ANALYSIS OF BAYESIAN GROUP-LASSO IN REGRESSION MODELS

By

MANU CHANDRAN

A THESIS PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2011
To Mom and Dad
ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Pramod Khargonekar, and my committee members, Dr. Paul Gader and Dr. Tanmay Lele, for the guidance and support throughout this study and research. I would also like to thank Dr. John Shea, for inspiring me by his enthusiasm in teaching and research. I would also like to thank Dr. Brandi Ormerod, for giving an opportunity to work with her team. Additionally, I would like to thank my roommates Manu Nandan, Clint George and Abbin Antony, for their support throughout my stay at the University of Florida. I would also like to thank my parents, Mohana Chandran and Padma Kumari, and my close friends, Aswin Sreenivas, Jaya Lekshmi and Biji Rose, for their constant support and encouragement.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>4</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>6</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>7</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>8</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>9</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>9</td>
</tr>
<tr>
<td>1.2 Organization</td>
<td>11</td>
</tr>
<tr>
<td>2 BACKGROUND</td>
<td>12</td>
</tr>
<tr>
<td>2.1 Using a Sparsity Regularization Term</td>
<td>13</td>
</tr>
<tr>
<td>2.1.1 Lasso</td>
<td>13</td>
</tr>
<tr>
<td>2.1.2 Group-Lasso</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Bayesian Learning with Independent Priors</td>
<td>16</td>
</tr>
<tr>
<td>3 ALGORITHM AND DISCUSSION</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Bayesian Group-Lasso Model</td>
<td>19</td>
</tr>
<tr>
<td>3.1.1 Equivalence to Group-Lasso</td>
<td>20</td>
</tr>
<tr>
<td>3.1.2 Hierarchical Model</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Bayesian Group-Lasso Implementation</td>
<td>21</td>
</tr>
<tr>
<td>3.2.1 Sampling</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2 Convergence</td>
<td>22</td>
</tr>
<tr>
<td>3.2.3 Choosing Model Hyperparameters</td>
<td>23</td>
</tr>
<tr>
<td>3.3 Concept of Group Structure</td>
<td>23</td>
</tr>
<tr>
<td>3.3.1 Strong Group Sparsity</td>
<td>24</td>
</tr>
<tr>
<td>3.3.2 Weak Group Sparsity</td>
<td>24</td>
</tr>
<tr>
<td>4 RESULTS</td>
<td>26</td>
</tr>
<tr>
<td>5 CONCLUSION AND FUTURE WORK</td>
<td>32</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>33</td>
</tr>
<tr>
<td>BIOGRAPHICAL SKETCH</td>
<td>35</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>4-1</td>
<td>Data set 1 results</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Lasso and ridge regression estimation</td>
<td>14</td>
</tr>
<tr>
<td>3-1</td>
<td>Hierarchical structure of Bayesian Group-Lasso regression model</td>
<td>21</td>
</tr>
<tr>
<td>3-2</td>
<td>Example of regression coefficient structure for the case of strong group sparsity</td>
<td>24</td>
</tr>
<tr>
<td>3-3</td>
<td>Example of regression coefficient structure for the case of weak group sparsity</td>
<td>25</td>
</tr>
<tr>
<td>4-1</td>
<td>Changing number of samples for a strong group sparsity case</td>
<td>28</td>
</tr>
<tr>
<td>4-2</td>
<td>Changing number of groups for a strong group sparsity case</td>
<td>29</td>
</tr>
<tr>
<td>4-3</td>
<td>Changing number of samples for a weak group sparsity case</td>
<td>30</td>
</tr>
</tbody>
</table>
Abstract of Thesis Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Master of Science

ANALYSIS OF BAYESIAN GROUP-LASSO IN REGRESSION MODELS

By
Manu Chandran
August 2011

Chair: Pramod Khargonekar
Major: Electrical and Computer Engineering

The Group-Lasso estimator, used in regression analysis, does not calculate the variance estimates of regression coefficients. Such estimates are important, since they represent the confidence of the model in its estimation. By using the Bayesian version of Group-Lasso, known as Bayesian Group-Lasso, we can estimate the variance estimates of the regression coefficients. Bayesian Group-Lasso has already been proposed and used for classification models [15]. In this thesis, we use the Bayesian Group-Lasso model for regression problems. It is implemented using a Gibbs sampling technique. We evaluate its performance and compare it with Lasso and Group-Lasso techniques, using datasets generated from known model parameters.
CHAPTER 1
INTRODUCTION

1.1 Motivation

In many fields such as genomics, computational biology and text categorization, the data available for processing has been growing enormously. It is becoming increasingly difficult to make sense of this large amount of data. Many techniques have been developed in the past few decades to address this challenge. Sparse data representations offer a very promising avenue to deal with such problems. A sparse representation is a linear combination of a small number of elementary signals, which retains all or most of the information contained in the original signal. The transformation of data into a sparse representation also helps in storing the data more efficiently. The process of storing data in an efficient manner, with or without loss of information is known as data compression.

A relatively new technique in data compression is compressive sensing. If a signal is known to be sparse in some domain, then compressive sensing needs fewer samples for signal reconstruction than needed by the conventional Nyquist-Shannon sampling [2]. It uses a nonlinear process for signal reconstruction. In contrast, when conventional sampling techniques are used, linear interpolation is used for signal reconstruction. Compressive sensing is now generally considered a revolutionary technique in the field of information theory. Some other algorithms that transform data into sparse representations are discussed in depth in Section 2.

