Indian Statistical Institute, Kolkata

M. Tech. (Computer Science) Dissertation

Sparse Fuzzy Switching Regression Model

A dissertation submitted in partial fulfillment of the
requirements
for the award of Master of Technology
in
Computer Science

Author:
Biswajit Majumder
Roll No: CS-1430

Supervisor:
Prof. Nikhil R. Pal
ECSU Unit, ISI

M.Tech(CS) DISSERTATION THESIS COMPLETION CERTIFICATE
Student: Biswajit Majumder (CS1430)
Topic: Sparse Fuzzy Switching Regression Model
Supervisor: Prof. Nikhil R. Pal

This is to certify that the thesis titled "***Sparse Fuzzy Switching Regression Model***" submitted by **Biswajit Majumder** in partial fulfillment for the award of the degree of Master of Technology is a bonafide record of work carried out by him under my supervision. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission. The results contained in this thesis have not been submitted to any other university for the award of any degree or diploma.

Date: Prof. Nikhil R. Pal

Dedication

To my parents and my well wishers, without your help and encouragement it would not have been possible.

Acknowledgements

I would like to thank my dissertation supervisor Prof. Nikhil R. Pal for agreeing to guide me and for helping me to undertake work in the topic.

Last but not the least I am grateful to all my lab mates and seniors of **The Computational Intelligence Lab** of Indian Statistical Institute, Kolkata for helping me though out the project with their valuable suggestions.

Abstract

Unlike multiple regression, in switching regression, data are assumed to have come from more than one regression model but the association between the sample points and the models is not known. One approach to obtain the parameters of the switching regression model, is to formulate the problem using a mixture distribution. The estimators for this kind of distribution can be obtained using an iterative maximum likelihood method. The second approach is to obtain a fuzzy partition of the data using the fuzzy c-regression model (FCRM) algorithm. Here, the prototypes of the clusters are in the form of regression models. For switching regression, although there are evidences/reasons to believe that the data are generated by more than one model, usually it is not known whether all predictors are important for all regimes. This work is based around identifying useful predictors, independent variables, and eliminating the irrelevant ones in the fuzzy switching regression setup. We employ two different regularizers in the FCRM objective function to induce sparsity in the models and thereby select useful features. In the first case, the ordinary FCRM objective function is regularized using the least absolute shrinkage and selection operator (lasso) penalty i.e., using the ℓ_1 norm of the parameters of the regression models as the regularizer. In order to deal with the ℓ_1 norm, each parameter is modelled using two non-negative variables. For a given partition matrix, it leads to a bound constraint quadratic optimization problem. In the second case, we formulate the *non-negatve garrotte* penalty for the fuzzy c-regression model. In this case, for each variable we associate a non-negative weight or importance. We consider two versions of the problem: (1) for every model we use a different set of weights, (2) only one common set of weights is used for all models. We test both approaches on synthetic as well as real datasets. After comparing results of both the cases on these datasets, we conclude that garrotte is more effective in inducing sparisty, maintaining the same level of root mean square error. Lastly, we discuss a method to evaluate goodness of the feature selection methods. This evaluation method affirms that features selected by the *non-negative garrotte* penalty are useful.

Contents

Chapter 1

Introduction

Multiple linear regression model can be used to learn about the relationship between several independent or predictor variables and a dependent variable. The dependent variable is assumed to be a linear combination of the predictor variables. Assuming the data set $(X, \boldsymbol{y}) = \{(\boldsymbol{x_i}, y_i); i = 1, 2, \ldots, n\}$, where $\boldsymbol{x_i} = (x_{i1}, ..., x_{id})^T$ is the vector of predictor variables and y_i is the dependent variable, a multiple linear regression model can be written as,

$$y_i = a + b_1 x_{i1} + b_2 x_{i2} + ... + b_d x_{id} + \epsilon_i \tag{1.1}$$

where $(b_1, b_2, \ldots, b_d)$ are the predictor coefficients and ϵ_i's follows $N \sim (0, \sigma^2)$ distribution.

Unlike multiple regression, in switching regression data are assumed to have come from more than one regression model. Consequently, in switching regression, instead of one regression model we consider l different models to account for each data pair $m_j = (\boldsymbol{x_j}, y_j)$. Here, $\boldsymbol{x_j} \in \mathbb{R}^d$, $y \in \mathbb{R}$ and n is the number of data points. Here we assume the following holds:

$$y_j = f_i(\boldsymbol{x_j}; \boldsymbol{\beta}_j) + \epsilon_i, \qquad 1 \leq i \leq l \tag{1.2}$$

where $\boldsymbol{\beta_i}$ is the predictor coefficient of i^{th} model and ϵ_i's are random noise, having zero mean and variance σ_i^2, such that the j^{th} data point m_j has come from the i^{th} model. The probability density function of ϵ_i can be written as,

$$p(\epsilon_i; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\epsilon_i - \mu)^2}{2\sigma^2}} \tag{1.3}$$

Switching regression has been widely studied and applied in economics, social science and engineering. Let us discuss some application of switching regression studies in various research papers. Hosmer[2] illustrates switching regression using the following example. The sex of halibut fish, a major commercial catch, cannot be determined without dissecting it. After these fish have been cleaned, their length and age are measured. We say that a sample is labeled if its sex (population of origin) is known, otherwise the sample is unlabelled. The fisheries department wants to determine estimates of the mean and standard deviation of length and proportion of each sex for each year. It is known that the average length of a female halibut for a particular age is greater than that of its male counterpart and this difference increases with age. Therefore, the average length of each sex can be represented using a density function of age and its parameter estimated by maximum likelihood approach. Assuming the function is linear in age and $l = 2$, the distribution of length may be represented as

$$\begin{aligned} y &= f_1(x, \boldsymbol{\beta_1}) + \epsilon_1 = \beta_{11}x + \beta_{12} + \epsilon_1 \\ y &= f_2(x, \boldsymbol{\beta_2}) + \epsilon_2 = \beta_{21}x + \beta_{22} + \epsilon_2 \end{aligned} \tag{1.4}$$

where y = length, x = age. The values of ϵ_1 and ϵ_2 are independent for different data, and distributions of ϵ_1 and ϵ_2 are normal with mean 0 and (unknown) standard deviations σ_1 and σ_2 respectively. In this example, if the data were labeled then the parameter estimation would be very simple.

But here the data are unlabeled. A maximum likelihood based method coupled with expectation maximization algorithm can be employed to obtain the parameters of eq(1.4). Given length and age, the sex of an unknown fish can now be predicted using the estimated switching regression model. Alternatively, the mean length of either the male or female catches over a year which can also be determined. The information that for any age the average length of females exceeds the average length of male and this difference increases with age is vital. It should be noted that estimation of the mean and standard deviation of length and proportion of each sex for each year is not possible without this information.

In another study[14] to understand the factors that determine the wages in public and private sectors, a switching regression model was employed. Two separated equations of wage rate each for public and private sector were assumed. Data used here are labeled and the predictor coefficients were estimated using ordinary least squared error method.

One method of estimating the predictor coefficients of the switching regression model is to formulate the problem using mixture distribution. Using the fish example case of eq(1.4), the switching regression (also called mixture of regression) model can be written as:

$$y_i = \begin{cases} f_1(\boldsymbol{x_i}, \boldsymbol{\beta_1}) + \epsilon_{1i} & \text{with probability } \lambda, \\ f_2(\boldsymbol{x_i}, \boldsymbol{\beta_2}) + \epsilon_{2i} & \text{with probability } 1 - \lambda, \end{cases} \tag{1.5}$$

where $\epsilon_{ji} \sim N(0, \sigma_j^2)$ are independent, $j = 1, \ldots, l; i = 1, \ldots, n$. Given an x_i, density of y_i can be written as,

$$p(y_i) = \lambda p_1(y_i) + (1 - \lambda) p_2(y_i) \tag{1.6}$$

where $p_j(y_i) \sim N(f_1(x_i, \beta_j), \sigma_j^2), j = 1, 2$. The parameter vector of this model is denoted by $\theta = (\lambda,\ \beta_{11},\ \beta_{12},\ \beta_{21},\ \beta_{22},\ \sigma_1,\ \sigma_2)$. It is a mixture of normals.

Day[3] (1969) described an iterative method to obtain the maximum likelihood estimators for the case of equal variances for this kind of mixture distribution. Later, Hathaway[7] showed that the method employed by Day[3] is an instance of expectation maximization (EM)[5] algorithm. In the EM setup, the probability of data to the two component populations are equal to the current conditional probability of component's membership.

A second approach for estimation of the parameters $\{\beta_{ij}\}$ in eq(1.4), where data are unlabeled, is by first finding a crisp l partition using any conventional clustering algorithm and then separately solve each of the l models using the partitioned data.

1.1 Fuzzy c-Regression Model (FCRM)

Hathaway et al.[6] suggested an alternative framework to fit switching regression models. It is called Fuzzy c-Regression Model (FCRM). It gives a fuzzy cluster wise regression model of the data based on how well the dependent variable of each data sample is approximated by the switching models. We again refer to the fishery example to discuss FCRM. As evident in eq(1.5), the classical approach of solving switching regression problem using mixture distribution makes a distributional assumption on ϵ_i. Fuzzy c-Regression Model makes no such assumption, instead the estimates are obtained by minimizing an objective function which consists of measure of the error done by a model in predicting the dependent variable. The measure of error in $f_i(\boldsymbol{x_k}; \boldsymbol{\beta_i})$ as an approximation to y_k is denoted by $E_{ik}(\boldsymbol{\beta_i})$. Lets define fuzzy l-partition (U) of the data, such that, U_{ik} is the membership of k^{th} data sample in the i^{th} fuzzy cluster of the dataset. For switching regression problem, we interpret U_{ik} as the importance or weight attached to the extent to which the prediction model $f_i(\boldsymbol{x_k};\boldsymbol{\beta_i})$ matches y_k, i.e., the degree of membership of $\boldsymbol{x_k}$ to the i^{th} model. This permits an observation partially belong to more than one regression model. Crisp membership place all weight to one model for each k in the approximation of y_k by $f_i(\boldsymbol{x_k};\boldsymbol{\beta_i})$. This approach formulates the two problems (fuzzy partition of the data and estimate $\beta = \{\boldsymbol{\beta_1},\ldots,\boldsymbol{\beta_l}\}$) so that a simultaneous solution may be obtained. The general family of fuzzy c regression models objective functions is defined as,

$$\begin{aligned} &\min J_m(U,\beta) = \sum_{k=1}^{n}\sum_{i=1}^{l} U_{ik}^{m} E_{ik}(\beta_i), \\ &\text{subject to } \sum_{i=1}^{l} U_{ik} = 1, \forall k = 1,\ldots,n; 0 \leq U_{ik} \leq 1; 0 < \sum_{k} U_{ik} < n, \forall i. \end{aligned} \tag{1.7}$$

The regression parameters (β_is) are learnt in conjunction with the fuzzy partition (U) using the FCRM algorithm[6]. Hathaway et al. argued that FCRM converges more rapidly than EM, but sometimes terminated at undesirable estimates when poor initializations of U matrix were used. But the two methods produced very similar quality results on the datasets that they tested.

