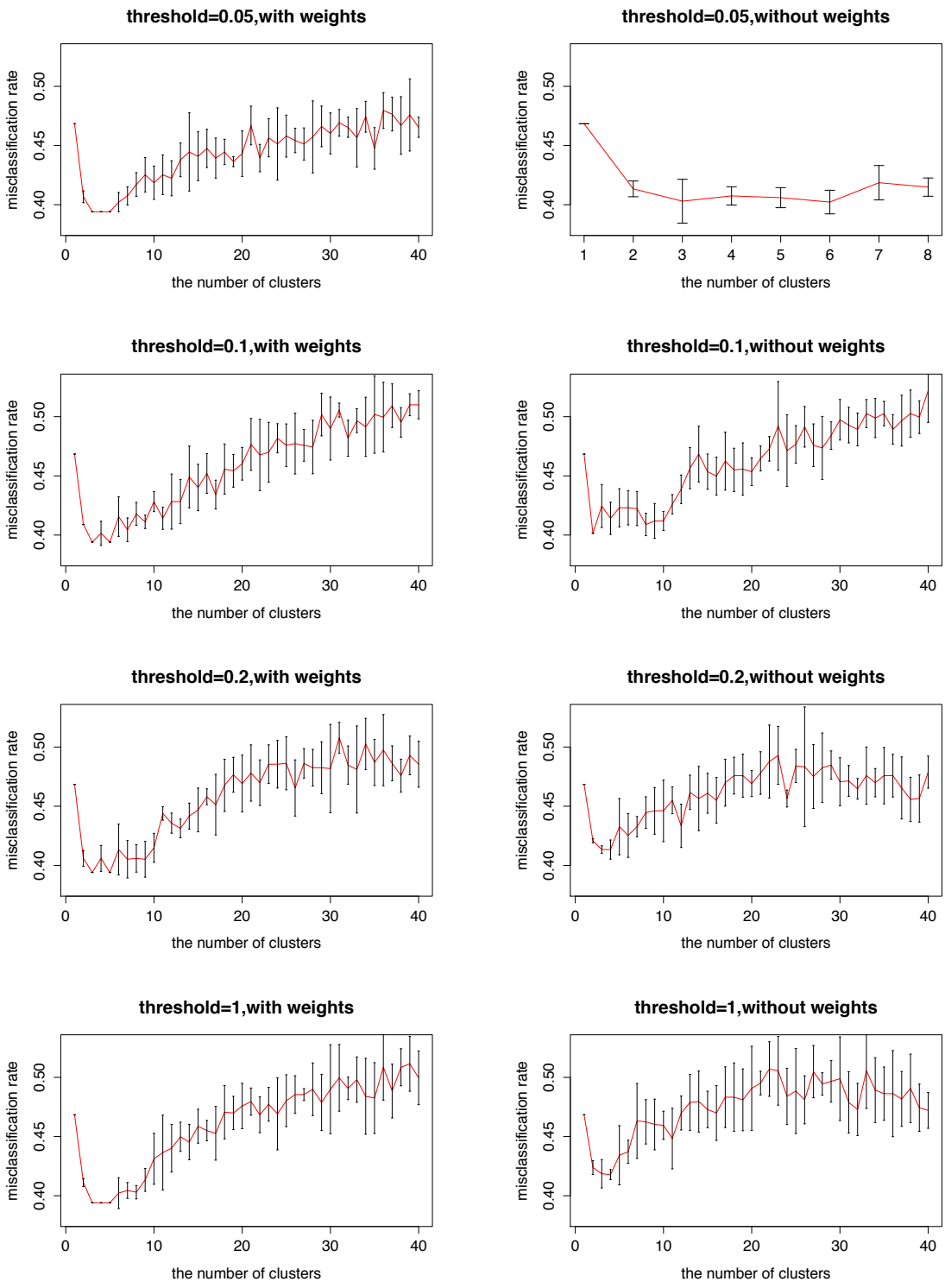

Figure 4.3 Misclassification rate V.S the number of clusters using cross-validation

4.3. Comparison with other methods

In this section, we apply SRIDHCR, stepwise logistic regression, a classification tree and a random forest to the JIA data, and compare their performances with our supervised clustering method.