Although compression of data is a desirable objective, the main goal of many machine learning algorithms is to determine a model that best describes the available data. Supervised learning is a category of machine learning algorithms in which the input data consists of the values of observed variables known as features and the desired output variable. The desired output variable is also referred to as target variable. The input data is usually split into training data and test data. The training data is used to tune certain parameters of the model, so that the output of the model is similar to the measured values of the target variable. The test data is used to verify the performance of the estimated model. Supervised learning problems can be
of two types, classification problems and regression problems. In classification problems, the
(target variable can take only discrete values, while in regression problems the target variable
values are assumed to take continuous values. In both these methods, to infer the underlying
model, it is helpful to focus on the most prominent features. Once the most prominent
features are identified, we need to store information of only those features. Thus we can
transform data into its sparse representation. An example of supervised learning problem that
requires sparse representation of data is micro-arrays, from the field of computational biology.
Recently, micro-arrays are increasingly used for diagnosis of a variety of diseases. As the use
of micro-arrays has increased, the use of feature selection algorithms in micro-arrays has
also increased [7]. A very important technique used in feature selection is the Least absolute
shrinkage and selection operator (Lasso). We will discuss Lasso and its properties in Section
2.1.1.

In many computational biology applications, grouping of features can be helpful in
determining the model of the system. Grouping of features and selection of a group of features
is known as group selection. This is used when a group structure of the features is already
available. For example, if we know that two features belong to the same group, then it
might be appropriate to include or exclude them simultaneously. Grouping of features also
finds application in fusion of data. Performance of a number of algorithms, such as mass
spectrometry for cancer detection and hyper spectral imagery, can be improved by grouping the
features [16]. An extension of Lasso, known as Group-Lasso technique has been proposed to
improve the performance of these algorithms [21].

Although, Group-Lasso can be used in problems requiring grouping of features, it gives a
point estimate of the solution. The variances of estimated solutions, represent the confidence
of the model in the estimation, which is a desirable property. Bayesian Group-Lasso is an
extension of the Group-Lasso algorithm, which finds the solution and its variance estimate. We
will discuss Bayesian Group-Lasso in detail in Section 3.1.
In this thesis, we analyze the performance of the algorithm Bayesian Group-Lasso, as applied to regression problems, using simulated datasets. Bayesian Group-Lasso was used in [15] to address learning problems in classification. An in-depth performance analysis of Bayesian Group-Lasso in regression problems has never been done. A previous attempt in [10] did not bring out the effect of change in sample size and group structure on the performance of Bayesian Group Lasso. In this work, performance of Bayesian Group-Lasso is compared with Lasso and Group-Lasso for datasets of varied sample size and group structure.

1.2 Organization

This thesis is organized as below. In Chapter 2, we explain the most important algorithms used in feature selection. The concept of feature selection using regularization parameter and usage of priors for regularization are presented. In Chapter 3, the Bayesian Group-Lasso model is presented. The analysis, implementation, advantages and disadvantages of the Bayesian Group-Lasso are discussed. In Chapter 4, the results of this study are presented. The study mainly includes analysis of simulated data sets to bring out the properties of Bayesian Group-Lasso. Finally, Chapter 5 summarizes this study and presents a discussion on possible extensions of the current work.
As mentioned in the introduction, many upcoming fields find use of feature selection algorithms. This has fueled an increasing interest in the development of feature selection algorithms. In some cases, identification of the prominent features is important. In others, identification of a group of prominent features is important. In this chapter, we discuss the most prominent feature selection algorithms found in the literature. The algorithms that we will discuss in this chapter are for regression problems. These algorithms, perform feature selection and regression simultaneously. Feature selection algorithms can be divided into two categories. 1) Techniques that use a sparsity regularization term, for example, Lasso and Group-Lasso and 2) Techniques that use independent priors, for example Bayesian Lasso, Joint Classifier and Feature Optimization (JCFO) and Relevance vector machine (RVM). In this chapter, we will review some of these techniques in detail, bringing out their advantages and disadvantages.

Consider a dataset of \( N \) data points. Each data point consists of a pair \( (x_i, y_i) \) where \( x_i \) is the input variable and \( y_i \) is the target variable. Each input \( x_i \) is a vector in \( \mathbb{R}^D \), where \( D \) denotes the number of features. We define \( X \) to be the \( N \times D \) matrix whose rows consist of input data vectors \( x_i, i = 1, 2, ..., N \) and \( y \) to be the vector \( (y_1, ..., y_N)^T \), \( y_i \in \mathbb{R} \).

The models described in this thesis assume a linear relationship between \( y \) and \( X \). Let \( w \) denote the regression coefficient vector, a \( D \) dimensional vector defined by \( w = (w_1w_2...w_D)^T \). The model for linear regression is given by the Equation (2–1), where \( \epsilon \) denote the error in the linear relationship between \( y \) and \( X \):

\[
y = Xw^T + \epsilon
\]  

(2–1)

The objective of the algorithms described in this section is to find an estimate of the regression coefficient vector \( w \). We denote the estimated regression coefficient vector as \( w^* \). The target variable calculated from this estimated regression coefficient vector is given by \( y^* = Xw^{*T} \). One of the objectives of the feature selection algorithms is to minimize the difference between
\(y\) and \(y^*\). The other objective is to identify the prominent features. We can define a function \(f(w)\), which gives the error between \(y\) and \(y^*\). A commonly used error function is the squared \(l_2\)-norm of the difference between \(y\) and \(y^*\), expressed as \(f(w) = \|y - y^*\|_2^2\).