There are several points of difference between the classical mixture distribution approach to switching regression and the FCRM approach. In FCRM, the fuzzy l-partition matrix consists of the membership values (U_{ik}), which is a measure of degree of belongingness of the feature vector $\boldsymbol{x_k}$ to i^{th} cluster/regression model. In the classical case of estimation by mixture distribution, the following can be interpreted as the posterior probability that $\boldsymbol{x_k}$ came for i^{th} model [4]

$$w_{ik} = \frac{\lambda_j p_j(x_k)}{\sum_{i=1}^{l} \lambda_i p_i(x_k)} \tag{1.8}$$

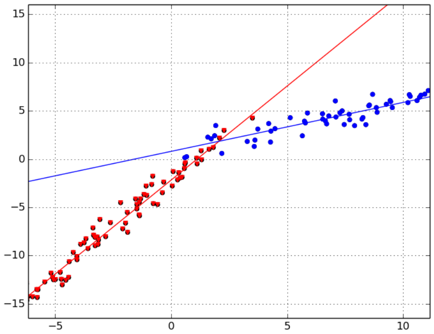
We emphasize that in certain scenario, interpretation of result is more meaningful when FCRM is employed for fitting data. Let us illustrate the last point using an example. Consider the case where we want to cluster an object into two clusters and it has characteristics of both the cluster, say a pixel in satellite image. Here fuzzy membership of the object belonging to each class is more meaningful than speaking about probability that the object came from either of the two clusters. For example, in a problem of clustering individuals into artist and scientist, diVinci might be assigned the memberships (0.6, 0.4). On the other hand, saying that he was a artist with probability 0.6 and scientist with 0.4 is not appealing!

In other cases, the benefit of using fuzzy c-regression model is modelling advantage. Let us consider the study[11] which aims at understanding the determinants of yield spread on new agency bonds and Treasury securities, which uses a switching regression model. It deals with the case of a single explanatory variable whose relationship with the dependent variable changed at some point in time. The problem of estimating the breakpoint in the coefficient of independent variable, as mentioned in the work, involves the use of a multiplicative dummy variable. We observed that an alternative way is to fit a fuzzy c-regression model. Figure 1 shows a scenario where FCRM identifies the true model, and hence the breakpoint. Data for the figure is generated synthetically. It should be noted that the data for a particular model were generated only upto a threshold of the independent variable. Data for the other model was generated when the independent variable took values greater than the threshold. Note that, there could be many switching points. In Table 1, we show the terminal objective values of 3 runs to emphasize that such algorithms sometimes may land at poor minima. Hence, it is necessary to run a few times for different initial conditions and use the best solution.

Table 1.1: FCRM result on dataset with a breakpoint

	Objective value (Run-1)	Objective value (Run-2)	Objective value (Run-3)
Iteration-1	807.83	210.63	210.63
Iteration-2	210.63	-	-

Figure 1.1: Regression clusters obtained by FCRM



1.2 Motivation of our work

Suk and Hwang[12] proposed a method, called regularized fuzzy clusterwise ridge regression,. It combines ridge regression with regularized fuzzy clustering in a unified framework. The method is useful in handling potential multicollinearity among predictor variables.

In supervised learning, let us consider the case where input feature space is large, but a small subset of the features is sufficient to approximate the target concept. Special mechanisms such as regularization, which forces the parameters to be small in magnitude, is usually employed to simplify the model. Such a mechanism also helps to reduce overfitting of the learning algorithm[9].

For switching regression, although there are evidences/reasons to believe that the data are generated by more than one model, usually it is not known whether all predictors are important for all regimes. Identification of unnecessary predictor variables brings more transparency to the models and result in better understanding of the underlying process. So, our objective is to find useful predictor independent variables with a view to obtain more transparent models. Before presenting how irrelevant variables can be eliminated for fuzzy switching regression, we discuss how to solve the problem for ordinary regression.

Chapter 2

Methods Proposed

2.1 Lasso for ordinary regression

Suppose that we have data set $(X, \boldsymbol{y}) = \{(\boldsymbol{x_i}, y_i); i = 1, 2, \ldots, N\}$, where $\boldsymbol{x_i} = (x_{i1}, ..., x_{id})^T$ is the vector of predictor variables and y_i is the response variable. We assume that x_{ij} are standardized. So, the intercept term can be neglected. Letting $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_d)^T$, the least absolute shrinkage and selection operator (lasso) estimate of $(\hat{\boldsymbol{\beta}})$ is obtained by solving the following optimizatiom problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{d} \beta_j x_{ij})^2, \qquad \text{subject to} \sum_{j=1}^{d} |\beta_j| \leq t \tag{2.1}$$

The above constrained optimization problem can be transformed into an equivalent unconstrained one

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} (y_i - \sum_{j=1}^{d} \beta_j x_{ij})^2 + \lambda (\sum_{j=1}^{d} |\beta_j|) \right\} \tag{2.2}$$

where λ is the Lagrangian multiplier. Lasso penalty in high dimension will bias towards sparse solutions[13] thereby increasing interpretability of model.

It exhibits stability of ridge regression. We shall now extend the idea of lasso to fuzzy c-regression model.

2.2 Lasso for fuzzy c-regression model

The lasso regularizer when applied to the FCRM objective function can be writen as

$$\begin{aligned} &\underset{\{\boldsymbol{\beta_i}\},U}{\operatorname{argmin}} \quad J_m(U,\{\boldsymbol{\beta_i}\}) = \underset{\{\boldsymbol{\beta_i}\},U}{\operatorname{argmin}} \quad \sum_{i=1}^{l}\Bigg(\sum_{k=1}^{n} u_{ik}^m E_{ik}(\boldsymbol{\beta_i})\Bigg) \\ &\text{subject to } \big(\sum_{j=1}^{d}|\beta_{ij}|\big) \leq t_i; i=1,\ldots,l; \\ &\sum_{i=1}^{l} u_{ik} = 1, \forall k = 1,\ldots,n; 0 \leq u_{ik} \leq 1; 0 < \sum_{k} u_{ik} < n, \forall i. \\ &where \\ &E_{ik}(\boldsymbol{\beta_i}) = (y_k - (\boldsymbol{x_k})^T\boldsymbol{\beta_i})^2 \end{aligned} \tag{2.3}$$

Equation (2.3) contains a number of variables. These are defined below.

$$X = \{\boldsymbol{x_1^T}, \boldsymbol{x_2^T}, ..., \boldsymbol{x_n^T}\}^T = \text{ the data } (n \times d) \text{ matrix,} \tag{2.4a}$$

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}, ..., x_{id})^T = \text{feature vector, } x_{ij} \text{ are standardized, } x_{ij}\epsilon R \tag{2.4b}$$

$$\boldsymbol{y} = \{y_1, y_2, ..., y_n\}^T \subset \boldsymbol{R^n} = \text{ dependent variable} \tag{2.4c}$$

$$l = \text{number of models; } 2 \leq l < n, \tag{2.4d}$$

$$\boldsymbol{\beta_i} = (\beta_{i1}, \beta_{i2}, ..., \beta_{id})^T = \ i^{th} \text{ model } (d \times 1) \text{ parameter vector,} \tag{2.4e}$$

$$U = \text{ fuzzy partition } (l \times n) \text{ matrix of the data,} \tag{2.4f}$$

$$u_{ik} = \text{degree of membership of data point } \boldsymbol{x_k} \text{ in } i^{th} \text{ regression model,} \tag{2.4g}$$

$$m = \text{fuzzifier/weighing components, } 1 < m < \infty \tag{2.4h}$$

For a given $\{\boldsymbol{\beta_i}\}$, the update equation for u_{ik} remains the same as that of FCRM. Since the optimization algorithm for (2.3) alternates between update of U (for a fixed $\{\boldsymbol{\beta_i}\}$) and update of $\{\boldsymbol{\beta_i}\}$ for a fixed U, from now on, for notational simplicity, we shall not include U as a parameter for optimization.

Following is the equivalent unconstrained objective function of the optimization problem in eq(2.3).

$$J_m(U, \{\boldsymbol{\beta_i}\}) = \sum_{i=1}^{l} \sum_{k=1}^{n} \left\{ u_{ik}^m E_{ik}(\boldsymbol{\beta_i}) + \lambda_i \big(\sum_{j=1}^{d} |\beta_{ij}| \big) \right\} \tag{2.5}$$

So, the predictor coefficients of regularized FCRM with lasso are the one which minimizes the objective function in eq(2.5). We rewrite eq(2.5) in matrix notation by introducing new variables.

$$\underset{\{\boldsymbol{\beta_i}\}}{\operatorname{argmin}}\, J_m(U, \{\boldsymbol{\beta_i}\}) = \underset{\{\boldsymbol{\beta_i}\}}{\operatorname{argmin}} \sum_{i=1}^{l} \left\{ \|D_i \boldsymbol{y} - D_i X \boldsymbol{\beta_i}\|_2^2 + \lambda_i \|\boldsymbol{\beta_i}\|_1 \right\} \tag{2.6}$$

where $D_i = \left[d_{kk}^i\right]_{n \times n}$ is a $(n \times n)$ diagonal matrix whose k^{th} diagonal entry $d_{kk}^i = (u_{ik})^{\frac{1}{2}}$; λ_i is the Lagrangian multiplier of the i^{th} model associated with the ℓ_1 penalty. Given a fixed U eq(2.6) can be separated into l objective functions and the predictor coefficients of all l models can be obtained by optimizing each one of them separately. This is possible because for a fixed U the parameters of a particular model $\boldsymbol{\beta_i}$ occur with objective function of one model.