4.3.1. SRIDHCR on JIA data

Since SRIDHCR is also based on Euclidean distance to assign clusters, it can only handle numeric explanatory variables. We use the recoding method described in Section 3.2.2 to create sets of numerical variables out of each categorical variable. We conduct the SRIDHCR algorithm five times for each different value of β (0.1 and 0.3), and select the solution with minimum value of the fitness function.

SRIDHCR selects the best clustering based on the same fitness function that we used for our supervised clustering method. Thus, we list the results of these two methods using the same penalty coefficients for comparison in Table 4.5. When $\beta = 0.1$, the two methods choose similar number of clusters, but the impurity of SRIDHCR is much better than our method. When β rises to 0.3, SRIDHCR still chooses the number of clusters more than 30, while our method suggests 2 as the optimal number of clusters. Even though SRIDHCR provides better impurity, we cannot conclude that SRIDHCR will have better performance on classifying new data. Because this impurity is measured within JIA data itself rather than on the new data, overfitting could happen.

In order to check the performance of SRIDHCR on classifying new data, we use cross-validation to calculate a misclassification rate. Moreover, cross-validation not only provides an estimate of the out-of-sample misclassification rate, but also gives an approach to select optimal β. For different values of β $(0.1, 0.2, \dots, 1)$, we perform SRIDHCR using cross-validation with five repetitions, and then record the average misclassification rates and their standard errors. We apply the "1-SE" rule (choose the largest β whose misclassification rate is no more than one standard error above the minimum misclassification rate) to select optimal β and record the corresponding misclassification rate. The results are given in Table 4.6 and will be discussed in Section 4.3.2.

Table 4.5 Comparative performances of SRIDHCR and our supervised clustering method on JIA data

	$\beta = 0.1$				$\beta = 0.3$			
	Optimal K	$q(C)$	Impurity (C)	Penalty term	Optimal K	$q(C)$	Impurity (C)	Penalty term
SRIDHCR	32	0.210	0.177	0.033	31	0.301	0.202	0.099
Our Method	38	0.371	0.334	0.037	2	0.409	0.409	0

4.3.2. Classification tree, random forest and stepwise logistic regression on JIA data

In a classification tree, we use cross-validation to prune a tree, and compute the misclassification rate at the same time. In a random forest, an analogous misclassification rate—"out of bag" (OOB) misclassification rate—can be computed, which also builds the classifier using part of the data and predicts on the rest of the data. In the stepwise logistic regression, we use forward selection to select the best logistic model, and then predict class response according to this best model. The misclassification rate is also calculated by cross-validation. We will record the cross-validated misclassification rates of the classification tree, stepwise logistic regression, and our supervised clustering method and the OOB misclassification rate of random forest to compare their performance of predicting classes. Moreover, we will calculate their corresponding 95% confidence intervals.

The cross-validation procedure for pruning a classification tree is shown in Figure 4.4. Applying "1-SE" rule, the cross-validation suggests no split in the classification tree, which means classifying all the items into the majority classes. After computing the out-of-sample misclassification rate of random forest and stepwise logistic regression, we list all the results in Table 4.6 including SRIDHCR. Applying cross-validation to SRIDHCR selects 0.8 as the optimal value of β, and the resulting cross-validated misclassification rate is worse than that of our supervised clustering method. Our method has the smallest out-of-sample misclassification rate among these methods, but there are obvious overlaps between their 95% confidence intervals. In conclusion, among these methods, our supervised clustering method has slightly better performance for predicting class response in JIA data, although the differences are not significant.

