1-1-2016

Feature Grouping Using Weighted L1 Norm For High-Dimensional Data

Karthik Kumar Padthe`
Wayne State University,

FEATURE GROUPING USING WEIGHTED ℓ_1 NORM FOR HIGH-DIMENSIONAL DATA.

by

KARTHIK KUMAR PADTHE

THESIS

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

2016

MAJOR: COMPUTER SCIENCE

Approved By:

_____

Advisor Date

DEDICATION

To my parents.

ACKNOWLEDGMENTS

I thank my advisor Dr. Chandan K. Reddy for guiding me throughout my masters. Without his guidance, this work would not be possible. I am grateful to him for providing me with an opportunity to start my research career. I would also like to thank him for providing me with all the resources and environment to learn new things and complete my thesis.

I am thankful to Dr. Zaki Malik and Dr. Zichun Zhong for taking time from their busy schedule and agreeing to be on my committee.

I thank Bhanukiran Vinzamuri for sharing his knowledge and helping me understand the nuances of research, I also thank Ping Wang for sharing her valuable knowledge with me. I am also thankful to all other students of Data Mining and Knowledge Discovery (DMKD) lab for accepting me as one of them.

A special thanks to my family, I thank my parents and siblings for providing me with emotional and financial support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Motivation and Overview

Feature selection [2–4] from real-world data is defined as a task of selecting a representative set of features which are useful for data mining tasks such as clustering, classification, to name a few. This task can further be divided into two categories, namely, supervised and unsupervised feature selection methods [2]. Unsupervised feature selection methods try to extract feature sets directly from the data and are used before applying clustering methods on the data. Supervised feature selection methods on the other hand try to extract features which are relevant to the class. These methods can be applied for both the classification and regression problems [3, 4].

One of the challenges involved in building effective supervised feature selection is to propose methods which can capture the relevance of features and groups of features with respect to a given class label. In this context, it is observed that groups of homogeneous features within a group can have a uniform effect on the class label. Homogeneity of features can be quantified using metrics such as correlation and feature dependence [3]. In such scenarios, it is desirable to build feature selection methods which can account for the entire group of homoegeneous features uniformly. This task is also called as *supervised feature grouping* [3, 5]. We simply refer to this task as feature grouping in the rest of this thesis. Some real-world examples of feature grouping are the following:

- **Healthcare Analytics**: Electronic health records (EHRs) contain patient information obtained from several disparate sources such as demographics, labs, comorbidities, medications and procedures. EHR driven phenotyping [6] is one of the emerging research areas where clinicians are interested in determining groups of features (phenotypes) across each of these sources which are important in determining the risk of the disease. These groups of features can also serve as biomarkers which can be used to track the progression of the disease for a patient [7].

- **Text Analytics**: Real-world text analytics datasets have a pre-defined hierarchical structure due to which there is an overlap of content among different documents. An illustration of this hierarchical structure for the 20-Newsgroups dataset [1] is given in Fig 1.1. Prediction models which can exploit groups of features representing each node in this hierarchy will be more effective than learning a unified model on the whole text corpus.

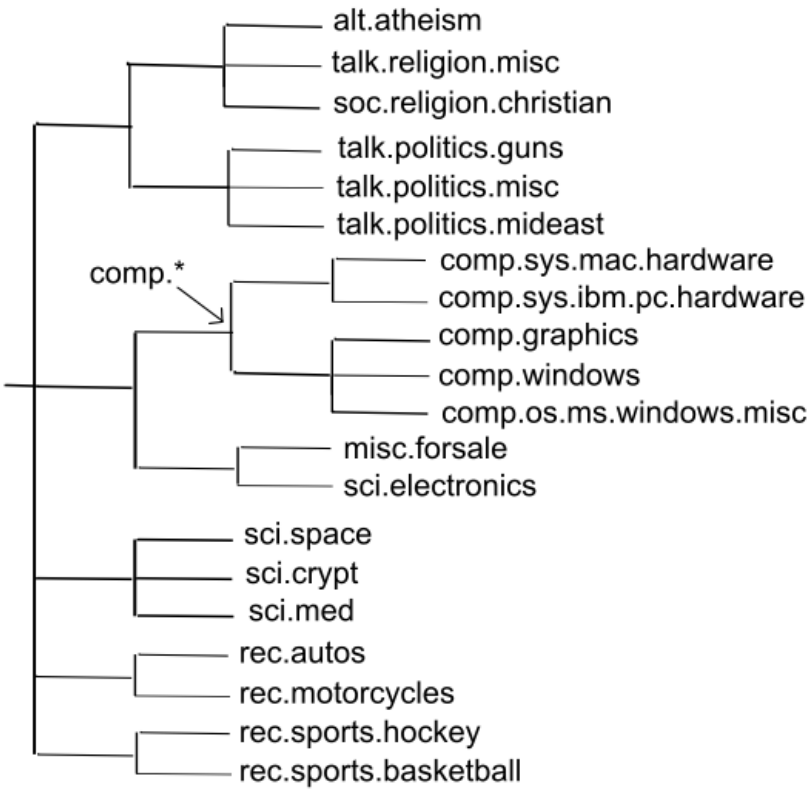


Figure 1.1: Hierarchical structure in 20-Newsgroups dataset [1].

One of the advantages of developing accurate feature grouping methods for such real-world datasets is to discover inherent feature groups present in the dataset, and then utilize structured sparsity methods such as the group lasso along with this discovered grouping structure to build effective models with good predictive ability [5, 8–11]. It is also desirable for regression models built on high-dimensional data to recover cohesive and homogeneous feature groups with good accuracy, as this reduces the error variance of the model and increases its generalizability. However, existing feature grouping methods are not capable

of extracting stable feature groups and resolving the misfusion problem which are explained below.