### 2.1 Using a Sparsity Regularization Term

Let us say that the aim is to find a solution for \(w^*\) that has a small number of nonzero values i.e., a sparse solution [5]. This can be found by minimizing the error function \(f(w)\) with the constraint that \(w\) is sparse. An indirect approach to this problem is via the following optimization problem.

\[
w^* = \arg \min_w [f(w) + \lambda g(w)]
\]

Here \(g(w)\) is a sparsity promoting function that penalizes non-sparse solutions. Here \(\lambda \geq 0\) is a regularization parameter which balances the trade off between original objective function \(f(w)\) and sparsity promoting term \(g(w)\). The value of \(\lambda\) that gives the solution with the minimum error, is determined using techniques such as cross validation [7]. The regularization term \(g(w)\) also helps in avoiding overfitting. Overfitting refers to the behavior of learning techniques in which the learned model becomes increasingly tuned to the random noise in the target variable [3]. In this thesis, we focus on techniques that generate sparse solutions. A very important regularization function is given by \(g(w) = \|w\|_0\), where \(\|\cdot\|_0\) denotes \(l_0\)-norm which is defined to be the number of nonzero elements in \(w\).

#### 2.1.1 Lasso

Although the idea of using \(l_0\)-norm of \(w\), to solve (2–2) is appealing, optimizing (2–2) with \(g(w) = \|w\|_0\) is a hard problem [11]. Hence, \(g(w)\) is usually chosen to be some relaxed form of the \(l_0\)-norm. The choice is usually some \(l_p\)-norm where \(p \leq 2\). A widely applied technique is to use the \(l_2\)-norm as the regularization function \(g(w)\), which is known as Ridge regression. The Ridge regression estimate can be mathematically expressed as in Equation (2–3).

\[
\min_{w,\lambda} \left[ (y - Xw^T)^T (y - Xw^T) + \lambda \sum_{j=1}^{D} |w_j|^2 \right]
\]
Figure 2-1. Lasso and ridge regression estimation: The Error function \( f(\mathbf{w}) \) and regularization function \( g(\mathbf{w}) \) of Lasso and Ridge regression estimation is shown in these figures. Function \( f(\mathbf{w}) \) is represented by the elliptical contours. A) Lasso: The contour of regularization function \( g(\mathbf{w}) = \| \mathbf{w} \|_1 \), for Lasso is represented by the square contours. B) Ridge regression: The regularization function \( g(\mathbf{w}) = \| \mathbf{w} \|_2 \), for Ridge regression, is represented by the circular contours which has no edges.

Although the regularization function \( g(\mathbf{w}) \) in Ridge regression helps in avoiding overfitting, it does not encourage sparsity. To encourage sparsity a popular technique is to use \( l_1 \)-norm as the regularization function \( g(\mathbf{w}) \). This is known as Least absolute shrinkage and selection operator (Lasso). Lasso was suggested as a selection method for linear regression by Tibshirani [18]. It is widely regarded as a benchmark technique for sparse regression. The Lasso estimate can be mathematically expressed as in Equation (2-4), where \( \lambda \geq 0 \) is a Lagrange multiplier:

\[
\min_{\mathbf{w}, \lambda} \left[ (\mathbf{y} - \mathbf{X}\mathbf{w}^T)^T (\mathbf{y} - \mathbf{X}\mathbf{w}^T) + \lambda \sum_{j=1}^{D} |w_j| \right] \tag{2-4}
\]

We illustrate Lasso in Figure 2-1A. As the value of \( \lambda \) increases, the size of the square also increases. The outer most points of the Lasso regularization function lie on the axis, represented by the edges of the square. Contrast this to the case for Ridge regression shown in Figure 2-1B. Here again, as the value of \( \lambda \) increases, the size of the circle increases. The circular contour of the regularization function of Ridge regression does not have any edges, so the possibility of finding a sparse solution is less. However, in Lasso the edges of \( g(\mathbf{w}) \) are on
the axes. Hence, the chance of Lasso estimating a sparse solution on a point on the axes, is higher than in Ridge regression. These ideas have been mathematically formalized in [18].

2.1.2 Group-Lasso

In many regression problems, some of the features are known to belong to certain groups. In such cases it might be important to find sparsity at a group level than at the individual level. For example, if we know that two features belong to the same group, then it might be appropriate to include or exclude them simultaneously. Some areas where grouping of features are important are simultaneous sparse approximation [19], multi-task compressive sensing [9] and sensor fusion [17]. These applications have motivated the development of a variant of Lasso estimation, popularly known as Group-Lasso [21].

Suppose that there are D features, which are divided into G groups. Let \( P_g \) denote the number of coefficients in the group \( g \in \{1, 2, ..., G\} \). Let the matrix \( X_g \) contain the feature vectors corresponding to the \( g \)th group. The vector \( w_g \) corresponds to the regression coefficients of the \( g \)th group. The regularization function used in Group-Lasso, \( g(w) \) is designed to account for the grouping of features. It is given by \( g(w) = \sum_{g=1}^{G} \sqrt{P_g} \| w_g \|_2 \). The Group-Lasso minimizes the objective function, which is mathematically represented as

\[
\min_{w,\lambda} \left[ \left( y - \sum_{g=1}^{G} X_g w_g \right)^T \left( y - \sum_{g=1}^{G} X_g w_g \right) + \lambda \sum_{g=1}^{G} \sqrt{P_g} \| w_g \|_2 \right] \tag{2-5}
\]

From the equation of the regularization function \( g(w) \), we can notice that an \( l_2 \)-norm is used for coefficients within a group and \( l_1 \)-norm is used at the group level. It is interesting to note that, Group-Lasso does not yield sparsity within a group, while it tries to attain sparsity at the group level. Sparse selection will act only on a group level and an entire group of predictors may be dropped. To make this idea more clear let us consider the example detailed in [22].