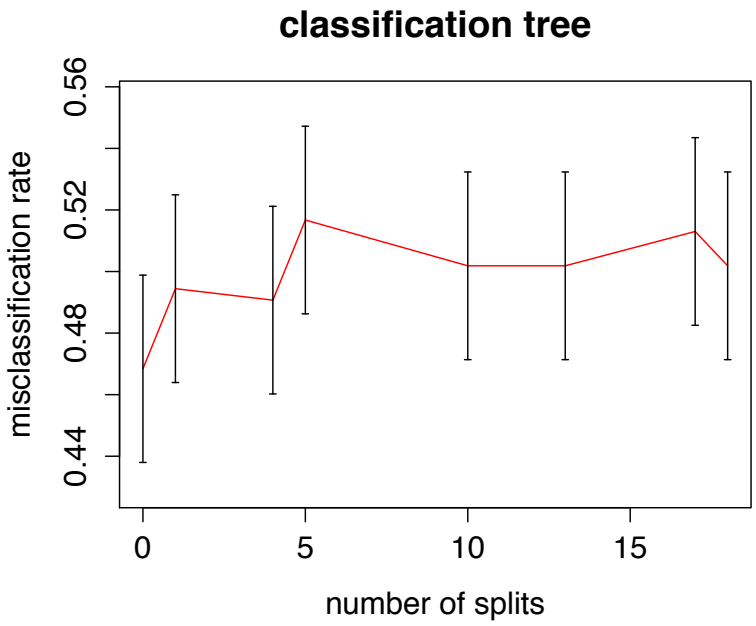

Figure 4.4 ***Cross-validation procedure for pruning classification tree on JIA data***

Table 4.6 ***Comparative performances of different methods on JIA data***

	Classification Tree	Random Forest	Our Method	SRIDHCR ($\beta = 0.8$)	Stepwise Logistic
Misclass. Rate	0.468	0.413	0.394	0.446	0.424
CI	[0.408, 0.528]	[0.354, 0.472]	[0.336, 0.452]	[0.387, 0.505]	[0.365, 0.483]

4.4. Multiple-class examples

In the JIA data, our supervised clustering method provides competitive performance compared with other classification methods and SRIDHCR. However, the response we used in JIA data has only two classes. We would like to know how well our supervised clustering method could perform on data with more than two classes of response. Therefore, we apply our method as well as the alternative methods to other data sets with multiple classes. These data sets are available from UCI Machine

Learning Repository (http://archive.ics.uci.edu/ml/datasets.html). Table 4.7 gives a summary for these data sets.

Table 4.7 Summary of multiple-classes data sets

Data Set Name	Number of Observations	Response Levels	Explanatory variables
Iris	150	3	4 numeric
Flag	194	4	18 categorical, 10 numeric
Lymphography	148	4	18 categorical

In our supervised clustering method, we use threshold $\rho = 1$ and weight explanatory variables differently depending on their associations with the class response. The out-of-sample misclassification rates and corresponding 95% confidence intervals calculated by classification tree, random forest, SRIDHCR, stepwise logistic regression and our method are listed in Table 4.7. In the Iris data, our method gives the third-best misclassification rate. In the Flag data, our method gives the best misclassification rate. In the Lymphography data, our method gives the third-best misclassification rate. Since most confidence intervals are highly overlapped, there is no significant difference among the results of these methods. Overall, our supervised clustering method maintains competitive performance on these multiple-class data sets, being behind only random forest and stepwise logistic regression sometimes, which do not provide an interpretable grouping of units.

Table 4.8 Comparative performances of different methods on multiple-classes data sets

Data		Our Method	Classification Tree	Random Forest	SRIDHCR	Stepwise Logistic
Iris	Mis. rate	0.044	0.053	0.040	0.059	0.033
	CI	[0.011, 0.077]	[0.017, 0.089]	[0.009, 0.071]	[0.021, 0.097]	[0.004, 0.062]
Flag	Mis. rate	0.251	0.289	0.283	0.495	0.258
	CI	[0.190, 0.312]	[0.225, 0.353]	[0.220, 0.346]	[0.425, 0.565]	[0.196, 0.320]
Lym	Mis. rate	0.524	0.547	0.500	0.566	0.507
	CI	[0.444, 0.604]	[0.467, 0.627]	[0.419, 0.581]	[0.486, 0.646]	[0.426, 0.588]

5. Simulation Study

In Chapter 4, our supervised clustering method, SRIDHCR, classification tree, and random forest are applied to the JIA data. However, none of them classifies the data well. A possible reason is that the explanatory variables do not provide strong enough information to predict the class response. In this chapter, in order to further investigate the performance of our supervised clustering method, we simulate class responses based on the explanatory variables of JIA data, and apply our method to the simulated data. For comparison, SRIDHCR, classification tree, and random forest are applied on the simulated data as well.