The optimization problem in eq(2.6) is a bound-constrained quadratic programming that can be solved using Gradient Projection for Sparse Reconstruction (GPSR)[10] method. This is done by splitting the variable $\boldsymbol{\beta_i}$ into its positive and negative parts[10]. Formally, we introduce the vectors $\boldsymbol{v}$ $(d \times 1)$ and $\boldsymbol{w}$ $(d \times 1)$ and make the substitution

$$\boldsymbol{\beta} = \boldsymbol{v} - \boldsymbol{w}; \quad \boldsymbol{v} \geq \boldsymbol{0}; \quad \boldsymbol{w} \geq \boldsymbol{0}; \quad i = 1, ..., l \tag{2.7}$$

Note that, for each $\boldsymbol{\beta_i}$, there are two sets of variables $\boldsymbol{v_i}$ and $\boldsymbol{w_i}$. Here, the following relations are satisfied by $\boldsymbol{v_i} = (\boldsymbol{\beta_i})_+$ and $\boldsymbol{w_i} = (-\boldsymbol{\beta_i})_+$ for all $i = 1, \ldots, d$, where $(x)_+ = \max\{0, x\}$. Since each problem is solved separately, for notational clarity, we have dropped the subscript i from $\boldsymbol{v}$ and $\boldsymbol{w}$. Thus we have $\|\boldsymbol{\beta_i}\|_1 = \mathbf{1}_d^T\boldsymbol{v} + \mathbf{1}_d^T\boldsymbol{w}$, where $\mathbf{1}_d = [1, 1, ..., 1]^T$ is the vector consisting of d ones, so eq(14) can be written as the following bound-constrained quadratic program:

$$\begin{aligned} \underset{\boldsymbol{v},\boldsymbol{w}}{\operatorname{argmin}} \quad & \|D_i\boldsymbol{y} - D_iX(\boldsymbol{v} - \boldsymbol{w})\|_2^2 + \lambda\mathbf{1}_d^T\boldsymbol{v} + \lambda\mathbf{1}_d^T\boldsymbol{w} \\ \text{subject to} \quad & \boldsymbol{v} \geq \mathbf{0}, \boldsymbol{w} \geq \mathbf{0} \end{aligned} \tag{2.8}$$

Substituting $\boldsymbol{b} = D_i\boldsymbol{y}$, $A = D_iX$ in eq(2.8), it can rewritten into a standard bound-constrained quadratic (BCQP) program. Following are the steps for conversion of the objective function in eq(2.8) into the objective function of standard BCQP. We can discount $\boldsymbol{b}^T\boldsymbol{b}$ from the objective function in the derivation below since it is a constant.

$$
\begin{aligned}
&\|\boldsymbol{b} - A(\boldsymbol{v} - \boldsymbol{w})\|_2^2 + \lambda \mathbf{1}_d^T \boldsymbol{v} + \lambda \mathbf{1}_d^T \boldsymbol{w} \\
&= \big(\boldsymbol{b} - A(\boldsymbol{v} - \boldsymbol{w})\big)^T \big(\boldsymbol{b} - A(\boldsymbol{v} - \boldsymbol{w})\big) + \lambda \mathbf{1}_{2d}^T \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} \\
&= \big(\boldsymbol{b}^T \boldsymbol{b} - 2\boldsymbol{b}^T A(\boldsymbol{v} - \boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T A^T A(\boldsymbol{v} - \boldsymbol{w})) + \lambda \mathbf{1}_{2d}^T \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} \\
&= \big(- 2\boldsymbol{b}^T A(\boldsymbol{v} - \boldsymbol{w}) + (\boldsymbol{v} - \boldsymbol{w})^T A^T A(\boldsymbol{v} - \boldsymbol{w})) + \lambda \mathbf{1}_{2d}^T \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} \\
&= (\boldsymbol{v} - \boldsymbol{w})^T A^T A(\boldsymbol{v} - \boldsymbol{w}) + (-2\boldsymbol{b}^T A)\boldsymbol{v} + (2\boldsymbol{b}^T A)\boldsymbol{w} + \lambda \mathbf{1}_{2d}^T \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} \\
&= (\boldsymbol{v} - \boldsymbol{w})^T A^T A(\boldsymbol{v} - \boldsymbol{w}) + \begin{pmatrix} -2\boldsymbol{b}^T A & 2\boldsymbol{b}^T A \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} + \lambda \mathbf{1}_{2d}^T \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{v}^T & \boldsymbol{w}^T \end{pmatrix} \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} + \Big\{ \begin{pmatrix} -2\boldsymbol{b}^T A & 2\boldsymbol{b}^T A \end{pmatrix} + \lambda \mathbf{1}_{2d}^T \Big\} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}
\end{aligned}
\tag{2.9}
$$

Rewriting the optimization problem in eq(2.8) using the new objective function form of eq(2.9).

$$
\begin{aligned}
\underset{\boldsymbol{v},\boldsymbol{w}}{\text{argmin}} \quad & \Big\{ \begin{pmatrix} -2\boldsymbol{b}^T A & 2\boldsymbol{b}^T A \end{pmatrix} + \lambda \mathbf{1}_{2d}^T \Big\} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} + \\
& \begin{pmatrix} \boldsymbol{v}^T & \boldsymbol{w}^T \end{pmatrix} \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix} \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix} \\
\text{subject to} \quad & \boldsymbol{v} \geq \mathbf{0}, \boldsymbol{w} \geq \mathbf{0}
\end{aligned}
\tag{2.10}
$$

Comparing eq(2.10) with the following standard bound-constrained quadratic program (BCQP) form,

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{argmin}} \quad & \boldsymbol{c}^T\boldsymbol{z} + \frac{1}{2}\boldsymbol{z}^T B\boldsymbol{z} \\ \text{subject to} \quad & \boldsymbol{z} \geq \boldsymbol{0} \end{aligned} \tag{2.11}$$

we have,

$$\begin{aligned} &\boldsymbol{z} = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}, \quad \boldsymbol{c} = \lambda \boldsymbol{1}_{2d} + \begin{pmatrix} -\boldsymbol{a} \\ \boldsymbol{a} \end{pmatrix}, \quad \boldsymbol{a} = 2(D_iX)^T(D_i\boldsymbol{y}) \\ &B = \begin{pmatrix} 2(D_iX)^T(D_iX) & -2(D_iX)^T(D_iX) \\ -2(D_iX)^T(D_iX) & 2(D_iX)^T(D_iX) \end{pmatrix} \end{aligned} \tag{2.12}$$

Note that we have resubstituted $\boldsymbol{b} = D_i\boldsymbol{y}$ and $A = D_iX$ above. This problem now can be solved using any package that supports solution to BCQP. Here the solution for regularized FCRM with lasso is obtained using python optimization package.

The algorithm used to obtain the estimates of predictor coefficients of the regularized FCRM with ℓ_1 penalty is similar to unregularized FCRM algorithm[6]. In ordinary FCRM, for a given partition matrix U, the FCRM objective function is optimized with respect to predictor coefficients $\boldsymbol{\beta_i}$'s and estimators of the predictor coefficients are obtained. However, in regularized FCRM case, the FCRM objective function is replaced by regularized objective function in eq(2.5). Following is the formal algorithm used to obtained the estimates of regularized RFCRM with ℓ_1 penalty.

Algorithm 1: Regularised FCRM with ℓ_1 regularizer

```
1 Given data $S = \{(\boldsymbol{x_1}, y_1), ..., (\boldsymbol{x_n}, y_n)\}$. Select $m > 1$, specify
  regression model of the form in eq(1.2), and choose a measure of error
  $E = \{E_{ik}\}$ so that $E_{ik}(\boldsymbol{\beta_i}) \geq 0$ for $i^{th}$ model and $k^{th}$ sample. Initialize
  a termination threshold $\epsilon > 0$ and an initial partition of data $U^{(0)}$.
  Then for $r = 0, 1, 2, ...$ ;
2 Calculate the $l$ model parameters $\boldsymbol{\beta_i} = \boldsymbol{\beta_i^{(r)}}$that globally minimizes
  the unrestricted objective function in eq(2.5). As discussed before
  each of the $l$ objective functions are separately fed into a quadratic
  programming solver and the corresponding $l$ predictor coefficients are
  obtained ;
3 Update $U^{(r+1)} = U^{(r)}$, with $E_{ik} = E_{ik}(\boldsymbol{\beta_i^{(r)}})$, such that
   $$\begin{aligned} &U_{ik} = \frac{1}{\sum_{j=1}^{l} \big(\frac{E_{ik}}{E_{jk}}\big)^{\frac{1}{m-1}}} \text{ if } E_{ik} > 0 \text{ for } 1 \geq i \geq l \\ &\text{otherwise ,} \\ &\text{if } E_{ik} = 0 \text{ for exactly one i then } , U_{ik} = 1 \text{ and} \\ &U_{jk} = 0 \text{ if } E_{jk} > 0, \text{ so that } U_{jk} \in [0, 1] \\ &\text{with } (U_{1k} + ... + U_{lk}) = 1 \\ &\text{else if } E_{ik} = 0 \text{ for more than one } i, \text{ i.e. for } i \in I = \{i_1, i_2, \ldots, i_z\} \text{ then,} \\ &U_{ik} = 0 \text{ for all } i \text{ with } E_{ik} > 0 \\ &\text{and } U_{jk} = \alpha_j, \alpha_j > 0, \text{ with } \sum_{j \in I} \alpha_j = 1 \end{aligned} \tag{2.13}$$
   ;
4 Calculate $tolerance = \|U^{(r)} - U^{(r+1)}\|$ ;
5 Check for termination, if $tolerance \leq \epsilon$, then stop; otherwise set
  $r = r + 1$ and return to step 2.
```

Also when there is multicolinearity, least squares estimates have a high variance. Use of ℓ_2 regularizer helps to deal with this problem. ℓ_2 shrinks the coefficients but doesnot reduce to zero.