Consider a case where coefficients have two groups, \( w = [(w_{11}, w_{12}), w_2]^T \). Here, coefficients \( w_{11} \) and \( w_{12} \) belong to the same group and another coefficient \( w_2 \) belongs to the other group. In Lasso, the penalty function corresponds to \( l_1 \)-norm, \( g(w) = |w_{11}| + |w_{12}| + |w_2| \). In Group-Lasso, the \( l_1 \)-penalty acts on the group level and \( l_2 \)-penalty within the group. Hence, the
penalty function is given by the equation \( g(w) = \| (w_{11}, w_{12}) \|_2 + |w_2| \) and coefficients in the same group, such as \( w_{11} \) and \( w_{12} \) are estimated to be sparse or non-sparse simultaneously.

### 2.2 Bayesian Learning with Independent Priors

In Section 2.1 we saw that the objective function has a regularization term to encourage sparsity and to avoid overfitting. In a Bayesian learning setting, instead of using a regularization parameter, a prior probability distribution is assigned to the coefficients of the model. By choosing suitable priors, the Bayesian learning algorithms achieve objectives such as regularization and sparsity.

**Bayesian Lasso**

As we discussed in Section 2.1.1, Lasso is a popular technique in feature selection. However, it has some limitations. One of them is that the regularization parameter \( \lambda \) has to be determined using cross validation. Another limitation is that Lasso gives a point estimate of a coefficient and thus does not give an idea of the confidence in the estimated parameters [18].

On the other hand, Bayesian Lasso does not have these limitations[14]. Extending the idea of Lasso, Park and Casella [14] proposed Bayesian Lasso, a fully Bayesian analysis using conditional Laplace prior on \( w \). They proposed a Gibbs sampling technique [3] for Bayesian Lasso estimation. In Bayesian analysis, a likelihood function and a prior function are defined for the target variable and the coefficient vector. The likelihood distribution for \( y \), is defined by (2–6).

\[
P(y|X, w) = \prod_{i=1}^{N} N(y_i|x_iw^T, \sigma^2)
\]

It is defined as the product of \( N \) Normal distributions with mean \( x_nw \) and variance \( \sigma^2 \). 

\( x_n \) refers to the \( n^{th} \) sample, that is the \( n^{th} \) row of \( X \). The choice of product of Normal distributions is attributed to the assumption that the samples are generated independently and that the noise in the system is Gaussian in nature. The prior distribution of the coefficient vector, \( w \), \( P(w) \) is assumed to be a Laplace distribution.

\[
P(w|\sigma^2) = \prod_{j=1}^{D} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|w_j|/\sqrt{\sigma^2}}
\]
The posterior distribution $P(w|y)$ is proportional to the product of likelihood distribution and prior [3]. From the posterior distribution, variance estimates of the coefficients can be found. Additionally, the model gives an estimate for the error variance. A uniform distribution over the interval $[1, A)$, where $A$ is a large number, is used as the prior distribution on $\sigma^2$. 

$$P(\sigma^2) = 1/\sigma^2$$  \hspace{1cm} (2–8)

This makes the prior non informative. Note that the prior on $w$ is conditioned on $\sigma^2$. This conditional prior has the advantage that the induced full posterior distribution of $w$ is guaranteed to be unimodal [14]. If the full posterior is not unimodal, it could slow the convergence of the sampling process and the resulting point estimates might not be meaningful [14].

**Hierarchical Model**

If a Laplace distribution prior is directly used, the complexity in computation increases. Hence, an alternate representation of Laplace distribution as a mixture of normals scaled with exponential mixing density is commonly used [1][14]. The representation of Laplace distribution as a mixture of normals is expressed by Equation (2–11):

$$P(w|\sigma^2) = \prod_{j=1}^{D} \frac{\lambda}{2\sqrt{\sigma^2}} e^{-\lambda|w_j|/\sqrt{\sigma^2}}$$  \hspace{1cm} (2–9)

$$= \prod_{j=1}^{D} \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma^2 \tau_j} e^{-\frac{\lambda|w_j|^2}{2\sigma^2 \tau_j^2}} e^{-\frac{\lambda^2}{2}\tau_j^2} d\tau_j$$  \hspace{1cm} (2–10)

$$= \prod_{j=1}^{D} \int_{0}^{\infty} N(w_j|0, \sigma^2 \tau_j^2) \Gamma(\tau_j^2|1, \frac{\lambda^2}{2}) d\tau_j$$  \hspace{1cm} (2–11)

Here $\Gamma(\tau_j^2|1, \frac{\lambda^2}{2})$ is the gamma distribution with shape parameter 1 and rate parameter as $\frac{\lambda^2}{2}$ which is equivalent to the exponential distribution $\text{Expon}(\frac{\lambda^2}{2})$. The mode of the exponential distribution $\text{Expon}(\frac{\lambda^2}{2})$, is zero, for any $\lambda$. This implies that most values of $\tau_j^2$ are zero. From (2–11), we can see that most values of $w_j$ are going to be zero, when most $\tau_j^2$ are zero.
The variance of $\tau_j^2$ is heavily dependent on the value of $\lambda$. Thus, instead of presetting the value for $\lambda$, it is desirable to estimate $\lambda$ along with other parameters. This will account for the uncertainty in the selection of $\lambda$ for regression estimates. A Gamma(a,b) prior is assigned to $\lambda$ to maintain conjugacy [14].

The Bayesian Lasso claims to perform on par with the ordinary Lasso [14]. Bayesian Lasso is analytically simpler and is also easy to implement. Bayesian Lasso also computes the variance estimates for all coefficients, which represents the confidence of the model in the estimation. Even though Bayesian Lasso has so many advantages over conventional Lasso, it should be mentioned that Bayesian Lasso is computationally more intensive and may not be suitable for large datasets.
CHAPTER 3
ALGORITHM AND DISCUSSION

As discussed in the introduction, our work is focused on the performance of Bayesian Group-Lasso in regression problems. Bayesian Group-Lasso can be considered as an extension of the Group-Lasso procedure. In this section, we discuss the Bayesian Group-Lasso model and the method we used to implement it. Additionally, the similarity of Group-Lasso and Bayesian Group-Lasso will be discussed.