5.1. Simulate data

First, we use the recoding method described in Section 3.2.2 to create sets of numerical variables out of each categorical variable and standardize each variable to have mean=0 and standard deviation=1. Denote this set of explanatory variables by X_0. Then we use a logistic regression function to obtain the probability of remission in response class for each observation:

$$logit(\pi) = \alpha + X_0\gamma \tag{5.1}$$

where $\pi = (\pi_1, \dots, \pi_n)'$ is a vector of the probabilities of remission for each observation, and we assume $\gamma \sim N(0, \sigma^2 I)$. Based on properties of the normal distribution, $logit(\pi)$ also follows a normal distribution with mean α and variance $\sigma^2 X_0 X_0'$. The diagonal elements of $\sigma^2 X_0 X_0'$ are the variance of each $logit(\pi_i)$. To simplify the process of data simulation, we use the average diagonal value from $X_0 X_0'$ as the common variance. After obtaining π from (5.1), we randomly generate a binary response for each observation according to its probability of remission. This process of data simulation can be briefly described as follows:

$$\left.\begin{matrix}\text{Manipulate } X_0 \\ \text{Choose } \alpha \text{ and } \sigma\end{matrix}\right\} \xrightarrow{Equation\ (5.1)} \pi \xrightarrow{randomly\ generate} class\ response$$

Therefore, only two parameters, α and σ need to be determined when we are simulating data. Different values of α and σ result in different structures of class responses. In order to select α and σ, we create two constraints to describe the distribution of probabilities in the sample: $P(\pi_i < 0.5) = a_1$ and $P(\pi_i < 0.1) = a_2$. The goal is to predict $I(\pi_i > 0.5)$ for each observation. That is, we think of any simulated "patient" with $\pi_i > 0.5$ as being more likely to be a "success" (e.g., in remission) than not. Even though the random binary response might be opposite the true probability for some observations ($\pi_i > 0.5$ but $y_i = 0$, for example), we want our models to guess the underlying probability. To this end, observations with π_i close to 0.5 are more likely to have misleading responses than those with $\pi_i \approx 0$ or $\pi_i \approx 1$. A problem is "hard-to-classify" if it has many observations with $\pi_i \approx 0.5$, and "easy-to-classify" if it has many π_i's near 0 or 1. In the simulation study, we consider three different scenarios.

Scenario 1: To simulate a "hard-to-classify" class response with equal class probability, we want approximately half of observations have remission probability π_i less than 0.5, and many $\pi_i's$ are around 0.5. We set $a_1 = 0.5$ and $a_2 = 0.05$ to spread out the probability but maintain a mound shape. Solving these two constraints, we get $\alpha = 0$ and $\sigma = 0.22$. Figure 5.1 shows the histogram of π_i after simulating 100 data sets of size 269.

Scenario 2: To simulate an "easy-to-classify" class response, we want most remission probabilities π_i to be extremely low or high. We use $a_1 = 0.5$ and $a_2 = 0.25$ to form this valley shape of π_i. Solving these two constraints, we get $\alpha = 0$ and $\sigma = 0.69$. We simulate data 100 times and draw the histogram of π_i in Figure 5.2.

Scenario 3: In the JIA data, we could define another two-class response: "full remission" (remission in 12 months with no medication) and not "full remission". There are 66 "full remission" patients out of 269. Two classes are no longer approximately even at this time. We would like responses that resemble this situation. Thus, we use $a_1 = 1 - 66/269$ and play with a_2 values until we find one that gives us a generally

decreasing density of π_i. That one turns out to be $a_2 = 0.3$, which leads to $\alpha = -1.25$ and $\sigma = 0.3$. We simulate data 100 times and draw the histogram of π_i in Figure 5.3.