1.2 Unstable Feature Selection

In this section, we provide an illustration of the unstable feature selection problem using the Lasso method [12]. This is illustrated in Figure 1.2 which shows the behavior of Lasso when applied on a sample dataset with a feature space consisting of four unique groups of features, represented in four different colors.

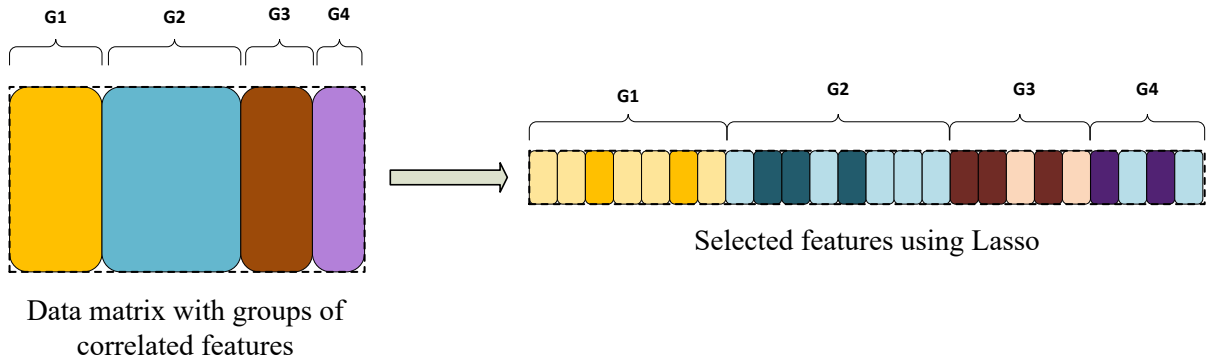


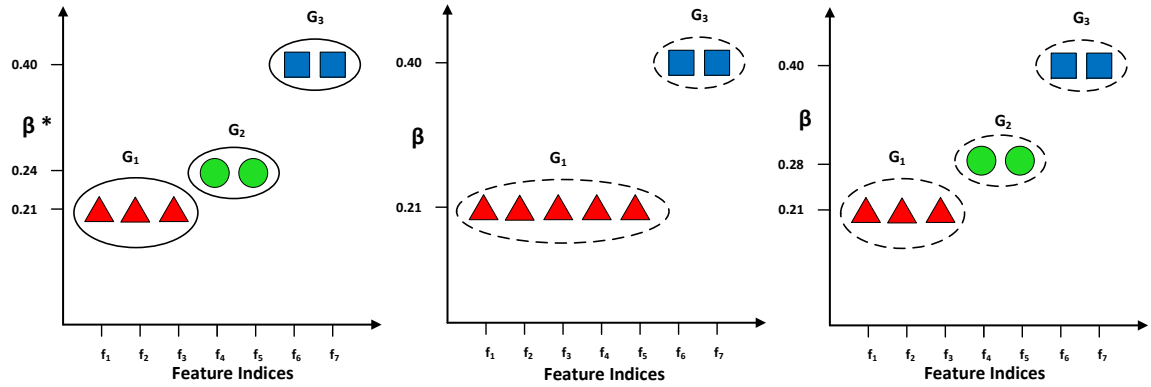
Figure 1.2: Unstable feature selection in Lasso.

From this illustration, where the selected features are represented with dark shaded colors and the features which are not selected i.e., the features with weight 0 are represented in lighter shade, one can clearly observe the problem associated with using Lasso for feature selection on groups of features. It randomly picks individual features among groups of features by discarding the rest which is incorrect. We now illustrate another problem associated with existing feature grouping methods which is called the misfusion problem [13].

1.3 The Misfusion Problem

In this section, we provide an illustration of the *misfusion* problem on small synthetic dataset. In Figure 1.3, we present a scenario of how feature grouping algorithms such as oscar are unable to resolve the misfusion problem . We consider a small dataset with seven features $F = \{f_1, f_2, \ldots, f_7, \}$ and plot these feature indices on the X-axis and their corresponding ground truth regression coefficient values β^* on the Y-axis in the left of Figure 1.3. Ground

truth β^* values are segregated into three groups which are G_1=$\{f_1, f_2, f_3\}$ with $\beta^*_{G_1}$=0.21, G_2=$\{f_4, f_5\}$ with $\beta^*_{G_2}$=0.24, and G_3=$\{f_6, f_7\}$ with $\beta^*_{G_3}$=0.4. The response variable Y=$X\beta^*$+ϵ is created where $X \in \mathbb{R}^{100\times 7}$ is a random feature vector matrix created using the normal distribution $\mathcal{N}(0,1)$, and ϵ is the error term which is created using $\mathcal{N}(0,1)$. Subsequently, we fit an existing state-of-the-art regression model (such as oscar method [14]) on this dataset and plot the learned regression coefficient values (β) on the Y-axis in Figure 1.3(b).



(a) ground truth (b) existing methods (c) our approach

Figure 1.3: A simple illustration demonstrating the misfusion problem and the results obtained by applying existing methods and our approach.

One can clearly observe from Figure 1.3(b) that oscar has *misfused* groups G_1 and G_2 without recovering G_2 correctly. This is due to the proximity of their regression coefficient values and oscar is unable to differentiate features in group G_1 from G_2. In contrast to existing methods, our approach presented in this thesis effectively resolves the misfusion problem as can be seen in Figure 1.3(c), with a minor trade-off being the complete recovery of the ground truth. This misfusion problem can be seen in many high-dimensional regression problems where coefficient values vary marginally across feature groups, and it needs to be addressed appropriately in order to build robust and accurate prediction models.