3.1 Bayesian Group-Lasso Model

As explained in Section 2.1.2, the Group-Lasso is used when the group structure of the coefficient vector, $w$ is known. Group-Lasso tends to drive the coefficients within a group to zero or nonzero simultaneously. It has been shown that, if the group structure is known, Group-Lasso gives better performance than Lasso [8]. The Bayesian Group-Lasso [15] is developed from Bayesian Lasso [14] and Group-Lasso[21][15]. In Bayesian Lasso, independent and identical Laplace priors are assumed over individual regression coefficients. However, in Bayesian Group-Lasso instead of independent and identical Laplace priors over each regression coefficient, each group is assumed to have independent and identical Multi-Laplace priors, with dimension $P_g$, given by (3–2). Here $P_g$ is the size of the $g^{th}$ group, in the group structure. The vector $w_g$ is defined as the regression coefficients, that belongs to the $g^{th}$ group. The Bayesian Group-Lasso model can be represented as below:

\[
P(y|X, w, \sigma^2) = \prod_{n=1}^{N} N(y_n|x_n^Tw, \sigma^2)
\]

(3–1)

\[
P(w_g|\rho) = \text{M-Laplace}(w_g|0, (P_g\rho/\sigma^2)^\frac{1}{2})
\]

(3–2)

\[
P(\sigma^2|\nu_0, s_0^2) = \text{InvGamma}(\sigma^2|\nu_0, s_0^2)
\]

(3–3)

\[
P(\rho|r, s) = \text{Gamma}(\rho|r, s)
\]

(3–4)

The likelihood distribution for $y$ is given by (3–1). This is the same as in Bayesian Lasso as explained in Section 2.2. For estimating the noise, an inverse Gamma prior is assumed.
over the variable $\sigma^2$, given by Equation (3–3). Another parameter in the model, which has to be estimated is $\rho$. As we will see in Section 3.1.2, inverse of $\rho$, affects the variance of the regression coefficients. A Gamma distribution is assumed for the variable $\rho$, so that the posterior distribution will maintain conjugacy.

### 3.1.1 Equivalence to Group-Lasso

In the Group-Lasso, we have a regularization term given by

$$g(w) = \sum_{g=1}^{G} \sqrt{P_g} \|w_g\|_2$$

(2–5). For ease of analysis the Group-Lasso equation is restated here.

$$\min_{w, \lambda} \left[ \left( y - \sum_{g=1}^{G} X_g w_g \right)^T \left( y - \sum_{g=1}^{G} X_g w_g \right) + \lambda \sum_{g=1}^{G} \sqrt{P_g} \|w_g\|_2 \right]$$

(3–5)

If we notice the Bayesian Group-Lasso model, a multivariate $P_g$ dimensional Multi-Laplacian prior is assumed over each $g^{th}$ regression coefficient group. In the log-space of the posterior distribution, this prior will correspond to the regularization term of the Group-Lasso [3, 15].

### 3.1.2 Hierarchical Model

The use of Multi-Laplace prior for the regression coefficients $w_g$ is useful to obtain sparse solutions of $w_g$. However, this prior causes computational difficulties. So a two-level hierarchical model has been proposed to represent this prior [15] which is easier to implement and analyze. This is similar to Bayesian Lasso, where the Laplace prior on $w$ is expanded as a hierarchical model in Section 2.2. The Multi-Laplace prior can be expressed as a mixture of normal distributions with a Gamma mixing distribution, Gamma $\left( \tau^2_j | \frac{P_g+1}{2}, \frac{2}{P_g \rho} \right)$ [15]:

$$P(w_g|\sigma^2, \rho) = \text{M-Laplace}(w_g|0, (P_g\rho/\sigma^2)^{-\frac{1}{2}})$$

(3–6)

$$\propto (P_g\rho/\sigma^2)^{-\frac{P_g}{2}} \exp\left(-\left(\frac{P_g\rho}{\sigma^2}\right)^{\frac{1}{2}} \|w_g\|_2\right)$$

(3–7)

$$\propto \int_0^\infty N(w_j|0, \sigma^2 \tau_j^2) \text{Gamma} \left( \tau_j^2 | \frac{P_g+1}{2}, \frac{2}{P_g \rho} \right) d\tau_j^2$$

(3–8)

The dependency structure of the Bayesian Group-Lasso model is represented graphically in Figure 3-1 [15]. In the Figure 3-1, the covariance of $w$ is denoted by $\Sigma$, which is a DxD
Figure 3-1. Hierarchical structure of Bayesian Group-Lasso regression model: The most important part of the diagram is the hierarchical Normal-Gamma representation of regression coefficient.

matrix. The matrix $\Sigma$ is a diagonal matrix with $\tau_j^2$ as diagonal elements, with each $\tau_j^2$ repeated $P_g$ times. The parameter $\rho$ is assigned a Gamma prior, with hyperparameters $r$ and $s$. Thus, all parameters of the model are estimated from the data. However, we still have to select appropriate hyperparameters for the model. We will discuss the selection of hyperparameters in the Section 3.2.3.