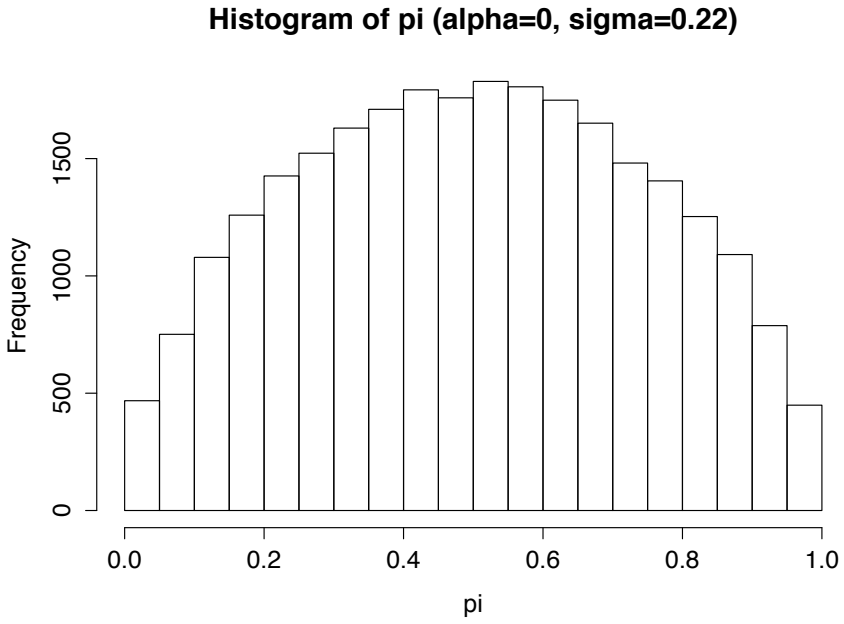

Figure 5.1 Histogram of π_i for $\alpha = 0$ and $\sigma = 0.22$

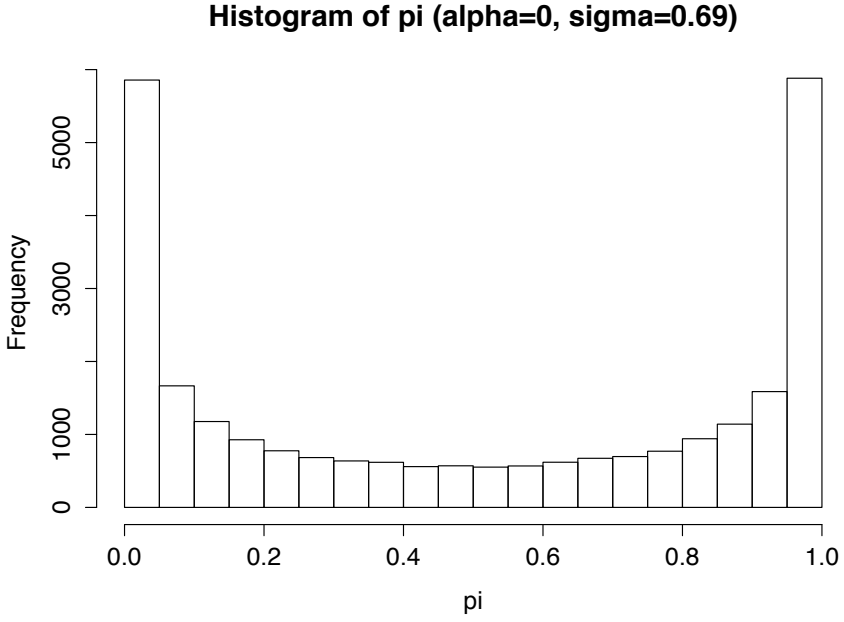

Figure 5.2 Histogram of π_i for $\alpha = 0$ and $\sigma = 0.69$

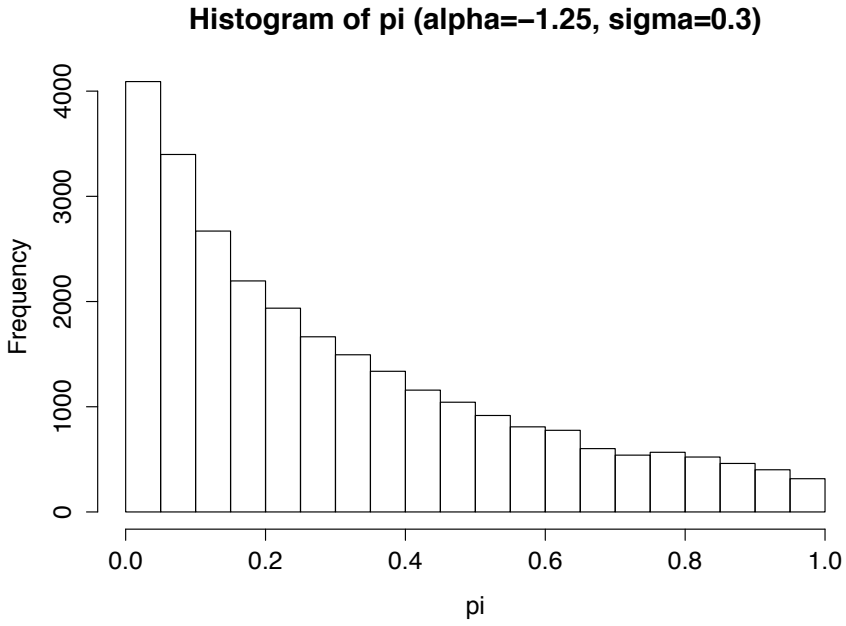

Figure 5.3 Histogram of π_i for $\alpha = -1.25$ and $\sigma = 0.3$

5.2. Pilot Study

Before we start to simulate data, a pilot study is conducted in order to determine the number of data sets we need in the simulation study. The objective of the simulation study is to obtain an estimate of the average population misclassification rate by comparing the predicted class to the expected class, $I(\pi_i > 0.5)$, and the corresponding confidence interval for each method. In order to obtain relatively accurate estimates of the misclassification rate, we want to keep the confidence interval narrow. Specifically, the width of 95% confidence interval should be no more than 0.02 (i.e., ± 0.01).

If we assume the population misclassification rate of method l, m_l, approximately follows a normal distribution and that all methods have the same variance, then we can use normal-based methods to find confidence intervals and estimate sample size. A 95% confidence interval for the misclassification of a method is

$$\left[\bar{m}_l - Z_{0.025} \cdot \sqrt{\frac{Var(m_l)}{M}},\ \bar{m}_l + Z_{0.025} \cdot \sqrt{\frac{Var(m_l)}{M}}\right],$$

where $\overline{m}_l$ is the sample mean of m_l, $Var(m_l)$ is the variance of m_l, M is the number of data sets we simulate, and $Z_{0.025}$ is the 97.5% quantile of standardized normal distribution. We wish to estimate misclassification rates to within ± 0.01 with 95% confidence. When we set the width of a 95% confidence interval to 0.02, we still need to know $Var(m_l)$ in order to calculate the number of data sets we need. We estimate $Var(m_l)$ using a pilot study. Since SRIDHCR and our supervised clustering method take a relatively long time for running, we use random forest on 100 simulated data sets for each scenario to estimate the common variances. The estimated variance of m_l and estimated number of data sets are listed in Table 6.1. In order to make the width of 95% confidence interval of misclassification rate under 0.02 for all these three scenarios, we decide to simulate 120 data sets for each scenario, in case variances for other methods are slightly larger.

Table 6.1 Estimated variance of e_i for three scenarios

	Scenario 1	Scenario 2	Scenario 3
Estimated Variance	0.00264	0.00108	0.00128
Number of data sets	101	41	49

5.3. Results and Discussions of Simulation Study