1.4 Our Contributions

The major contributions of this thesis are as follows:

- We propose a novel weighted ℓ_1 norm regularized linear regression algorithm for feature grouping which solves the misfusion problem to build a more effective predictive model

compared to existing feature grouping methods such as elastic net, fused-lasso and oscar.

- We formulate this as a convex optimization problem and solve it efficiently using the fast iterative soft-thresholding algorithm (FISTA).

- We evaluate the goodness of prediction of our approach using state-of-the-art convex and non-convex feature grouping methods on high-dimensional real-world datasets, namely 20-Newsgroups and breast cancer gene-expression datasets. We also evaluate our approach on four synthetic datasets and visualize the feature groups obtained.

1.5 Organization of this Thesis

In Chapter 2, we survey several existing convex and non-convex feature grouping methods, and also provide a brief review of proximal operators. In Chapter 3, we introduce the preliminaries needed to comprehend our weighted ℓ_1 norm-based framework. We formulate the corresponding weighted ℓ_1 regularized linear regression problem as a convex optimization problem and we provide an efficient algorithm for solving this problem. In Chapter 4, we evaluate the performance of weighted ℓ_1 norm-based model by comparing it with several convex and non-convex based feature grouping models on 20-Newsgroups data, gene-expresion data and synthetic datasets. In Chapter 5, we draw conclusions and provide directions for future work.

CHAPTER 2 A SURVEY OF FEATURE GROUPING ALGORITHMS

In this chapter, we review existing state-of-the-art feature grouping methods and we also present the required background for proximal gradient algorithms.

2.1 Related Background

We divide the literature being surveyed in this section into three parts: (i) graph-based convex methods, (ii) graph-based non-convex methods and (iii) other methods.

2.1.1 Graph-based Convex Methods

1. **Octagonal Selection and Clustering Algorithm for Regression (OSCAR)**: OSCAR [14] uses the combination of ℓ_1 norm which provides sparsity and the ℓ_∞ which encourages the equality of the coefficients, this regularizer can be written as follows:

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \underbrace{\lambda_1 \parallel \beta \parallel_1 + \lambda_2 \sum_{i<j} \max\{|\beta_i|, |\beta_j|\}}_{h(\beta)} \tag{2.1}$$

   where $X \in \mathbb{R}^{n \times p}$, $\beta \in \mathbb{R}^p$. When λ_2 is 0 this regularizer becomes Lasso, but when λ_1 is 0 this regularizer becomes the ℓ_∞ norm. The norm ball of this regularizer is octagonal in shape. OSCAR can be solved using quadratic programming (QP) and first-order methods efficiently.

2. **Graph Oscar (goscar)**: Graph-oscar [15] is a modified version of oscar [14] which uses a pre-specified directed feature graph. Its formulation is given below.

$$\arg\min_{\beta} \frac{1}{2} \parallel y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{(i,j)\epsilon E} max\{|\beta_i|, |\beta_j|\} \tag{2.2}$$

   Due to its convex formulation this optimization problem can be solved using the Alternate Direction Method for Multipliers (ADMM) method [16].

3. **Graph-guided Fused Lasso (gflasso)** : Gflasso [17] also uses the knowledge from

a pre-specified graph as in goscar. The resulting optimization problem for it is solved using coordinate descent method [18]. The formulation for the optimization problem is given below

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{(i,j)\epsilon E} \tilde{w}(i,j)|\beta_i - r(i,j)\beta_j| \tag{2.3}$$

In Eq. (2.3) λ_1 and λ_2 are regularization parameters, $w(i,j)$ is the absolute value of weight associated with edge between features i and j and $r(i,j)$ is $sign(w(i,j))$ which is the sign of the weight for an edge. The above formulation can be reduced as given below

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{(i,j)\epsilon E} |\beta_i - \beta_j| \tag{2.4}$$

2.1.2 Graph-based Non-convex Methods

Graph-based Non-convex methods provide certain advantages over convex methods as they can recover the sparse structure more efficiently and overcome the bias associated with convex methods in some cases. Hence, to overcome this problem, graph-based non-convex regularizers are used to handle feature grouping in the data. However, there is a trade-off here since these models are more difficult to solve. We describe important models that belong to this category in this section and also mention the algorithms that can be used to solve these methods. All these models mentioned in this thesis use the Difference of Convex functions (DC) programming method [19] to solve the optimization problem.

1. **Non-convex Feature Grouping and Selection (ncFGS):** NcFGS [20] uses the ℓ_1 norm for feature selection and the difference between absolute values of the coefficients of features connected in graph to perform feature grouping. The formulation is given below.

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{(i,j)\epsilon E} \tilde{w}(i,j)||\beta_i| - |\beta_j|| \tag{2.5}$$

In Eq. (2.5), second term in the penalty unlike gflasso formulation in Eq. (2.3) depends on the sign of the weights to decide whether β_i and β_j should be grouped together.

2. **Non-convex Truncated Feature Grouping and selection (ncTFGS):** NcTFGS [20] applies the ℓ_0 surrogate on both the ℓ_1 and the difference in absolute values of coefficients used in ncFGS. Here, the thresholding parameters are used to reduce the bias of the model. The formulation is given below.

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1 \sum_i J_\tau(|\beta_i|) + \lambda_2 \sum_{(i,j)\epsilon E} \tilde{w}(i,j) J_\tau(||\beta_i| - |\beta_j||) \tag{2.6}$$

where $J_\tau(x) = min(\frac{x}{\tau}, 1)$ is the threshold function used to reduce the estimation bias of the model.