2.3 Combined ℓ_1 and ℓ_2-norm

If number of features is much larger than the number of samples ($p >> n$), lasso can select almost n variables[16]. When there are some highly correlated variables, lasso tends to select one of them and rejecting the others. In such a case the variable selection can be unstable as with minor change in data the selected variable may be different. For usual $n > p$ situations, if there exist high correlations among predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression[13]. ℓ_2-norm is able to avoid outliers and the ℓ_1-norm is helpful for achieving the sparseness, which are both beneficial to accurate classification[15]. We, therefore, analyse the model parameter estimates obtained using regularized FCRM with combined ℓ_1 and ℓ_2-norm. Such models are sometimes called *elastic net.* The unconstrained objective function for regularized FCRM with combined ℓ_1 and ℓ_2 regularizer is

$$\underset{\{\boldsymbol{\beta_i}\}}{\operatorname{argmin}}\, J_m(U, \{\boldsymbol{\beta_i}\}) = \underset{\{\boldsymbol{\beta_i}\}}{\operatorname{argmin}} \sum_{i=1}^{l} \left\{ \|D_i\boldsymbol{y} - D_iX\boldsymbol{\beta_i}\|_2^2 + \lambda_i\|\boldsymbol{\beta_i}\|_1 + \gamma_i\|\boldsymbol{\beta_i}\|_2^2 \right\} \tag{2.14}$$

where $\|.\| = \ell_2$ norm, λ and γ are the Lagrangian multipliers of the ℓ_1 and ℓ_2 regularizer respectively. The optimization problem in eq(2.14) can similarly be written in standard bound constrained quadratic program form

$$
\begin{aligned}
&\underset{\boldsymbol{z}}{\text{argmin}} \qquad \boldsymbol{c}^T\boldsymbol{z} + \frac{1}{2}\boldsymbol{z}^T B\boldsymbol{z} \\
&\text{subject to} \quad \boldsymbol{z} \geq \boldsymbol{0} \\
&\textit{where} \\
&\boldsymbol{z} = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{w} \end{pmatrix}, \quad \boldsymbol{a} = 2A^T\boldsymbol{b}, \quad \boldsymbol{c} = \lambda_i \mathbf{1}_{\boldsymbol{2d}} + \begin{pmatrix} -\boldsymbol{a} \\ \boldsymbol{a} \end{pmatrix} \\
&B = \begin{pmatrix} 2\big\{(D_iX)^T(D_iX) + \gamma_i I\big\} & -2\big\{(D_iX)^T(D_iX) + \gamma_i I\big\} \\ -2\big\{(D_iX)^T(D_iX) + \gamma_i I\big\} & 2\big\{(D_iX)^T(D_iX) + \gamma_i I\big\} \end{pmatrix}
\end{aligned}
\tag{2.15}
$$

Here I is $(d \times d)$ identity matrix. We have used Algorithm-2 to obtain the parameter estimates for regularized FCRM with combined ℓ_1 and ℓ_2 norm. Algorithm-2 is similar to the ℓ_1 regularized FCRM algorithm described earlier with the modification of objetive functsion in step(2).

Algorithm 2: Regularised FCRM with combined ℓ_1 and ℓ_2 regularizer

```
1 Given data $S = \{(\boldsymbol{x_1}, y_1), ..., (\boldsymbol{x_n}, y_n)\}$. Select $m > 1$, specify
  regression model of the form in eq(1.2), and choose a measure of error
  $E = \{E_{ik}\}$ so that $E_{ik}(\boldsymbol{\beta_i}) \geq 0$ for $i^{th}$ model and $k^{th}$ sample. Initialize
  a termination threshold $\epsilon > 0$ and an initial partition of data $U^{(0)}$.
  Then for $r = 0, 1, 2, ...$ ;
2 Calculate the $l$ model parameters $\boldsymbol{\beta_i} = \boldsymbol{\beta_i^{(r)}}$that globally minimizes
  the unrestricted objective function in eq(2.5). As discussed before
  each of the $l$ objective functions are separately fed into a quadratic
  programming solver and the corresponding $l$ predictor coefficients are
  obtained ;
3 Update $U^{(r+1)} = U^{(r)}$, with $E_{ik} = E_{ik}(\boldsymbol{\beta_i^{(r)}})$, such that
  $U_{ik} = \frac{1}{\sum_{j=1}^{l} (\frac{E_{ik}}{E_{jk}})^{\frac{1}{m-1}}}$ if $E_{ik} > 0$ for $1 \geq i \geq l$
  otherwise ,
  if $E_{ik} = 0$ for exactly one i then ,$U_{ik} = 1$ and
  $U_{jk} = 0$ if $E_{jk} > 0$, so that $U_{jk} \in [0, 1]$
  with $(U_{1k} + ... + U_{lk}) = 1$
  else if $E_{ik} = 0$ for more than one $i$, i.e. for $i \in I = \{i_1, i_2, \ldots, i_z\}$ then,
  $U_{ik} = 0$ for all $i$ with $E_{ik} > 0$
  and $U_{jk} = \alpha_j, \alpha_j > 0$, with $\sum_{j \in I} \alpha_j = 1$          (2.16)
  ;
4 Calculate $tolerance = \|U^{(r)} - U^{(r+1)}\|$ ;
5 Check for termination, if $tolerance \leq \epsilon$, then stop; otherwise set
  $r = r + 1$ and return to step 2.
```

2.4 Learning of feature weight: Garrotte estimates

In this section we shall generalize another feature selection method which explicitly finds a weight for each variable, to fuzzy switching regression.

2.4.1 Ordinary regression case:

As discussed earlier, subset selection is a method of selecting a subset of features which does a good job in prediction to the target concept. Breimen (1995)[1] developed a method for subset selection in ordinary regression problems. Suppose that we have a data set $(X, \boldsymbol{y}) = \{(\boldsymbol{x_i}, y_i); i = 1, 2, \ldots, N\}$, where $\boldsymbol{x_i} = (x_{i1}, ..., x_{id})^T$ is a vector of predictor variables and y_i's are the responses. Let $\{\hat{\boldsymbol{\beta}}\}$ be the original OLS estimates. Let $c_k \geq 0$ be the weight of k^{th} variable indicating its importance. The problem is to find find $\boldsymbol{c} = \{c_k\}$ to minimize

$$J(\boldsymbol{c}, \boldsymbol{\beta}) = \sum_n \left(y_n - \sum_{k=1}^{d} c_k \hat{\beta}_k x_{nk} \right)^2 \text{ such that } \; c_k \geq 0, \sum_k c_k \leq s \qquad (2.17)$$

After having obtained $\boldsymbol{c} = \{c_1, ..., c_d\}$ which minimizes eq(2.17), $\tilde{\beta}_k(s) = c_k \hat{\beta}_k$ is calculated, which are the new predictor coefficients for the k^{th} feature. Higher the value of s, more the $\{c_k\}$ nears zero and therefore $\tilde{\beta}_k(s)$ are shrunken. This procedure is called *non-negative garrotte*[1]. Here we explore the possibility of extending this concept for successful feature selection in a fuzzy switching regression paradigm.

2.4.2 Fuzzy Switching Regression case

Let us call the vector $\boldsymbol{c} = \{c_1, ..., c_d\}$ as the feature weights/multipliers. These weights are multiplied to predictor coefficients of a model. We explore two cases. Case 1: there are l set of feature weight vectors, one for each model of the switching regression. Case 2: there is just one feature weight vector for all l models. We explore these two cases by separate algorithms towards the end of this section. We now formulate objective function regularized FSRM with *non-negative garrotte*. The optimization problem for case 1 can therefore be written as,

$$\begin{aligned} &\underset{\{\boldsymbol{c_i}\}}{\operatorname{argmin}} J_m(U, \{\boldsymbol{c_i}\}) = \underset{\{\boldsymbol{c_i}\}}{\operatorname{argmin}} \sum_{i=1}^{l} \left(\sum_{k=1}^{n} u_{ik}^m \big(y_k - \sum_{j=1}^{d} c_{ij} \hat{\beta}_{ij} x_{kj}\big)^2 \right) \\ &\text{subject to} \quad \boldsymbol{c_{ij}} \geq 0, \forall i = 1, \ldots, l \text{ and } j = 1, \ldots, d; \ \sum_{j=1}^{d} c_{ij} \leq s_i. \end{aligned} \tag{2.18}$$

As done for the lasso mode, for the same reason we donot show U as a parameter for optimization to simplify notations. For notational simplicity, we write $\boldsymbol{c_i} \geq \mathbf{0}$ to indicate $\boldsymbol{c_{ij}} \geq 0, \forall j = 1, \ldots, d$ After introducing new variables (defined later in eq(2.21)) we can rewrite the objective function in (2.18) as

$$\begin{aligned} &J_m(U, \{\boldsymbol{c_i}\}) = \sum_{i=1}^{l} \| D_i \boldsymbol{y} - D_i X B_i \boldsymbol{c_i} \|_2^2 \\ &\text{such that} \quad \boldsymbol{c_i} \geq \mathbf{0}, \forall i = 1, .., l; \ \sum_{j=1}^{d} c_{ij} \leq s_i, \forall i = 1, \ldots, l. \end{aligned} \tag{2.19}$$

Equation(2.19) can be converted into an equivalent unconstrained optimization problem by introducing l Lagrangian multipliers. The optimization problem of the i^{th} model of the switching regression thus becomes,

$$\underset{\boldsymbol{c_i}}{\operatorname{argmin}} \quad J_m(U, \boldsymbol{c_i}) = \underset{\boldsymbol{c_i}}{\operatorname{argmin}} \quad \left\{ \| D_i \boldsymbol{y} - D_i X B_i \boldsymbol{c_i} \|_2^2 + \lambda_i \mathbf{1}_{\boldsymbol{d}}^{\boldsymbol{T}} \boldsymbol{c_i} \right\} \tag{2.20}$$
$$\text{subject to} \quad \boldsymbol{c_i} \geq \mathbf{0}.$$

The new variables used in eq(2.19) and (2.20) are defined below.