We apply our supervised clustering method, SRIDHCR, classification tree and random forest to each simulated data set, then calculate cross-validation misclassification rates for our method, SRIDHCR and classification tree, and OOB misclassification rate for random forest. We also calculate a population misclassification rate by comparing the predicted class to the expected class, $I(\pi_i > 0.5)$. In our supervised clustering method, we use $\rho = 0.1$ and 1, and use both weighted and unweighted versions. We also use cross-validation to choose the optimal settings among these four. We record the average out-of-sample misclassification rate $\bar{e}$ and the average population misclassification rate $\overline{m}$ for all the methods, and the average number of clusters $\overline{K}$ in our method and the average number of terminal nodes in classification tree. In addition, we calculate 95% confidence intervals for the mean of e_l:

$$\left[\bar{e}_l - t_{0.025,119} \cdot \sqrt{{S_l}^2/120}\,,\ \bar{e}_l + t_{0.025,119} \cdot \sqrt{{S_l}^2/120}\right]$$

where S_l^2 is the sample variance of the misclassification rates for method l, and $\mathrm{t}_{0.025,119}$ is the 97.5% quantile of a Student's t-distribution with 119 degrees of freedom. Similarly, 95% confidence intervals for the mean population misclassification rate m_l are also calculated. The results for all three scenarios are listed in Table 6.2.

First, we compare the results of our supervised clustering method with different tuning parameters in following aspects:

- Using $\rho = 1$ without weighting is actually using no supervision by the associations between explanatory variables and the class response. Compared to the other three selecting or/and weighting approaches, it provides both worse out-of-sample misclassification rates and worse population misclassification rates, which means selecting or/and weighting explanatory variables improves the performance of our method for predicting class response.
- Using the same ρ, a weighted approach provides smaller misclassification rate than unweighted approach does. Thus, weighting explanatory variables by $-log(p)$ seems to help to improve the performance of our method for predicting class response.
- Using weighted explanatory variables, both $\rho = 1$ and $\rho = 0.1$ result in similar misclassification rates, and they are not much different from the optimal result of our method. The choice of ρ has relatively small influence on the results of our method.
- Running all four methods and selecting the one with the best cross-validation error results in only a very small improvement in population misclassification rate compared to the two weighted methods.