3. **Non-convex Truncated Fused Feature Grouping and Selection (ncTF):** NcTF applies the ℓ_0 surrogate on the fusion penalty term of gflasso formulation as in Eq. (2.3). The formulation is given below.

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1||\beta||_1 + \lambda_2 \sum_{(i,j)\epsilon E} \tilde{w}(i,j) J_\tau(|\beta_i - r(i,j)\beta_j|) \tag{2.7}$$

4. **Non-convex Truncated ℓ_1 Feature Grouping and Selection (ncTL):** NcTL has the formulation similar to ncTF, but here ℓ_0 surrogate of ℓ_1 regularizer is used instead of using ℓ_0 surrogate of fusion penalty term. The formulation is given below.

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1 \sum_i J_\tau(|\beta_i|) + \lambda_2 \sum_{(i,j)\epsilon E} \tilde{w}(i,j)|\beta_i - r(i,j)\beta_j| \tag{2.8}$$

5. **Non-convex Truncated ℓ_1 and Fused Feature Grouping and Selection (ncTLF):** NcTLF has a formulation similar to ncTF and ncTL, but here the ℓ_0 surrogate is applied on both the terms of the regularizer instead of applying it on any one term. The

formulation is given below.

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1 \sum_i J_\tau(|\beta_i|) + \lambda_2 \sum_{(i,j)\epsilon E} \tilde{w}(i,j) J_\tau(|\beta_i - r(i,j)\beta_j|) \tag{2.9}$$

2.1.3 Other Methods

Both graph-based convex and non-convex methods need a pre-computed feature graph to be provided in order to recover feature groups, so they are not automatic feature grouping methods. We now discuss other methods which have been used for feature grouping. We provide some intuition on each of these regularizers below.

1. ℓ_q **norm**: ℓ_2 norm is defined as given below.

$$\| \beta \|_2^2 = \sum_{i=1}^{p} \beta_i^2 \tag{2.10}$$

The shape of this norm of spherical. The ℓ_1 norm is defined below which is convex and non-smooth.

$$\| \beta \|_1 = \sum_{i=1}^{p} |\beta_i| \tag{2.11}$$

However, the sparsity recovered using ℓ_1 can be biased sometimes. This motivates us to study non-convex ℓ_q norms with $q < 1$.

The ℓ_0 [19, 21] norm is known to produce a more efficient sparse solution compared to the ℓ_1 norm. The formulation is given below.

$$\| \beta \|_0 = |\{i : \beta_i \neq 0\}| \tag{2.12}$$

However, this regularizer is not capable of performing feature grouping. On the contrary, the ℓ_∞ norm is convex and it can perform feature grouping. The formulation is given below.

$$\| \beta \|_\infty = \max\{|\beta_1|, \ldots, |\beta_p|\} \tag{2.13}$$

2. **Elastic net**: The elastic net [22] is another convex regularizer which can perform feature grouping. It is defined as a convex combination of the ℓ_1 and ℓ_2 norms. The formulation is given below.

$$h(\beta) = (1-\alpha) \parallel \beta \parallel_1 + \alpha \parallel \beta \parallel_2^2 \tag{2.14}$$

Here, when $\alpha = 1$ this penalty becomes ℓ_2 penalty and when $\alpha = 0$ this penalty becomes ℓ_1 penalty. But when $\alpha \in (0, 1)$ this penalty will have the characteristics of both ℓ_1 and ℓ_2, ℓ_1 provides the property of parameter sparsity and ℓ_2 provides strictly convex nature. This penalty is also capable of soft feature grouping [23] in the presence of perfectly correlated variables.

3. **Fused-lasso**: Fused-lasso [24] uses the combination of ℓ_1 and a smoothness term to promote equality of coefficients among features to capture feature groups. This regularizer can be written as follows:

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{i=2}^{p} |\beta_i - \beta_{i-1}| \tag{2.15}$$

This regularizer is not capable of grouping the positive and negative variables together even if they have similar magnitude of regression coefficients. It also assumes a natural ordering of features in the dataset.

4. **Trace Norm:** This regularization model uses nuclear norm as the penalty term, this term is typically applied on matrices and acts in similar way as Lasso, this regularization is also known as trace-lasso [25]. Following is the formulation used:

$$\arg\min_{\beta} \frac{1}{2}||y - X\beta||_2^2 + \lambda_1 ||XDiag(\beta)||_* \tag{2.16}$$

This regularizer behaves like ℓ_2 regularizer in presence of strongly correlated predictors, but if the predictors are not correlated it will behave like Lasso.

2.2 Proximal Operator-based Methods

Proximal operators are widely used to solve convex optimization problems efficiently. Consider the formulation below consisting of a smooth convex function $f(x)$ and a non-smooth convex function $h(x)$.

$$\arg\min_{\beta} f(\beta) + h(\beta) \tag{2.17}$$

For a closed proper convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ the proximal operator $prox_f : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$\text{prox}_f(v) = \arg\min_{\beta}(f(\beta) + \frac{1}{2} \parallel \beta - v \parallel_2^2) \tag{2.18}$$

In each iteration of a proximal gradient method, the smooth convex function is minimized and then the effect of non-smooth convex function is incorporated. Proximal operator solves the problem of moving the weight vector v towards optimum of $h(\beta)$. Proximal operator for the regularization function $h(\beta)$ can be written as

$$\text{prox}_h(v) = \arg\min_{\beta}(h(\beta) + \frac{1}{2} \parallel \beta - v \parallel_2^2) \tag{2.19}$$

As this moves the initial weight vector v towards the minimum of the function $h(\beta)$ while still remaining close to v, $\text{prox}_h(v)$ is also known as a proximal point with respect to $h(\beta)$.