### 3.2 Bayesian Group-Lasso Implementation

As discussed above, priors are assigned to many of the parameters in the model. Several integrations are required to estimate the parameters of the model. However, in practice these integrations are too complex to be done analytically. Hence, a Gibbs sampling approach is suggested for estimating the parameter distributions. For sampling parameters of a model, conditional posterior distributions of all the parameters are required. The two level hierarchical modeling, discussed in Section 3.1.2 allows derivation of conditional posterior distributions for this model [20].
3.2.1 Sampling

Gibbs sampling belongs to the family of Markov Chain Monte Carlo (MCMC) methods. In Gibbs sampling, we sample from fully conditional posterior distributions of the model parameters. Each conditional posterior distribution of the parameters is conditioned on the current values of all other parameters and observed data. In each sampling iteration, of the Bayesian Group-Lasso model, we sample the parameters $w, \sigma^2, \tau_j^2$ and $\rho$ from their conditional distributions. We stop the sampling iterations once we reach convergence. The criteria used in testing convergence for this model is described in Section 3.2.2. The samples generated from the Markov Chain after convergence is assumed to be from the joint posterior of the model. Upon convergence, a large number of samples are drawn from the joint posterior to compute the posterior mean and variance. The posterior means are used as the estimate of regression coefficients. In this work, the software WinBUGS14 [12] is used for Gibbs sampling.

3.2.2 Convergence

Convergence of the sampling techniques is important, since only the samples after convergence can be considered to be samples from the true posterior distribution [4]. If the convergence criterion is not proper, difference between the estimated posterior distribution and true distribution might be too large. This will lead to errors in the estimation of parameters. In this work, to test for convergence, three parallel instances (also known as chains) of sampling were run from dispersed starting points and were monitored for their convergence. The number of iterations will determine the convergence of the estimated posterior distribution. In this case, the convergence was decided based on overlap of density function of coefficients of the three chains, formed from samples after the burn-in period. The burn-in period is the number of iterations, required by the sampler to reach convergence and in our experiments we fixed the burn-in period as 10000 iterations. Once the burn-in period was over, 5000 samples were collected and treated as samples from the joint posterior distribution. Another parameter to choose in sampling, is the value of thinning($T_r$). Thinning is a process in which only every $T_r^{th}$ sample after the burn-in period is used and the rest of the samples discarded. Thinning the
sequence helps in reducing the autocorrelation between saved samples. In our experiments, the thinning ratio was chosen as $T_r = 5$ which resulted in an autocorrelation of 0.1 for the stored samples. The small value of autocorrelation for stored samples is another test for convergence.

3.2.3 Choosing Model Hyperparameters

Although all the parameters are estimated from the data, we have to set the values of the hyperparameters. The hyperparameters in this model are $v_0, s_0^2, r$ and $s$. Here $v_0, s_0^2$ are the shape and rate parameter of the inverse Gamma distribution on $\sigma^2$. By choosing small values of $v_0$ and $s_0^2$, the inverse Gamma distribution becomes a diffused prior, almost tending to a uniform distribution [14, 20]. This hyperparameter selection thus assumes no previous knowledge of the underlying system. However, if we have the knowledge of the noise level of the data, then we can incorporate such information in these prior parameters.

The variance associated with the weight parameter $w_j$ is $\tau^2_j$. Hence, the values of $\tau^2_j$ determine the sparsity of estimated coefficients. As in Bayesian Lasso, when most values of $\tau^2_j$ are zero, most values of $w_j$ will be estimated as zero. The Gamma distribution of $\tau^2_j$ has the rate parameter proportional to $1/\rho$. Instead of assigning a value to $\rho$, it is appealing to assign a prior on $\rho$ and estimate it along with other parameters. To maintain conjugacy, $\rho$ is assigned a Gamma distribution with shape and rate parameter $r$ and $s$. This is equivalent to sampling the rate parameter of $\tau^2_j$ from an inverse Gamma distribution. If we have some knowledge about the sparsity of the underlying model, we could tailor the hyperparameters $r$ and $s$ to include the sparsity information. For simulation purposes, values of $r$ and $s$ were chosen to be small values (for example, $r = 0.1, s = 0.1$ ), so that the rate parameter of $\tau^2_j$, is sampled from a diffused prior. So here again, we are assuming no previous knowledge of the underlying system [15, 20].

3.3 Concept of Group Structure

This section describes the concept of grouping. To study the properties of Group-Lasso, Huang and Zhang [8] introduced the concept of strong group sparsity and weak group sparsity. We will discuss these two concepts in this section. The notations used in this section will be
referred in Chapter 4, where we discuss the results. In typical feature selection scenarios, the
indicator of sparsity is the $l_0$-norm of the regression coefficient vector, $w$. We define groups of
$w$ based on our knowledge of group structure. A non-sparse group is a group of coefficients
with at least one non-zero coefficient. Let $G_{ns}$ denote the number of non-sparse groups. Let
$P_g^j$ denote the size of $j^{th}$ non-sparse group. Let $k$ denote the sum of group sizes of non-sparse
groups. Thus, $k$ is given by Equation (3–9)

$$k = \sum_{j=1}^{G_{ns}} P_g^j$$  \hspace{1cm} (3–9)

3.3.1 Strong Group Sparsity

A coefficient vector is defined to have strong group sparsity if the value for $k/\|w\|_0$ is
small. This is the case when the non-zero coefficients are effectively covered by the groups.
Consider the case when $k/\|w\|_0 = 1$. This is when all the non-sparse groups have all the
coefficients as non-zero.

Figure 3-2. Example of regression coefficient structure for the case of strong group sparsity

| 1, 1, 1, 1, 1, 1 | 0, 0, 0, 0, 0, 0 | 0, 0, 0, 0, 0, 0 |

The Figure 3-2 corresponds to the values in a coefficient vector, when it has strong
group sparsity. Here, the coefficients are assumed to have values of either 0 or 1. It gives an
example of how non-zero coefficients are grouped together in a strong group sparsity case. The
dimension (D) of this coefficient vector is 18, number of sparse features ($\|w\|_0$) is 6, number
of Groups ($G_{num}$) is 6, number of non-sparse groups ($G_{ns}$) is 2 and from Equation (3–9),
the total number of coefficients in non-sparse groups ($k$) is 6. Note here that, for this example
$k/\|w\|_0 = 1$. This is an example of strong sparsity.