Based on these results, we can recommend the use of the weighted procedure without prior variable selection for this classification task.

Table 6.2 The results of simulation study (average misclassification rate, confidence interval, the number of clusters)

		Our Method					Class. Tree	Rand. For.	SRID-HCR
		$\rho = 0.1$, weighted	$\rho = 0.1$, unweighted	$\rho = 1$, weighted	$\rho = 1$, unweighted	CV-Optimal t			
Scenario 1	$\bar{e}$	0.340	0.353	0.338	0.392	0.330	0.377	0.362	0.393
	CI ($\bar{e}$)	[0.334, 0.347]	[0.347, 0.360]	[0.332, 0.344]	[0.385, 0.399]	[0.325, 0.336]	[0.369, 0.386]	[0.354, 0.370]	[0.385, 0.400]
	$\bar{m}$	0.268	0.291	0.263	0.336	0.258	0.284	0.246	0.320
	CI ($\bar{m}$)	[0.257, 0.278]	[0.280, 0.301]	[0.253, 0.272]	[0.325, 0.347]	[0.249, 0.266]	[0.271, 0.297]	[0.238, 0.253]	[0.311, 0.329]
	$\bar{K}$	8	9	9	10	10	3		
Scenario 2	$\bar{e}$	0.239	0.273	0.240	0.307	0.233	0.263	0.214	0.290
	CI ($\bar{e}$)	[0.232, 0.247]	[0.264, 0.281]	[0.233, 0.247]	[0.299, 0.316]	[0.226, 0.239]	[0.254, 0.271]	[0.207, 0.221]	[0.282, 0.299]
	$\bar{m}$	0.211	0.251	0.210	0.287	0.205	0.203	0.165	0.261
	CI ($\bar{m}$)	[0.202, 0.219]	[0.243, 0.259]	[0.202, 0.218]	[0.278, 0.295]	[0.197, 0.212]	[0.194, 0.212]	[0.159, 0.171]	[0.253, 0.269]
	$\bar{K}$	17	15	17	15	18	4		
Scenario 3	$\bar{e}$	0.246	0.258	0.246	0.274	0.241	0.271	0.256	0.280
	CI ($\bar{e}$)	[0.240, 0.252]	[0.252, 0.264]	[0.240, 0.252]	[0.268, 0.280]	[0.235, 0.246]	[0.265, 0.278]	[0.250, 0.262]	[0.274, 0.286]
	$\bar{m}$	0.166	0.182	0.165	0.198	0.163	0.181	0.151	0.198
	CI ($\bar{m}$)	[0.158, 0.173]	[0.174, 0.190]	[0.157, 0.173]	[0.189, 0.207]	[0.156, 0.171]	[0.173, 0.190]	[0.146, 0.157]	[0.190, 0.205]
	$\bar{K}$	11	11	11	10	14	2		

Next we compare the results of our method ($\rho = 1$, weighted) to the other three alternative methods. Our supervised clustering method provides the best out-of-sample misclassification rate among these four methods in scenario 1 and scenario 3, and the second best misclassification rate in scenario 2. Considering the population misclassification rate, random forest shows the best ability to predict the population class response. Our supervised clustering method has better performance than classification tree except in the easy-to-classify Scenario 2, where there is considerable overlap in their confidence intervals. SRIDHCR has the worst population misclassification rate of

the four methods studied. Overall, our supervised clustering method shows competitive performance for predicting class response compared to SRIDHCR, classification tree, and random forest. In particular, our method is significantly better than the classification tree in two of the three cases studied.

6. Conclusion and Future Work

This project developed a supervised clustering method applied to the JIA problem, which is a typical classification problem. In order to evaluate how well our supervised clustering method works on predicting class response, we compared it with an existing supervised clustering method (SRIDHCR) and two classification methods (classification tree and random forest). In Section 6.1, we summarize the main difficulties and challenges we faced and solved when developing our supervised clustering method, and discuss the performance of our method on predicting class response compared to other alternative methods. Section 6.2 suggests some future work that remains to be done.