For a semi-continuous function $f(x)$ and a scalar value $\gamma > 0$, the Moreau envelope $f^\gamma(x)$ and proximal operator $\text{prox}_{\gamma f}(x)$ can be defined as [26]:

$$\begin{aligned} f^\gamma(x) &= \min_z \Big\{ f(z) + \frac{1}{2\gamma} \parallel z - x \parallel_2^2 \Big\} \leq f(x) \\ \underset{\gamma f}{\text{prox}}(x) &= \arg\min_z \Big\{ f(z) + \frac{1}{2\gamma} \parallel z - x \parallel_2^2 \Big\} \end{aligned} \tag{2.20}$$

Below we discuss two important properties of proximal operators and its relation to the gradient of the Moreau envelope function [26, 27].

- For a given function f, the proximal operator can be related to gradient-descent step. We consider a envelope function $f^\gamma(x)$ to prove this relation, the Moreau derivative

can be written as

$$\partial f^{\gamma}(x) = \partial \min_{z}\Big\{ f(z) + \frac{1}{2\gamma} \parallel z - x \parallel_2^2 \Big\} \;=\; \frac{1}{\gamma}\big[x - \hat{z}(x)\big] \tag{2.21}$$

where $\hat{z}(x) = \text{prox}_{\gamma f}(x)$ is the minimum value. Hence,

$$\underset{\gamma f}{\text{prox}} = x - \gamma \partial f^{\gamma}(x) \tag{2.22}$$

- Second, the proximal operator generalizes Euclidean projection. To demonstrate this, we consider a case where $f(x) = \iota_c(x)$ which is a set of indicator functions which belong to some convex set C, for these set of functions the proximal approximation can be written as $\text{prox}_f(x) = \arg\min_{z \in C} \parallel x - z \parallel_2^2$ which can be interpreted as Euclidean projection of x onto C.

CHAPTER 3 PROPOSED WEIGHTED ℓ_1 APPROACH

In this chapter we introduce the proposed weighted ℓ_1 norm based regression model and also we solve this optimization problem using a proximal operator based efficient solver. We also theoretically analyze the feature grouping nature of the model. Before introducing the details of the proposed model we present the required preliminaries.

3.1 Preliminaries

Table 3.1: Notations used in this thesis.

Notation	Description
n	number of instances.
p	number of features.
X	$\mathbb{R}^{n\times p}$ feature matrix.
y	$\mathbb{R}^n$ response variable.
β	$\mathbb{R}^p$ regression coefficient vector.
$\lvert x\rvert_\downarrow$	non-increasing sorted $\lvert x\rvert$.
$P(x)$	permutation matrix.
$\Omega(\beta)$	weighted ℓ_1 norm.
λ_1, λ_2	scalar regularization parameters.
w	$\mathbb{R}^p$ weight vector.
$J_\tau()$	truncated ℓ_1 norm.
E	connected graph of features.
K_m^+	monotone non-negative cone.

In this section, we present the preliminaries needed to comprehend our weighted ℓ_1 norm based algorithm for feature grouping. Table 3.1 presents important terms and notations used in this thesis. We now explain the interpretation of each of these notations in detail. Lowercase letters x, y denote column vectors and their transposes are denoted as x^T, y^T, respectively. The i^{th} and j^{th} components of these vectors are written as x_i and y_j. Matrices are written in uppercase (such as X) and the i^{th} column vector of X is represented as X_i. The vector with the absolute values of the components of the vector x is written as $|x|$. For a vector $x \in \mathbb{R}^p$, the i^{th} largest component of x is represented as $x_{[i]}$. This implies that $x_{[1]} \geq x_{[2]} \geq \ldots \geq x_{[p]}$. Using this analogy, we define $|x|_\downarrow$ which represents the vector obtained by sorting the absolute values vector of x (denoted by $|x|$) in non-increasing order so that

$|x|_{[1]} \geq |x|_{[2]} \ldots \geq |x|_{[p]}$ and the ties are broken arbitrarily. This vector based transformation of $|x|$ to $|x|_{\downarrow}$ can be done using the permutation matrix P i.e., $|x|_{\downarrow} = P(|x|)|x|$. The permutation matrix follows the property, $P(|x|)^{-1} = P(|x|)^T$, and it sorts the entries of $|x|$ in a non-increasing order. For any given weight vector $w \in \mathbb{R}^{p+}$ such that $w_1 \geq w_2 \geq \ldots w_p \geq 0$, $\Delta_w = min\{w_l - w_{l+1}, l = 1, 2, \ldots, p-1\}$ is the minimum gap between consecutive components of the weight vector w. With this background, we now discuss the formulation of oscar briefly and introduce the weighted ℓ_1 norm.

Oscar is convex and shape of the ball is octagonal. The oscar regularizer is defined as given in Eq. (3.1), where the ℓ_1 term promotes sparsity and the pairwise ℓ_∞ term promotes equality in magnitude of each pair of elements $|\beta_i|, |\beta_j|$ among the $\frac{p(p-1)}{2}$ feature pairs present in the dataset. This can also be interpreted as the feature grouping component of oscar.

$$h(\beta) = \lambda_1 \parallel \beta \parallel_1 + \lambda_2 \sum_{i<j} max\{|\beta_i|, |\beta_j|\} \tag{3.1}$$

We now define the weighted ℓ_1 norm and the regularized linear regression problem in Eq. (3.2).