3.3.2 Weak Group Sparsity

A coefficient vector is defined to have weak group sparsity if the value for $k/\|w\|_0$
is large. This implies that the group structure chosen does not correspond to the sparsity
structure of the coefficients. The non-sparse groups might have many sparse coefficients in their group. Group-Lasso considers sparsity at the group level. Thus, when \( k \) is higher compared to \( \| w \|_0 \), performance of Group-Lasso will deteriorate. We will quantitatively analyze the performance in Section 4.

\[
1, 1, 0 \quad 1, 0, 1 \quad 0, 1, 0 \quad 0, 1, 0 \quad 0, 0, 0 \quad 0, 0, 0
\]

Figure 3-3. Example of regression coefficient structure for the case of weak group sparsity

The Figure 3-3 corresponds to the values in a coefficient vector, when it has weak group sparsity. The dimension (D) of this coefficient vector is 18, number of sparse features (\( \| w \|_0 \)) is 6, number of Groups (Gnum) is 6, number of non-sparse groups (Gns) is 4 and from Equation (3–9), the total number of coefficients in non-sparse groups (k) is 12. Note here that \( k / \| w \|_0 = 2 \). This is an example of weak sparsity. Further discussion of the effect of group structure and sample size N is given in Section 4.
CHAPTER 4
RESULTS

In this chapter we will analyze the performance of Bayesian Group-Lasso. The simulation datasets are designed to test certain desirable qualities of a feature selection algorithm. Most of these tests are adopted from available literature, notably from [13] [6] and [15]

Simulation Results

In this section performance of feature selection algorithms such as Lasso, Group-Lasso and Bayesian Group-Lasso are compared using different datasets, prepared from literature. The results of Lasso and Group-Lasso were computed using 5-fold cross validation [7]. The algorithms estimate the coefficient vector using available data \( y \) and \( X \). For quantitative evaluation, we must define the error in the estimated coefficients. The recovery error is the relative difference in \( l_2 \)-norm between estimated coefficient vector \( w^* \) and true coefficient vector \( w \).

\[
\text{Err} = \frac{\|w^* - w\|_2}{\|w\|_2}
\]  

(4–1)

Strong Group Sparsity

In this experiment, we will assume that the coefficients have strong group sparsity. We defined and discussed strong group sparsity in Section 3.3.1. In this scenario, the Group-Lasso and Bayesian Group-Lasso are supposed to perform better than Lasso, by using the knowledge of group structure. In this simulation we aim to quantitatively measure the difference in performance of these methods. This experiment is a modification of the experiment described in [8]. The error performance of Lasso, Group-Lasso and Bayesian Group-Lasso are compared. The feature vectors are samples generated from a standard Gaussian distribution, \( N(0,1) \). The regression coefficient vector is randomly generated with values \( \pm 1 \). A zero-mean Gaussian noise with standard deviation \( \sigma = 0.01 \) is added to the samples. The dimension \( D = 128 \), number of non-zero coefficients \( \|w\|_0 = 32 \), and group size is selected to be even with value \( G_{\text{size}} = 8 \). So there are 16 groups and the number of non-sparse groups \( G_{ns} = 4 \). Here the number of coefficients in non-sparse groups, \( k \) is 32. The target variable \( y \) is computed using
the linear model, \( y = Xw^T + \epsilon \), where \( \epsilon \) is a Gaussian noise with variance \( \sigma^2 = 0.01 \). The number of samples \( N \) is chosen to be equal to \( 3k \) that is \( N = 96 \). The results are given in Table 4-1. The experiment was run 10 times and the mean error value and the variance are reported. As is clear from the results, the Bayesian Group-Lasso and Group-Lasso perform better than Lasso.

In the next two experiments we will see how the performance of Lasso, Group-Lasso, Bayesian Group-Lasso varies with changes in number of samples and with changes in number of groups.

**Experiment: changing number of samples**

In this experiment, we generated the coefficient vector and feature set just as in the previous experiment. As before, the dimension \( D = 128 \), number of non-zero coefficients \( \|w\|_0 = 32 \) and group size is selected to be uniform with value \( G_{\text{size}} = 8 \). Thus, there are 16 groups and the number of non-sparse groups \( G_{\text{ns}} = 4 \). Here the number of coefficients in non-sparse groups, \( k \) is 32. The number of samples is varied, \( N = 32, 96, 160 \). This corresponds to \( N/k = 1, 3, 5 \). In the Figure 4-1, x axis represents the value \( N/k \) and y axis represents the Err value, as defined in Equation 4–1. Each data point in the graphs below is calculated by running the experiment 10 times and calculating the variance and mean of the Err value.

As we can see in the Figure 4-1, when the number of samples \( N = k \), Lasso, Group-Lasso and Bayesian Group-Lasso have similar performance. This could be because the number of samples is quite low compared to the number of features and the technique used does not give any advantage. As the number of samples increase to \( N = 3k \), we can see that Group-Lasso has the lowest error. We should also note that Bayesian Group-Lasso performs better than

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean error value</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lasso</td>
<td>0.3875</td>
<td>0.113</td>
</tr>
<tr>
<td>Group-Lasso</td>
<td>0.0017</td>
<td>0.003</td>
</tr>
<tr>
<td>Bayesian Group-Lasso</td>
<td>0.1621</td>
<td>0.036</td>
</tr>
</tbody>
</table>
Figure 4-1. Changing number of samples for a strong group sparsity case

Lasso. However, in this case the error value of Bayesian Group-Lasso is in between Lasso and Group-Lasso. As the number of samples increases, the error value of Bayesian Group-Lasso and Group-Lasso converge to almost similar values. This is the case when $N = 5k$. We can see from Figure 4-1 that Group-Lasso and Bayesian Group-Lasso perform slightly better than Lasso.