6.1. Conclusion

When developing a supervised clustering method, a challenging issue is how to use the target class response to supervise on clustering. In the existing supervised clustering algorithm, SRIDHCR, supervision is implemented by using the impurity of a clustering to guide the choice of clustering directly. In our method, supervision is imposed by weighting explanatory variables differently according to their associations with the class response, extending an idea that was used originally in supervised principal component analysis. However, SPCA works only when all explanatory variables are continuous. Here we fit logistic regression model with one explanatory variable each time, and use the p-value from the likelihood ratio test of significance as the measure of association between the response and the explanatory variable. With this improvement, our supervised clustering method can be applied to data with both continuous and categorical explanatory variables.

All clustering algorithms assign observations into clusters based on some distance measurement. But there is no definition of distance in a categorical variable.

Therefore, before implementing a clustering algorithm, categorical variables need to be recoded so that they can make the same contribution to a distance measurement as the numeric variables do. We create dummy variables to replace categorical variables. The specific approach differs according to the type of categorical variable.

In the JIA example and simulation study, our supervised clustering method provides competitive results for predicting class response compared to SRIDHCR, classification tree and random forest. To be specific, the out-of-sample performance of our method relative to random forest fluctuates across different data sets, and our method shows better out-of-sample performance for predicting class response than classification tree. In the population misclassification rate, our method was clearly second-best among the four. This is not surprising. Random forest is known to be a good classifier (Hastie, Tibshirani, and Friedman 2009), and we had expected it to perform better than our method. We had hoped that our method could provide similar performance to a classification tree and perhaps be better than the alternative supervised clustering method. This is exactly what has happened. Moreover, our supervised method provides the form of results that patients' parents prefer to see, thus our supervised clustering method should be a good candidate approach in JIA problem.

While we have not explored any follow-up analysis of the clustering produced by our method, it would be easy to identify variables that are important in the clusterings, for example by using ANOVA tests with the explanatory variable as the response and the clusters as the treatment groups. Also, we could present mean values of each variable in each cluster, as well as each cluster's classification results (similar to those in Figure 2.1) to aid in a clinical interpretation of the clusters.

Even though our method provides encouraging performance, it is not without drawbacks. Compared with the high speed of the classification tree and the random forest (i.e., less than one second per data set), our method is very time-consuming using our R code. For example, our method takes about 10 minutes for the JIA data. If there are thousands of observations or hundreds of variables, the running time would be a very serious problem.

6.2. Future work

Although our supervised clustering method has already presented competitive performance for predicting class response, there are still some details that need to be investigated. We made a few arbitrary decisions in our method. For example, we choose the K-means algorithm for our clustering. In fact, we could try other clustering algorithms, which might give better results. Since performing a clustering algorithm is the last step of our supervised clustering method, we could follow all the original steps of the procedure, and just use a different clustering algorithm at the last step. In addition, we use $-log(p)$ to weight explanatory variables, but there is no rigorous justification for this. Many other options could be used for weighting explanatory variables. Furthermore, we simulated only three specific scenarios based on the JIA data, and used the same model for creating binary responses each time. More simulation of different data structures should be conducted in order to check the performance of our method in general. Finally, the R code for our method should be optimized so that the running time can be reduced, if possible.

References

Allison, P. D. (2002). *Missing data*. Thousand Oaks, Calif.: Sage Publications.

Bair, E., Hastie, T., Paul, D., & Tibshirani, R. (2006). Prediction By Supervised Principal Components. *Journal of the American Statistical Association*, *101*(473), 119-137.

Breiman, L. (1984). *Classification and regression trees*. New York [etc.: Chapman & Hall.

Eick, C., Zeidat, N., & Zhao, Z. (2004). Supervised Clustering -- Algorithms and Benefits. *16th IEEE International Conference on Tools with Artificial Intelligence*, 774-776.

Hashkes, P. J., and Laxer, R. M.(2005). Medical Treatment Of Juvenile Idiopathic Arthritis. *JAMA: The Journal of the American Medical Association*, *294*(13), 1671-1684.

Hastie, T., Tibshirani, R., & Freidman, J. (2009). *The elements of statstical learning: Data mining, inference, and prediction*. (2nd ed. ed.). New York: Springer Science Business Media, LLC.

Izenman, A. J. (2008). *Modern multivariate statistical techniques regression, classification, and manifold learning*. New York: Springer