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \parallel y - X\beta \parallel_2^2 + \Omega(\beta) \tag{3.2}$$

$$\Omega(\beta) = \parallel w \odot |\beta|_{\downarrow} \parallel_1$$

In this equation, w is a weight vector of non-increasing weights, which is defined as $w = \{w_1 \geq w_2 \geq \ldots w_p \geq 0\}$ and $\odot$ is the element-wise multiplication (Hadamard Product). This can be written as $w \in \mathrm{K}_m^+$ which represents the monotone non-negative cone [28]. This definition of the weighted ℓ_1 norm now makes the oscar regularizer a specific case of this weighted ℓ_1 problem with the weights as $(w_i = \lambda_1 + \lambda_2(p-i) \quad \forall i = 1, 2, \ldots, p)$. Apart from oscar, other regularizers such as the lasso and ℓ_∞ also become special cases of the weighted ℓ_1 norm. When all the w_i values are fixed, the weighted ℓ_1 norm becomes the weighted lasso. Similarly, when $w_1 = 1$ and $w_i = 0 \; \forall \; i = 2, 3, \ldots, p$, then the weighted ℓ_1 norm becomes the ℓ_∞ norm.

A Moreau proximal operator [26] can be derived to solve such regularized problems, as it can be interpreted as a gradient-descent step for the objective function. Proximal operators also have a distinct advantage when dealing with non-smooth regularizers such as the weighted ℓ_1 norm, as they are a generalization of the projection operator, which in turn is used to solve non-smooth optimization problems. In the next section, we derive the proximal operator for the weighted ℓ_1 norm and use it within an accelerated proximal gradient (APG) algorithm for solving this problem efficiently.

3.2 The Proposed Method

In this section, we present an accelerated proximal gradient FISTA algorithm to solve the weighted ℓ_1 norm regularized linear regression problem. This algorithm uses the proximal operator for the weighted ℓ_1 norm and we present the method for obtaining it efficiently. Subsequently, we provide theoretical analysis where we prove the convexity and the feature grouping property of this weighted ℓ_1 norm which proves why it is effective at resolving the misfusion problem.

3.2.1 Proximal operator for Weighted ℓ_1 Norm

The proximal operator for Ω, which is denoted by $\text{prox}_\Omega(.)$, is defined in Eq. (3.3) for any $v \in \mathbb{R}^p$ using the standard definition of a proximal operator proposed in [26]. We now try to simplify the proximal operator using the steps provided below and explain the procedure for obtaining it.

$$\text{prox}_\Omega(v) = \arg\min_{\beta \in \mathbb{R}^p} \left(\frac{1}{2} \parallel \beta - v \parallel_2^2 + \Omega(\beta) \right) \tag{3.3}$$

Using Eq. (3.3) we can estimate $\text{prox}_\Omega(v)$ in order to employ it within the FISTA framework. We use the fact that $w, \beta \in \text{K}_m^+ \subset \mathbb{R}^p$ and mention the steps needed to simplify

Eq. (3.3) further as follows:

$$\begin{aligned} \text{prox}_{\Omega}(v) &= \arg\min_{\beta \in \mathrm{K}_m^+} \frac{1}{2} \parallel \beta - v \parallel_2^2 + w^T \beta \\ &= \arg\min_{\beta \in \mathrm{K}_m^+} \frac{1}{2} \parallel \beta - (v - w) \parallel_2^2 \\ &\text{s.t} \quad \beta_1 \geq \beta_2 \geq \ldots \geq \beta_p \geq 0 \end{aligned} \tag{3.4}$$

The simplification yields Eq. (3.4) which needs to be solved to obtain $\text{prox}_{\Omega}(v)$. This computation can be interpreted as consisting of two operations which are (i) obtaining the projection $(v - w)$ onto the monotone cone $\mathrm{K}_m = \{\beta_1 \geq \beta_2 \geq \ldots \geq \beta_p\}$ by solving Eq. (3.5), and (ii) applying a subsequent projection of this result onto $\mathbb{R}^{p+}$ by clipping the negative values.

$$\begin{aligned} &\arg\min_{\beta \in \mathrm{K}_m} \frac{1}{2} \parallel \beta - (v - w) \parallel_2^2 \\ &\text{s.t} \quad \beta_1 \geq \beta_2 \geq \ldots \geq \beta_p \end{aligned} \tag{3.5}$$

This projection problem in Eq. (3.5) has the form as given in Eq. (3.6) which is also called the isotonic regression problem which is a submodular convex optimization problem [29]. Hence, in order to solve Eq. (3.5), we use an existing isotonic regression solver like the pool adjacent violators algorithm (PAVA) [30].

$$\begin{aligned} &\arg\min_{y \in \mathbb{R}^p} \sum_{i=1}^{p} f_i(y_i) \\ &\text{s.t} \quad y_1 \leq y_2 \leq \ldots \leq y_p \end{aligned} \tag{3.6}$$

PAVA is one of the most efficient methods for solving the isotonic regression problem with $O(plogp)$ time complexity [31]. We briefly describe the intuition behind this algorithm. PAVA computes a non-decreasing sequence of y_i such that the problem is optimized. It starts with y_1 on the left and moves to the right until it encounters the first violation $y_i > y_{i+1}$. Once it encounters the violation it forms a block of y_i and y_{i+1}, then computes a update based on a solver that results in $y_{i+1} = s(y_i)$ as needed to get the monotonicity. Then, it

continues to the right until it finally reaches y_p. By applying this PAVA algorithm to solve Eq. (3.5) and then by applying the clipping operator to project the result onto $\mathbb{R}^{p+}$, we obtain $\text{prox}_\Omega(v)$. This proximal operator is now used within the FISTA based algorithm given in Algorithm 3.1 which is the proposed weighted ℓ_1 norm regularized linear regression solver.