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{i}} = (\hat{\beta}_{i1}, ..., \hat{\beta}_{id})^T = \ i^{th} \text{ cluster coefficient estimated by FCRM,} \tag{2.21a}$$

$$\boldsymbol{c_i} = (c_{i1}, c_{id}, ..., c_{id})^T = \ \text{vector of feature weights for } i^{th} \text{ model} \tag{2.21b}$$

$$B_i = \left[b^i_{kk} \right]_{d \times d} = (d \times d) \text{ diagonal matrix constructed using } b^i_{kk} = \beta_{ik} \tag{2.21c}$$

The other terms have the usual meaning. The new predictor coefficients of the i^{th} model are $\tilde{\beta}_{ik}(\lambda) = c_{ik}\hat{\beta}_{ik}, \forall k = 1, ..., d; \ \forall i = 1, ..., l$.

Following are the steps for conversion of eq(2.20) into a standard bound constrained quadratic program; letting $\boldsymbol{b} = D_i\boldsymbol{y}; A = D_iXB_i; \boldsymbol{c} = \boldsymbol{c_i}, \lambda = \lambda_i$ we rewrite eq(2.20) as,

$$\begin{aligned}
&\underset{\boldsymbol{c}}{\operatorname{argmin}} \ \ \|\boldsymbol{b} - A\boldsymbol{c}\|_2^2 + \lambda \mathbf{1}_d^T \boldsymbol{c} \\
=&\underset{\boldsymbol{c}}{\operatorname{argmin}} \ \ \big(\boldsymbol{b} - A\boldsymbol{c}\big)^T \big(\boldsymbol{b} - A\boldsymbol{c}\big) + \lambda \mathbf{1}_d^T \boldsymbol{c} \\
=&\underset{\boldsymbol{c}}{\operatorname{argmin}} \ \ \big(\boldsymbol{b}^T\boldsymbol{b} - \boldsymbol{b}^T A\boldsymbol{c} - \boldsymbol{c}^T A^T \boldsymbol{b} + \boldsymbol{c}^T A^T A \boldsymbol{c}\big) + \lambda \mathbf{1}_d^T \boldsymbol{c} \\
=&\underset{\boldsymbol{c}}{\operatorname{argmin}} \ \ \big(- 2\boldsymbol{b}^T A\boldsymbol{c} + \boldsymbol{c}^T A^T A\boldsymbol{c}\big) + \lambda \mathbf{1}_d^T \boldsymbol{c} \\
=&\underset{\boldsymbol{c}}{\operatorname{argmin}} \ \ \boldsymbol{c}^T A^T A\boldsymbol{c} + (-2\boldsymbol{b}^T A + \lambda \mathbf{1}_d^T)\boldsymbol{c}
\end{aligned} \tag{2.22}$$

The standard BCQP form is,

$$\underset{\boldsymbol{z}}{\operatorname{argmin}} \ \ \frac{1}{2}\boldsymbol{z}^T B \boldsymbol{z} + \boldsymbol{a}^T \boldsymbol{z} \tag{2.23}$$
$$\text{such that} \ \ \boldsymbol{z} \geq \mathbf{0}$$

Comparing eq[2.22] and eq[2.23] we have,

$$\begin{aligned}
\boldsymbol{z} &= \boldsymbol{c} \\
\boldsymbol{a} &= \lambda \mathbf{1}_d - 2A^T\boldsymbol{b} \\
B &= 2A^TA
\end{aligned} \tag{2.24}$$

We now propose an algorithm, Algorithm 3 to obtain the model parameters of regularized FSRM with *non-negative garrotte*. In case 1, l different feature weight vectors, one for each l model, are considered. The optimization problem in eq(2.18) is separated into l parts. Each such part has the form as in eq(2.20). The algorithm loops l times to optimize over the l different models. In step 3, $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$ is initialized using the ordinary FCRM solution and then $\boldsymbol{c_i}$, the coefficient multiplier of the i^{th} model, is calculated by solving the optimization problem in eq(2.20). The conversion from the form in eq(2.20) to standard BCQP has been already derived above. For solving the problem the quadratic optimization package provided by Python Software Foundation is used. Using the new predictor coefficients ($\tilde{\beta}_k(\lambda) = c_k\hat{\beta}_k$), the membership matrix is calculated. The algorithm is said to have converged if between two consecutive iterations, the U's calculated in step 9 are similar. The matrix frobenius norm is used as a measure of this similarity. A check for convergence on the norm of the membership matrix is done. And if not converged already, the coefficient multiplier $\boldsymbol{c_i}$ is recalculated using the membership matrix which was calculated in the last step.

Algorithm 3: Algorithm to estimate parameters of regularized FSRM with *non-negative garrotte*

```
Initialize $r = 0$, membership matrix $U^r$ and $\epsilon$;
for <each model i, i=1,..,l> do
    Initialize $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$ with the ordinary FCRM solution;
    Initialize $tolerence = \epsilon$ ;
    while $tolerence \geq \epsilon$ do
        Estimate $\boldsymbol{c_i}$ by solving the optimization problem in eq(2.20)
        using $U = U^r$;
        Set $r = r + 1$ ;
        Calculate $\tilde{\beta}_{ik}(s) = c_{ik}\hat{\beta}_{ik}$;
        Calculate U using $\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{(r)}} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{i}}^{\boldsymbol{(r)}}$ and update rule in Step 3 of the
        Algorithm 1 for regularised FSRM with $\ell_1$-norm. Call this $U$ as
        $U^r$;
        Calculate $tolerence = \|U^r - U^{(r-1)}\|$ ;
    end
end
```

Algorithm 4: an alternative algorithm to estimate parameters of regularized FSRM with *non-negative garrotte*

```
Initialize membership matrix U.;
for <each model i, i=1,..,l> do
    Initialize $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$ with the ordinary FCRM solution;
    Estimate $\boldsymbol{c}_{\boldsymbol{i}}$ by solving the optimization problem in eq(2.20) ;
    Calculate $\tilde{\beta}_{ik}(s) = c_{ik}\hat{\beta}_{ik}$;
    Calculate U using $\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{(r)}} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{i}}^{\boldsymbol{(r)}}$ and update rule in Step 3 of the
    Algorithm 1 for regularised FSRM with $\ell_1$-norm. ;
end
```

We explore the possibility that with a good initialization of partition matrix U, a single iteration of step $6-9$ of Algorithm 3 would give estimates that are close to its global optimum solution. So one can think of a simplified version of Algorithm 3. We call this Algorithm 4. For this purpose, a slight alteration in Algorithm 3 is made to get Algorithm 4. Instead of repeating the estimation of $\tilde{\boldsymbol{\beta}}_{\boldsymbol{i}}$s and U until convergence, they are estimated only once. We compare the quality of estimates obtained by Algorithm 3 and 4. Later we shall compare result of these two versions.

Next we consider case 2, where a single feature weight vector is used for all the l models of the switching regression problem. The unconstrained optimization problem for the same is formulated in eq(2.25). To solve this we propose an algorithm similar to Alogrithm 3 with a slight change that the optimization problem in step 9 is replace by eq(2.25). This approach may be useful for selecting a common subset of features across all the l models.

$$\begin{aligned}&\underset{\boldsymbol{c}}{\operatorname{argmin}} \quad J_m(U, \boldsymbol{c}) = \underset{\boldsymbol{c}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{l} \left(\|D_i \boldsymbol{y} - D_i X B_i \boldsymbol{c}\|_2^2 \right) + \lambda \mathbf{1}_{\boldsymbol{d}}^{\boldsymbol{T}} \boldsymbol{c} \right\} \\ &\text{subject to} \quad \boldsymbol{c} \geq \mathbf{0}\end{aligned} \tag{2.25}$$

Algorithm 5: algorithm to estimate parameters of regularized FCRM with *single feature weight vector non-negative garrotte*

```
Initialize membership matrix $r = 0, U^r$, $\epsilon$;
Initialize $\hat{\boldsymbol{\beta}}_{\boldsymbol{i}}$ with the FCRM solution, $i = 0, .., l$;
Initialize $tolerence = \epsilon$ ;
while $tolerence \geq \epsilon$ do
    Estimate $\boldsymbol{c}$ by solving the optimization problem in eq(2.25) using
    $U = U^r$;
    Set $r = r + 1$ ;
    Calculate $\tilde{\beta}_{ik}(s) = c_k \hat{\beta}_{ik}$;
    Calculate U using $\boldsymbol{\beta}_{\boldsymbol{i}}^{\boldsymbol{(r)}} = \tilde{\boldsymbol{\beta}}_{\boldsymbol{i}}^{\boldsymbol{(r)}}$ and updation rule in Step 3 of the
    Algorithm 1 for regularised FSRM with $\ell_1$-norm. Call this $U$ as $U^r$;
    Calculate $tolerence = \|U^r - U^{(r-1)}\|$ ;
end
```

Chapter 3

Synthetic and Real World Data

3.1 Synthetic data generation

To ascertain the performance of our proposed method, we first fit our method to synthetically generated data. Four different synthetic datasets are generated.

The first dataset (**SYNT-1**) is simulated using 100 observations from each of the two models,

$$\begin{aligned} & y = \boldsymbol{\beta_1^T x} + \epsilon_1, && \text{for model-1} \\ & \textit{and} && \\ & y = \boldsymbol{\beta_2^T x} + \epsilon_2, && \text{for model-2} \end{aligned} \tag{3.1}$$

where $\boldsymbol{\beta_1} = (20, 5, 0, 0, 2)^T$, $\boldsymbol{\beta_2} = (50, 10, -7, 0, 0)^T$, ϵ_1 and ϵ_2 are noise generated by standard normal distribution with zero mean and unit variance. The dataset is 4-dimensional unnormalized and the first component of $\boldsymbol{\beta}$ is the intercept. So, X is augmented by an additional component with value 1. We construct a fuzzy c-regression model on this dataset.