**Experiment: changing number of groups**

In this experiment, we generated the coefficient vector and feature set just as in the previous experiment. As before, the dimension $D = 128$ and number of non-zero coefficients $\|w\|_0 = 32$. Here the number of coefficients in non-sparse groups is kept at $k = 32$. The number of samples is kept at $N = 96$. The group size is selected to be uniform, but varied as $G_{size} = 8, 2, 1$. Due to change in group size, the number of non-sparse groups also changes respectively, $G_{ns} = 4, 16, 32$. In the Figure 4-2, x axis represents the number of non-sparse groups $G_{ns}$ and the y axis represents the Err value, as defined in Equation 4–1
Consider the case when number of non-sparse groups, $G_{ns} = 32$. In this case, the number of non-sparse groups is the same as the number of non-sparse coefficients. The error value of Group-Lasso, Bayesian Group-Lasso and Lasso are almost equal, within statistical precision. Since $G_{size} = 1$, the problem formulations for Bayesian Group-Lasso and Group-Lasso become the same as Lasso. Hence, the performance of all three algorithms are similar. As the number of non-sparse groups decreases, Group-Lasso and Bayesian Group-Lasso exploit the knowledge of the group structure and perform better. Consider the case $G_{ns} = 4$. In this case, the error values for Bayesian Group-Lasso and Group-Lasso are far below the error value found by Lasso.

**Weak Group Sparsity**

In this experiment, we will investigate the case where the coefficients have weak group sparsity. As discussed in Section 3.3.2, weak group sparsity implies that the non-sparse groups might have many zero coefficients in their group. In this case, Lasso is expected to perform better than the Group-Lasso and Bayesian Group-Lasso.
Experiment: changing number of samples

In this experiment, we generated the feature set just as in the previous experiment. However, the coefficient vector is generated so that it has a weak sparse grouping. As before, the dimension $D = 128$ and number of non-zero coefficients $\|w\|_0 = 32$. Here the number of coefficients in non-sparse groups, $k$ is kept at 128. That is $k = 4 \|w\|_0$. Group size is selected to be uniform with value $G_{size} = 8$. Thus, there are 16 groups and the number of non-sparse groups $G_{ns} = 16$. The number of samples is varied, $N = 32, 96, 160$. This corresponds to $N/k = 1, 3, 5$. In the Figure 4-3, x axis represents the value $N/k$ and y axis represents the Err value, as defined in Equation 4–1.

As we can see in the Figure 4-3, when the number of samples $N = k$, Lasso, Group-Lasso and Bayesian Group-Lasso have similar performance. This could be because the number of samples is quite low compared to the number of features and the technique used does not give any advantage. As the number of samples increase to $N = 3k$, we can see that Group-Lasso has the highest error. This could be because of the weak group sparsity. Since

![Figure 4-3. Changing number of samples for a weak group sparsity case](image)
the groups have weak group sparsity, the number of coefficients considered to be non sparse by Group-Lasso could be higher than the actual values. It is interesting to compare the performance of Bayesian Group-Lasso and Group-Lasso in the strong group sparse situation and weak group sparse situation. Consider the case when $N/k = 2$. As seen in Figure 4-1, Bayesian Group-Lasso does not perform as well as Group-Lasso. On the same note, from Figure 4-3, for $N/k = 2$, performance of Bayesian Group-Lasso does not deteriorate like Group-Lasso.
CHAPTER 5
CONCLUSION AND FUTURE WORK

In this thesis, we studied three major methods in feature selection and analyzed their performance. Lasso, a benchmark among feature selection algorithms, is analyzed and the reason for its sparsity promotion is explained. Bayesian Lasso, an extension of Lasso gives similar performance to Lasso and also estimates the variance in coefficient estimation. Group-Lasso, although similar to Lasso uses a regularization function which selects coefficients in groups. Group-Lasso assumes knowledge of the group structure of the coefficients. We also implemented the Bayesian Group-Lasso model for regression and compared its performance with Lasso and Group-Lasso. We also discussed the performance of Bayesian Group-Lasso in comparison with Lasso and Group-Lasso, when the number of samples changes and when group size changes.

Many interesting areas of research came up during the analysis and testing of these algorithms. One shortcoming of almost all the techniques discussed in this thesis is that they assume a linear relation between the target variable and features. One of the possible future work is to implement the Bayesian Group-Lasso concept using a Kernel function.

Another future work is to implement the Bayesian Group-Lasso concept using Expectation Maximization method (EM method). The current proposed method uses sampling as the implementation method. One disadvantage of using sampling is the heavy computational overhead. If we can find a solution for Bayesian Group-Lasso using EM method, the solution can be found much faster.
REFERENCES


BIOGRAPHICAL SKETCH

Manu Chandran received his Bachelor of Technology degree in electronics and communication engineering from Kerala University, India in 2006. He joined the Department of Electrical and Computer Engineering at the University of Florida in August 2008. As a graduate student, he worked in interdisciplinary microsystems group and control systems group at the University of Florida. He conducted research in feature selection and other machine learning techniques. He received his Master of Science degree from the Department of Electrical and Computer Engineering in the summer of 2011. His other areas of interests include image processing, signal processing and embedded systems.