3.2.2 FISTA based Algorithm

In this section, we present the solver for the weighted ℓ_1 norm regularized linear regression problem which uses the fast iterative soft-thresholding algorithm (FISTA) [32]. FISTA is a variant of the iterative soft-thresholding algorithm (ISTA) which uses the accelerated proximal gradient (APG) method based on Nesterov's technique [33]. First-order optimization methods such as FISTA converge as $\text{O}(\frac{1}{n^2})$ compared to the traditional gradient methods which have a slow convergence rate of $\text{O}(\frac{1}{\sqrt{n}})$.

Algorithm 3.1: FISTA based Solver for the weighted ℓ_1 norm regularized linear regression.

Input: Feature vector $X \in \mathbb{R}^{n \times p}$, Response vector $Y \in \mathbb{R}^n$, Lipschitz constant $L = 2\Lambda_{max}(X^T X)$, Weight vector w, Tolerance parameter tol, max iterations max_iter.
Output: Regression coefficients $\beta \in \mathbb{R}^p$
Initialize: $\beta_0 \in \mathbb{R}^p, u_1 = \beta_0, t_1 = 1$;
for *k=1 to max_iter* **do**
  $\beta_k = \text{prox}_\Omega\left(u_k - X^T(Xu_k - y)/L\right)$ using Eq. (3.4) ;
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
  $u_{k+1} = \beta_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\beta_k - \beta_{k-1})$;
  if $\| \beta_k - \beta_{k-1} \|_2 < tol$ **then**
    break;
  end
  $k = k + 1$;
end
Return β_k ;

In Algorithm 3.1, we describe the FISTA based algorithm used to learn the regression coefficient vector. The inputs to the algorithm are X, Y, the Lipschitz constant L which

is estimated using the maximum value among all the Eigen values ($\Lambda(X^TX)$). The weight vector w is also provided, and it is used for the weighted ℓ_1 norm computation as given in Eq. (3.2). w satisfies the property that $w \in \mathbb{K}_m^+$ such that $w_1 \geq w_2 \geq \ldots \geq w_p \geq 0$. In this algorithm, after initializing the parameters, in line 3, prox_Ω is computed by solving Eq. (3.4) using the PAVA algorithm and the subsequent projection using the clipping operator onto $\mathbb{R}^{p+}$. In Lines 4 and 5, the updates are done as per the accelerated proximal gradient method. Subsequently, in lines 6-10, the convergent regression coefficient vector is returned.

3.2.3 Complexity Analysis

We now discuss the complexity of the weighted ℓ_1 norm regularized linear regression algorithm presented above. The number of iterations for the FISTA algorithm to obtain an ϵ-optimal solution is $O(1/\sqrt{\epsilon})$. The computation of the proximal operator for the weighted ℓ_1 norm requires solving Eq. (3.5) which has a time complexity of $O(plogp)$ as mentioned earlier for the PAVA algorithm. The projection onto $\mathbb{R}^{p+}$ using the clipping operator takes constant time. Hence, the total time complexity of the algorithm is $O\left(\frac{1}{\sqrt{\epsilon}}(p(n+logp)\right)$. We observe that for most of the real-world datasets $n \gg logp$ and hence the complexity of this algorithm is $O(np/\sqrt{\epsilon})$.

3.2.4 Theoretical Analysis

In this section, we prove the convexity and the feature grouping property of the weighted ℓ_1 norm. Before we state the theorem and provide its proof, we introduce several lemmas which state the convexity, norm property and the strong Schur convexity [34, 35] properties of the weighted ℓ_1 norm. These lemmas will also be used to prove the feature grouping property.

lemma 1. $\Omega(x)$ *is a convex function.*

Proof. Let $u, v \in \mathbb{R}^p$, $\theta \in [0,1]$, $x = \theta\, u + (1-\theta)\, v$, then

$$\begin{aligned}
\Omega(x) =& \| \; w \odot |x|_{\downarrow} \; \|_1 \\
=& \| \; w \odot P(|x|)|x| \; \|_1 \\
=& \| \; w \odot P(|x|)|(\theta u + (1-\theta)\, v)| \; \|_1 \\
\leq& \| \; w \odot P(|x|)(|\theta u| + |(1-\theta)\, v)| \; \|_1 \\
\leq& \; \theta \; \| \; w \odot P(|x|)u \; \|_1 + (1-\theta) \; \| \; w \odot P(|x|)v \; \|_1 \\
\leq& \; \theta \; \| \; w \odot P(|u|)u \; \|_1 + (1-\theta) \; \| \; w \odot P(|v|)v \; \|_1 \\
\leq& \; \theta \; \| \; w \odot P(|u|)|u| \; \|_1 + (1-\theta) \; \| \; w \odot P(|v|)|v| \; \|_1 \\
\leq& \; \theta \; \Omega(u) + (1-\theta) \; \Omega(v)
\end{aligned} \tag{3.7}$$

□

Here we assume without loss of generality that the permutation matrices for x, u, v vectors are the same. While deriving this proof, we applied the following properties for the absolute value function: for $u, v \in \mathbb{R}^p$, $|u+v| \leq |u| + |v|$ and $u \leq |u|$ to prove the convex function property.

lemma 2. *If $w \in \mathrm{K}_m^+$ then $\Omega(x)$ satisfies the conditions of a norm.*