A second dataset (**SYNT-**2) having **multicolinearity** is obtained using 100 observations generated from each of the two models in eq(3.1) where $\boldsymbol{\beta_1} = (20, 5, 0, 0, 2, 0)^T$, $\boldsymbol{\beta_2} = (50, 10, -7, 0, 0, 0)^T$, ϵ_1 and ϵ_2 are noise terms generated by standard normal distribution with zero mean and unit variance. In this five dimensional dataset, the first and fifth dimensions (note that the first component of $\boldsymbol{\beta}$ is the intercept term) are linearly dependent. That is, if x_{i1} and x_{i5} are respectively the first and fifth feature of the i^{th} sample then $x_{i5} = 2.2x_{i1} + \epsilon_i$ where ϵ_i follows standard normal distribution with zero mean and unit variance, $i = 1, ..., 200$. This dataset was used to fit an ordinary fuzzy c-regression model and subsequently illustrate that the later cannot detect correlated feature problem for unnormalized data. Also this un-normalized data was used to show that FCRM can get good estimates of the parameter.

A third dataset (**SYNT-**3) is generated using 100 observations from each of the two models in eq(3.1) where $\boldsymbol{\beta_1} = (20, 5, 0, 2)^T$, $\boldsymbol{\beta_2} = (50, 10, -7, 0)^T$, ϵ_1 and ϵ_2 are noise as for the other two datasets. This 4-dimensional dataset is then z-score normalized.

A fourth normalized dataset (**SYNT-**4) having dependent feature is simulated using 100 observations from each of the two models in eq(3.1) with $\boldsymbol{\beta_1} = (20, 5, 0, 2, 0)^T$, $\boldsymbol{\beta_2} = (50, 10, -7, 0, 0)^T$, ϵ_1 and ϵ_2 are $N(0, 1)$ noise. The first and fifth dimension of this 4-dimensional data are linearly dependent. That is, if x_{i1} and x_{i5} are the first and fifth features of the i^{th} sample then $x_{i5} = 2.2x_{i1} + \epsilon_i$ where ϵ_i follows $N(0, 1)$, $i = 1, ..., 200$. The dataset is then normalized using z-score normalization. **SYNT-**4 is used to explore the performance of regularized FSRM on normalized dataset having correlated feature problem.

x_{ij} is picked to be a uniform random number in $(-5, 5)$, $i = 1, \ldots, n$;

$j = 1, \ldots, l$. Next we describe a real life dataset.

3.2 Usercars data

The present data came from [8].The data were collected to predict the retail price of 804 general motors (GM) cars produced in 2005. We used six characteristics of the used cars as predictor variables: Mileage, the number of cylinders (Cylinders), engine volume (Liter), cruise control (Cruise), upgraded speakers (Speaker), and leather seats (Leather). A detailed description of the variables is provided in Table 3.1. We used six characteristics of the used cars as predictor variables: Mileage, the number of cylinders (Cylinders), engine volume (Liter), cruise control (Cruise), upgraded speakers (Speaker), and leather seats (Leather). We chose $l = 2$ as the number of clusters, $m = 2$ and $(\epsilon) = 0.0000005$. Other authors have also used this dataset for modelling by switching regression[12].

Table 3.1: variables involved in the used cars dataset

No.	Variable name	Desciption
1	Price	Suggested retail price of the used 2005 GM cars in dollars
2	Mileage	number of miles the car has been driven
3	Make	manufacturer of the car such as Saturn, Pontiac, and Chevrolet
4	Model	specific models for each car manufacturer such as Ion, Vibe, Cavalier
5	Trim (of car)	specific type of car model such as SE Sedan 4D, Quad Coupe 2D
6	Type	body type such as sedan, coupe, etc.
7	Cylinder	number of cylinders in the engine
8	Liter	a more specific measure of engine size
9	Doors	number of doors
10	Cruise	indicator variable representing whether the car has cruise control (1 = cruise)
11	Sound	indicator variable representing whether the car has upgraded speakers (1 = upgraded)
12	Leather	indicator variable representing whether the car has leather seats (1 = leather)

Chapter 4

Results

4.1 Results on synthetic dataset

The synthetic data are generated assuming $l = 2$ switching regression model in eq(3.1). Strictly speaking $m > 1$, but the limit $m \to 1^+$ leads to hard (or crisp) partitions[6]. The following initialization was used in the estimation of model parameters: m = 2, termination condition(ϵ) = 0.5×10^{-6} and U is intialized randomly. Table 4.1 shows the estimates of the unknown parameters of **SYNT**-1 dataset obtained using fuzzy c-regression models. The objective value mentioned in all the tables are calculated using

$$\text{objective value} = \sum_{i=1}^{l} \sum_{k=1}^{n} U_{ik} E_{ik}(\boldsymbol{\beta_i}). \tag{4.1}$$

To calculate the crisp objective value, each terminal partition matrix was converted to a hard partition matrix using maximum hardening membership rule as follows

$$\text{Crisp objective value} = \sum_{i=1}^{l} \sum_{\forall k \text{ in cluster } i} U_{ik}^{H} E_{ik}(\boldsymbol{\beta_i}), \tag{4.2}$$

where $U_{ik}^{H} \in \{0,1\}$ is the hardened membership value. Root mean square error (RMSE) is obtained by dividing the crisp objective using the hardened partition.

Table 4.1: unregularized FCRM on **SYNT 1**

	model parameters				true model	
β	model$-$1 $(i = 1)$	model$-$2 $(i = 2)$	objective value(crisp)	rmse	model1	model2
β_{i0}	19.851	49.889	326.92	0.638	20	50
β_{i1}	5.004	10.004	(81.457)		5	10
β_{i2}	0.005	-6.998			0	-7
β_{i3}	-0.016	-0.017			0	0
β_{i4}	2.040	-0.062			2	0

Table 4.2 shows the model parameters when FCRM is applied to the dataset **SYNT** 2 having colinearity between features. After examining Table 4.2 and Table 4.3 ($\lambda = 0$ case) it is clear that FCRM fails to detect the dependent feature problem. Note that in Tables 4.1 and 4.2 we have used un-normalized data first to show that FCRM can get good estimates of the parameter. An explanation for the same is that it is based around ordinary least squares regression[12]. Tables 4.3 and 4.4 respectively tabulate the results when regularised FCRM with ℓ_1 penalty and regularized FCRM with *non-negative garrote* was used to fit the normalised synthetic dataset (**SYNT**$-$4) having dependent features. Note that here we are using z-score normalized data. The results in Tables 4.2 and 4.3 corresponds to some typical runs. In some runs the solution could be poor due to local minima. In Table 4.8, we report the average of the regression coeffcients for *non-negative*

garrotte coefficient when applied to **SYNT-3** over 25 runs. Comparing Table 4.8 with Table 4.7, we find that the estimates are quite consistent over different runs. Results of regularised FCRM with combined ℓ_1 and ℓ_2 penalty are shown in Table 4.6. Results show that ℓ_2 penalty does not particularly help in achieving sparsity.

Table 4.2: FCRM on dataset **SYNT 2**

	model parameters				true model	
β	model-1 $(i = 1)$	model-2 $(i = 2)$	objective value(crisp)	rmse	model1	model2
β_{i0}	20.04	50.03	393.33	0.644	20	50
β_{i1}	4.70	9.76	(83.09)		5	10
β_{i2}	-0.01	-7.01			0	-7
β_{i3}	0.04	-0.003			0	0
β_{i4}	1.98	-0.05			2	0
β_{i5}	0.13	0.08			0	0

Table 4.5 shows the result of regularised FCRM with ℓ_1 penalty fitted to normalized dataset **SYNT-**3. In terms of model sparsity, not much benefit is observed which is consistent with the fact that there is no poor/correlated feature. Tables 4.7, 4.9 and 4.10 show results for regularised FCRM with *non-negative garrotte* on **SYNT-3** using Algorithm 3, Algorithm 4 and Algorithm 5, respectively. These results suggest that the two algorithms produce similar result except at higher values of the penalty weight. At higher values of the penalty weight λ, certain predictor coefficients are brought down to zero by Algorithm 3. Note that, Algorithm 3 cycles between feature weight estimation and estimation of $\boldsymbol{\beta_i}$s. This enables it to realize sparsity by drop-

ping less useful features. It should be noted that this sparsity is achieved at the cost of higher root mean square error.

Table 4.3: regularised FCRM with ℓ_1 penalty on dataset **SYNT-4**. Model parameter is abbreviated as M.P, objective value is abbreviated as O.V [2]

β	M.P ($\lambda = 0$)				M.P ($\lambda = 0.5$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
				(crisp)				(crisp)
β_0	0.48	1.21	0.045	0.798	0.47	1.19	0.046	0.837
β_1	0.11	0.23		(0.4)	0.11	0.22		(0.42)
β_2	0.0	-0.16			0.0	-0.16		
β_3	0.05	-0.01			0.04	0.0		
β_4	-0.01	-0.01			0.0	0.0		

β	M.P ($\lambda = 0.01$)				M.P ($\lambda = 1.0$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
				(crisp)				(crisp)
β_0	0.48	1.2	0.045	0.799	0.46	1.18	0.048	0.95
β_1	0.11	0.23		(0.4)	0.1	0.22		(0.47)
β_2	0.0	-0.16			0.0	-0.15		
β_3	0.05	-0.01			0.04	0.0		
β_4	0.0	0.0			0.00	0.0		

The single feature weight case of *non-negative garrotte* is not significantly different from *multiple feature weight garrotte* (where separate feature vector is used for each switching regression model) for the synthetic datasets we tried.

The Lagrangian multiplier λ controls the extent to which parameters are shrunk. Higher values of λ encourages more sparse models and this sparseness may be induced at the cost of increased RMS error.

Sometimes our algorithm terminated at extrema different from those obtained using the true initialization. Apparently the minimum of the FCRM function most of the time occurred for parameter values near the true values, but the algorithm was sometimes trapped at a bad local solution.

Since, the FSRM is not convex w.r.t the partition matrix U, depending on the initial condition, the final partition may be close to the best solution or could also be a very poor solution. In order to verify that the feature selection by regularized FCRM is useful, we performed the following exercise. A Fuzzy partition of the data was first obtained using unregularized FCRM algorithm using all features. Next, a second fuzzy clustering was obtained using the subset of features selected by regularized FCRM. We compare the similarity of these two partitions. A high value of this similarity measure suggests a reasonably good feature selection. We use Adjusted rand index (ARI) as a measure of similarity between two data clusterings. The ARI values are then analysed to comment on the usefulness of feature selection by the two methods. Table 4.17 reports ARI results for different subsets of features selected by regularized FCRM on **SYNT-3**.

4.2 Results on real world data: used cars data[8]

Next we discuss effectiveness of the proposed methods on the used cars data. The used cars dataset is normalised beforehand using z-score normalization. Tables 4.11 and 4.12 respectively show results when the used cars dataset is fitted to unregularized FCRM and regularised FCRM with ℓ_1 penalty. Tables 4.13, 4.14 and 4.15 respectively show results of regularized FCRM with *non-negative garrotte* using Algorithm 3, Algorithm 4 and *single feature weight non-negative garrotte* when fitted to usedcars data. The model coefficients are estimated using the following parameters: $c = 2, m = 2$, termination condition(ϵ) = 0.0000005.

The *single feature weight garrrotte* regularizer selects a subset of features which are common to both the models assumed for the used cars dataset. As described earlier, a comparison between the partitions obtained by FCRM using (1) all features and (2) selected features by regularized FSRM is necessary to test the quality of feature selection. In both cases the algorithms are run with the same initial partition. Table 4.16 shows this comparison using ARI which is a the measure of similarity between two partitions on the same data. Note that the ARI value decreases for each run as the number of selected features decreases. A total of 30 runs are made and ARI values are calculated. Since the FCRM cost function is a non convex one, algorithm terminated at extrema other than the desirable one in several cases. Table 4.16 reports the maximum ARI value, i.e., maximum similarity between the clustering among all the 30 runs. The average ARI is shown within parenthesis.

Tables 4.11 and 4.12 suggest that although ℓ_1 regularizer does some shrinkage of the coefficients, with $\lambda = 1.0$ it eliminates one feature from one model. While with $\lambda = 10.0$ two different features, one each from the two models and a common feature is eliminated. Tables 4.13 and 4.14 on the other hand, shows that garrotte is much more effective than lasso in inducing sparsity, maintaining the same level of root mean square error (RMSE). Comparing Tables 4.13 and 4.14, again we find that Algorithm 3 is more effective in inducing sparsity than Algorithm 4.

Table 4.15 reports the results on used cars data by *non-negative garrotte* (Algorithm 3) when a single weight vector is used for both models. For this dataset, we find that it is equally effective as the case with different weight vectors for different models, but at the cost of a significantly higher RMS error .

Table 4.4: regularised FCRM with *non-negative garrotte* on dataset **SYNT-4** using Algorithm 3

β	M.P ($\lambda = 0$)				M.P ($\lambda = 0.5$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
				(crisp)				(crisp)
β_0	0.48	1.21	0.045	0.798	0.47	1.19	0.06	1.44
β_1	0.11	0.23		(0.4)	0.08	0.21		(0.718)
β_2	0.0	-0.16			0.0	-0.14		
β_3	0.05	-0.01			0.0	0.0		
β_4	-0.01	-0.01			0.0	0.0		
β	M.P ($\lambda = 0.01$)				M.P ($\lambda = 1.0$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
				(crisp)				(crisp)
β_0	0.48	1.19	0.045	0.808	0.48	1.22	0.12	5.81
β_1	0.11	0.23		(0.4)	0.08	0.14		(2.9)
β_2	0.0	-0.16			0.0	0.0		
β_3	0.05	0.0			0.0	0.0		
β_4	0.0	0.0			0.0	0.0		

Table 4.5: Regularised FCRM with ℓ_1 penalty fitted to normalized dataset **SYNT**-3

β	M.P ($\lambda = 0$)				M.P ($\lambda = 0.5$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
β_0	0.49	1.23	0.02	0.13	0.49	1.23	0.02	0.13
β_1	0.11	0.22		(0.06)	0.11	0.21		(0.07)
β_2	0.00	-0.16			0.00	-0.15		
β_3	0.04	0.00			0.04	0.00		
β	M.P ($\lambda = 0.01$)				M.P ($\lambda = 1.0$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
β_0	1.23	0.49	0.02	0.13	0.49	1.23	0.02	0.16
β_1	0.22	0.11		(0.06)	0.10	0.21		(0.08)
β_2	-0.16	0.00			0.00	-0.15		
β_3	0.00	0.04			0.04	0.00		

Table 4.6: Regularised FCRM with combined ℓ_1, ℓ_2 penalty fitted to normalized dataset **SYNT−3** using algorithm−4

β	M.P ($\lambda = 0$, $\gamma = 0$)				M.P ($\lambda = 0.5$, $\gamma = 0.5$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
β_0	1.23	0.49	0.02	0.13	0.49	1.23	0.02	0.16
β_1	0.22	0.11		(0.06)	0.11	0.21		(0.08)
β_2	-0.16	0.00			0.00	-0.15		
β_3	0.00	0.04			0.04	0.00		
β	M.P ($\lambda = 0.01$, $\gamma = 0.01$)				M.P ($\lambda = 1.0$, $\gamma = 1.0$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
β_0	0.49	1.23	0.02	0.13	0.48	1.22	0.024	0.24
β_1	0.11	0.22		(0.06)	0.1	0.21		(0.12)
β_2	0.00	-0.16			0.00	-0.15		
β_3	0.04	0.00			0.04	0.00		

Table 4.7: Regularised FCRM with *non-negative garrotte* applied to normalized dataset **SYNT**-3 using algorithm-3. Feature weight is abbreviated as F.wt

β	M.P ($\lambda = 0$)				M.P ($\lambda = 0.5$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
					(F.wt)	(F.wt)		
β_0	0.49	1.23	0.02	0.13	1.23 (0.97)	0.49 (0.99)	0.05	0.83
β_1	0.11	0.22		(0.06)	0.21 (0.93)	0.07 (1.02)		(0.42)
β_2	0.00	-0.16			-0.14 (0.76)	0.00 (0.0)		
β_3	0.04	0.00			0.00 (0.0)	0.00 (0.0)		
β	M.P ($\lambda = 0.01$)				M.P ($\lambda = 1.0$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
	(F.wt)	(F.wt)			(F.wt)	(F.wt)		
β_0	1.23 (0.97)	0.49 (1.0)	0.02	0.13	1.23 (0.97)	0.48 (0.97)	0.07	1.95
β_1	0.22 (0.99)	0.11 (1.55)		(0.06)	0.19 (0.87)	0.03 (0.42)		(0.98)
β_2	-0.16 (0.82)	0.00 (0.0)			-0.13 (0.70)	0.0 (0.00)		
β_3	0.0 (0.0)	0.04 (2.06)			0.0 (0.00)	0.0 (0.00)		

Table 4.8: Average Predictor coefficients of RFCRM with *non-negative garrotte* applied to normalized dataset **SYNT**-3 ($\lambda = 1.0$) and used cars dataset using algorithm-3 ($\lambda = 1.0$) over 25 runs. The values within () are standard deviation over 25 runs.

	SYNT-3 dataset			**used cars** dataset	
	model-1	model-2	β	model-1	model-2
$\hat{\beta}_0$	1.236	0.487	$\hat{\beta}_0$	-0.13	-0.11
	(0.008)	(0.003)		(0.0)	(0.0)
$\hat{\beta}_1$	0.199	0.075	$\hat{\beta}_1$	0.95	0.0
	(0.016)	(0.01)		(0.0)	(0.0)
$\hat{\beta}_2$	-0.13	0.0	$\hat{\beta}_2$	0.0	-0.56
	(0.035)	(0.0)		(0.0)	(0.0)
$\hat{\beta}_3$	0.0	0.0	$\hat{\beta}_3$	0.0	0.24
	(0.0)	(0.0)		(0.0)	(0.0)
			$\hat{\beta}_4$	0.0	0.0
				(0.0)	(0.0)
			$\hat{\beta}_5$	0.0	0.0
				(0.0)	(0.0)

Table 4.9: Regularised FCRM with *non-negative garrotte* applied to normalized dataset **SYNT−3** using algorithm−4

β	M.P ($\lambda = 0$)				M.P ($\lambda = 0.5$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
	(F.wt)	(F.wt)			(F.wt)	(F.wt)		
β_0	1.23	0.49	0.02	0.13	0.49 (0.99)	1.23 (1.00)	0.04	0.64
β_1	0.22	0.11		(0.06)	0.09 (0.83)	0.21 (0.94)		(0.32)
β_2	-0.16	0.00			0.00 (0.00)	-0.14 (0.92)		
β_3	0.00	0.04			0.00 (0.00)	0.00 (0.00)		
β	M.P ($\lambda = 0.01$)				M.P ($\lambda = 1.0$)			
	model1	model2	rmse	O.V	model1	model2	rmse	O.V
	(F.wt)	(F.wt)			(F.wt)	(F.wt)		
β_0	0.49 (1.00)	1.23 (1.00)	0.02	0.13	0.48 (0.97)	1.23 (1.00)	0.05	1.11
β_1	0.11 (1.00)	0.22 (1.00)		(0.06)	0.07 (0.63)	0.19 (0.88)		(0.56)
β_2	0.00 (0.00)	-0.16 (1.00)			0.00 (0.00)	-0.13 (0.83)		
β_3	0.04 (0.98)	0.00 (0.00)			0.00 (0.00)	0.00 (0.00)		

Table 4.10: Regularised FCRM with *non-negative garrotte (single feature weight)* fitted to **SYNT**−3 using algorithm−5

β	M.P ($\lambda = 0$)				M.P ($\lambda = 0.5$)			
	model1	model2	rmse	O.V	model1 (F.wt)	model2	rmse	O.V
β_0	1.234	0.492	0.018	0.126	1.234 (1.00)	0.492	0.037	0.560
β_1	0.218	0.109		(0.063)	0.208 (0.96)	0.105		(0.280)
β_2	-0.155	0.0			-0.142 (0.91)	0.0		
β_3	0.002	0.044			0.0 (0.00)	0.0		
β	M.P ($\lambda = 0.01$)				M.P ($\lambda = 1.0$)			
	model1 (F.wt)	model2	rmse	O.V	model1 (F.wt)	model2	rmse	O.V
β_0	1.234 (1.00)	0.492	0.018	0.126	1.235 (1.00)	0.493	0.045	0.794
β_1	0.218 (1.00)	0.109		(0.063)	0.197 (0.91)	0.099		(0.397)
β_2	-0.155 (1.00)	0.0			-0.126 (0.81)	0.0		
β_3	0.002 (0.97)	0.043			0.0 (0.00)	0.0		

Table 4.11: FCRM on usedcars data

	model parameters				
β	model-1	model-2	objective value(crisp)	rmse	no. of iterations
β_0	-0.15	-0.125	230.71	0.379	64
β_1	1.14	-0.54	(115.35)		
β_2	-0.18	-0.02			
β_3	0.0	0.235			
β_4	0.07	-0.01			
β_5	0.08	0.09			

Table 4.12: regularised FCRM with lasso on the usedcars data. Absolute value sum is abbreviated as A.V.S.

	M.P($\lambda = 0.5$)			A.V.S		
β	model-1	model-2	rmse	model-1	model-2	O.V
β_0	-0.123	-0.151	0.379	1.0	1.55	230.8
β_1	-0.558	1.112				(115.4)
β_2	-0.001	-0.144				
β_3	0.228	0.0				
β_4	-0.007	0.064				
β_5	0.089	0.077				
	M.P($\lambda = 1.0$)			A.V.S		
β	model-1	model-2	rmse	model-1	model-2	O.V
β_0	-0.149	-0.121	0.379	1.475	0.99	231.07
β_1	1.080	-0.559				(115.54)
β_2	-0.114	0.0				
β_3	0.0	0.223				
β_4	0.059	-0.007				
β_5	0.073	0.084				
	M.P($\lambda = 10.0$)			A.V.S		
β	model-1	model-2	rmse	model-1	model-2	O.V
β_0	-0.076	-0.121	0.386	0.776	1.1	240.34
β_1	-0.525	0.938				(120.17)
β_2	0.0	0.0				
β_3	0.148	0.0				
β_4	0.0	0.011				
β_5	0.027	0.034				

Table 4.13: regularised FCRM with garrotte on the usedcars data using algorithm-3. F.Wt is abbreviation for Feature weight

	M.P($\lambda = 0.01$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.149 (1.886)	-0.125 (0.722)	0.383	1.240	1.095	236.477
β_1	0.965 (2.871)	0.0 (0.0)				(118.239)
β_2	0.0 (0.0)	-0.551 (4.794)				
β_3	0.0 (0.0)	0.309 (1.113)				
β_4	0.039 (19.676)	0.0 (0.0)				
β_5	0.087 (7.277)	0.11 (27.389)				

	M.P($\lambda = 0.1$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.148 (1.873)	-0.119 (0.688)	0.386	1.162	1.011	239.44
β_1	0.949 (2.824)	0.0 (0.0)				(119.72)
β_2	0.0 (0.0)	-0.585 (5.091)				
β_3	0.0 (0.0)	0.264 (0.949)				
β_4	0.0 (0.0)	0.0 (0.0)				
β_5	0.065 (5.383)	0.043 (10.693)				

	M.P($\lambda = 1.0$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.131 (1.663)	-0.105 (0.609)	0.391	1.084	0.905	245.749
β_1	0.952 (2.834)	0.0 (0.0)				(122.874)
β_2	0.0 (0.0)	-0.563 (4.897)				
β_3	0.0 (0.0)	0.236 (0.851)				
β_4	0.0 (0.0)	0.0 (0.0)				
β_5	0.0 (0.0)	0.0 (0.0)				

Table 4.14: regularised FCRM with garrotte on the usedcars data using algorithm-4

	M.P($\lambda = 0.01$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.125 (1.003)	-0.153 (0.995)	0.379	1.009	1.603	230.721
β_1	-0.545 (1.065)	1.145 (0.993)				(115.361)
β_2	-0.014 (0.0)	-0.175 (0.951)				
β_3	0.236 (0.991)	0.0 (0.0)				
β_4	-0.007 (0.0)	0.068 (0.968)				
β_5	0.094 (0.978)	0.082 (0.979)				

	M.P($\lambda = 0.1$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.124 (1.003)	-0.152 (0.995)	0.379	1.021	1.622	230.763
β_1	-0.559 (1.065)	1.137 (0.993)				(115.382)
β_2	0.0 (0.0)	-0.167 (0.951)				
β_3	0.235 (0.991)	0.0 (0.0)				
β_4	0.0 (0.0)	0.066 (0.968)				
β_5	0.091 (0.978)	0.080 (0.979)				

	M.P($\lambda = 1.0$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.145 (1.663)	-0.112 (0.609)	0.38	1.405	0.965	231.720
β_1	1.059 (2.834)	-0.556 (0.0)				(115.86)
β_2	-0.091 (0.0)	0.0 (4.897)				
β_3	0.0 (0.0)	0.22 (0.851)				
β_4	0.045 (0.0)	0.0 (0.0)				
β_5	0.065 (0.0)	0.076 (0.0)				

Table 4.15: regularised FCRM with single feature weight vector garrotte on the usedcars data using algorithm-3

	M.P($\lambda = 0.01$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.08 (1.011)	-0.175	0.424	1.129	1.710	288.822
β_1	0.23 (0.686)	0.335				(144.411)
β_2	0.707 (6.432)	-0.740				
β_3	-0.02 (1.538)	0.428				
β_4	0.001 (0.256)	0.003				
β_5	0.091 (7.583)	0.03				
	M.P($\lambda = 0.1$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.079 (1.0)	-0.173	0.424	1.105	1.681	289.041
β_1	0.227 (0.677)	0.33				(144.521)
β_2	0.711 (6.465)	-0.744				
β_3	-0.019 (1.481)	0.412				
β_4	0.0 (0.0)	0.0				
β_5	0.068 (5.654)	0.023				
	M.P($\lambda = 1.0$)			A.V.S		
β_i	model-1(F.Wt)	model-2(F.Wt)	rmse	model-1	model-2	O.V
β_0	-0.074 (1.663)	-0.162	0.426	1.027	1.597	292.098
β_1	0.22 (2.834)	0.319				(146.049)
β_2	0.716 (0.0)	-0.748				
β_3	-0.017 (0.0)	0.367				
β_4	0.0 (0.0)	0.0				
β_5	0.0 (0.0)	0.0				

Table 4.16: ARI values for the used cars data when different sets of features are selected

feature number (as in Table 6)	{2,7,8,10,12}	{2,7,8,10}	{7,8,10}
ARI	0.99 (0.925)	0.907 (0.51)	0.46 (0.42)

Table 4.17: ARI values for synthetic data **SYNT-**3 when different sets of features are selected

selected features	$\{x_1, x_2\}$	$\{x_1, x_2, x_3\}$
similarity	0.446 (0.446)	0.92 (0.92)

Chapter 5

Discussion

In this work, we attempt to simplify switching regression models by applying different regularizers to the ordinary FCRM objective function. After solving the regularized FSRM, we expect that the magnitude of predictor coefficients are reduced. Effect of the following three regularizers on model sparsity are analysed $-(1)$ ℓ_1 penalty, (2) ℓ_1 and ℓ_2 penalty, and (2) *non-negative garrotte*. In all the cases, suitable optimization algorithms are proposed. Regularized FSRM with *non-negative garrotte* is found to outperform regularized FSRM with ℓ_1 penalty in terms of feature selection. For *non-negative garrote* with multiple feature weight vectors, significant model sparsity is attained. But the subset of selected features may not be common across all the models of switching regression. Among the two variations of *non-negative garrote* - single feature weight vector and l feature weight vectors, the former selects a commom subset of feature. We have demonstrated the effectiveness of the proposed models on both synthetic and real datasets. Table 4.8 shows the case when FSRM with non-negative garrotte is fitted to the used cars data and the **SYNT-3** dataset. It is evident from the table that the variance of the model fitted to the used car data is lower than the synthetic data. *Non-negative garrotte* has two versions: (1) in the first version (Algorithm 3), the partition matrix U is estimated by holding the feature weights constant, next the feature weights are estimated holding U constant and this steps are repeated until convergence, (2) in the second version (Algorithm 4), both the partition matrix U and feature weights are estimated only once. Of the two, Algorithm 3 is found to yield better sparsity.

Bibliography

[1] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

[2] David W David Jr. Maximum likelihood estimates of the parameters of a mixture of two regression lines. *Communications in Statistics-Theory and Methods*, 3(10):995–1006, 1974.

[3] Neil E Day. Estimating the components of a mixture of normal distributions. *Biometrika*, 56(3):463–474, 1969.

[4] Richard D De Veaux. Mixtures of linear regressions. *Computational Statistics & Data Analysis*, 8(3):227–245, 1989.

[5] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[6] Richard J Hathaway and James C Bezdek. Switching regression models and fuzzy clustering. *IEEE Transactions on fuzzy systems*, 1(3):195–204, 1993.

[7] Richard Joseph Hathaway. *Constrained maximum-likelihood estimation for a mixture of m univariate normal distributions*. PhD thesis, Rice University, 1983.

[8] Shonda Kuiper. Introduction to multiple regression: How much is your car worth? *Journal of Statistics Education*, 16(3), 2008.

[9] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[10] Robert D Nowak, Stephen J Wright, et al. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of selected topics in signal processing*, 1(4):586–597, 2007.

[11] William L Silber. The market for federal agency securities: Is there an optimum size of issue? *The Review of Economics and Statistics*, pages 14–22, 1974.

[12] Hye Won Suk and Heungsun Hwang. Regularized fuzzy clusterwise ridge regression. *Advances in data analysis and classification*, 4(1):35–51, 2010.

[13] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[14] Jacques Van der Gaag and Wim Vijverberg. A switching regression model for wage determinants in the public and private sectors of a de-

veloping country. *The review of economics and statistics*, pages 244–252, 1988.

[15] Qi Zhu, Daoqiang Zhang, Han Sun, and Zhengming Li. Combining l1-norm and l2-norm based sparse representations for face recognition. *Optik-International Journal for Light and Electron Optics*, 126(7):719–724, 2015.

[16] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.