Proof. To prove that $\Omega(x)$ is a norm, we need to prove the *definiteness* condition that $\Omega(x) = 0 \iff x = 0$. As $w \in \mathrm{K}_m^+$ and $\Omega(x) = 0$ only if $x = 0$. The vice-versa statement is also true that if $x = 0$, then $\Omega(x) = 0$ using the definition of the weighted ℓ_1 norm from Eq. (3.2). The *positive homogeneity* condition which states that $\Omega(\alpha x) = \alpha\; \Omega(x)$ for any $\alpha \geq 0$ can also be proved trivially using the definition of the weighted ℓ_1 norm. We can also prove the *triangle inequality* condition that $\Omega(u+v) \leq \Omega(u) + \Omega(v)$ for any two vectors $u, v \in \mathbb{R}^p$ by following the steps similar to those provided in Lemma 1. This proves that $\Omega(x)$ satisfies the conditions of a norm. □

We now present a lemma which is based on the strong Schur convexity of the weighted ℓ_1 norm.

lemma 3. *Consider a vector $\beta \in \mathbb{R}^{p+}$ and two of its components β_i and β_j, such that $\beta_i > \beta_j$. Let $z \in \mathbb{R}^{p+}$ be obtained by applying to β an increment of $\epsilon \in (0, (\beta_i - \beta_j)/2)$, so that $z_i = \beta_i - \epsilon$, $z_j = \beta_j + \epsilon$, $z_k = \beta_k$, for $k \neq i, j$. Then*

$$\Omega(\beta) - \; \Omega(z) \geq \Delta_w \epsilon \tag{3.8}$$

Proof. x_i and x_j are non-negative and let l and m be their respective rank orders, i.e., $x_i = x_{[l]}$ and $x_j = x_{[m]}$; of course, $l < m$, because $x_i = x_{[l]} > x_{[m]} = x_j$. Now let $l + a$ and $m - b$ be the rank orders of z_i and z_j, respectively, i.e., $x_i - \epsilon = z_i = z_{[l+a]}$ and $x_j + \epsilon = z_j = z_{[m-b]}$. Of course, it may happen that a or b (or both) are zero, if ϵ is small enough not to change

the rank orders of one (or both) of the affected components of x. Furthermore, the condition $\epsilon < (x_i - x_j)/2$ implies that $x_i - \epsilon > x_j + \epsilon$, thus $l + a < m - b$. A key observation is that $x_\downarrow$ and $z_\downarrow$ only differ in positions l to $l + a$ and $m - b$ to m, thus we can write

$$\Omega_w(x) - \Omega_w(z) = \sum_{k=l}^{l+a} w_k(x_{[k]} - z_{[k]}) + \sum_{k=m-b}^{m} w_k(x_{[k]} - z_{[k]}). \tag{3.9}$$

In the range from l to $l + a$, the relationship between $z_\downarrow$ and $x_\downarrow$ is

$$z_{[l]} = x_{[l+1]}, z_{[l+1]} = x_{[l+2]}, ..., z_{[l+a-1]} = x_{[l+a]}, z_{[l+a]} = x_{[l]} - \epsilon,$$

whereas in the range from $m - b$ to m, we have

$$z_{[m-b]} = x_{[m]} + \epsilon, z_{[m-b+1]} = x_{[m-b]}, ..., z_{[m]} = x_{[m-1]}.$$

Plugging these equalities into Eq. (3.9) yields

$$\begin{aligned}\Omega_w(x) - \Omega_w(z) = &\sum_{k=l}^{l+a-1} w_k \underbrace{(x_{[k]} - x_{[k+1]})}_{\geq 0} + \sum_{k=m-b+1}^{m} w_k \underbrace{(x_{[k]} - x_{[k-1]})}_{\leq 0} \\ &+ w_{l+a}(x_{[l+a]} - x_{[l]} + \epsilon) + w_{m-b}(x_{[m-b]} - x_{[m]} - \epsilon)\end{aligned}$$

$$\begin{aligned}&\overset{(A)}{\geq} w_{l+a} \sum_{k=l}^{l+a-1} (x_{[k]} - x_{[k+1]}) + w_{m-b} \sum_{k=m-b+1}^{m} (x_{[k]} - x_{[k-1]}) \\ &\quad + w_{l+a}(x_{[l+a]} - x_{[l]} + \epsilon) + w_{m-b}(x_{[m-b]} - x_{[m]} - \epsilon)\end{aligned}$$

$$\begin{aligned}&= w_{l+a}\left(\sum_{k=l}^{l+a-1} (x_{[k]} - x_{[k+1]}) + (x_{[l+a]} - x_{[l]} + \epsilon) \right) \\ &\quad + w_{m-b}\left(\sum_{k=m-b+1}^{m} (x_{[k]} - x_{[k-1]}) + (x_{[m-b]} - x_{[m]} - \epsilon) \right) \\ &\overset{(C)}{=} \epsilon(w_{l+a} - w_{m-b}) \overset{(C)}{\geq} \epsilon\Delta_w,\end{aligned}$$

where inequality (A) results from $x_{[k]} - x_{[k+1]} \geq 0$, $x_{[k]} - x_{[k-1]} \leq 0$, and the components of w forming a non-increasing sequence, thus $w_{l+a} \leq w_k$, for $k = l, ..., l + a - 1$, and $w_{m-b} \geq w_k$, for $k = m - b + 1, ..., m$; equality (C) is a consequence of the cancellation of the remains of the telescoping sums with the two other terms; inequality (C) results from the fact that $l + a < m - b$ and the definition of Δ_w. □

Theorem 1. *Let $\hat{\beta}$ be a solution of Eq. (3.2), and let X_i and X_j be two columns of X. Then,*

- $\| X_i - X_j \|_2 < \Delta_w / \| y \|_2 \implies \hat{\beta}_i = \hat{\beta}_j$, *and*

- $\| X_i + X_j \|_2 < \Delta_w / \| y \|_2 \implies \hat{\beta}_i = -\hat{\beta}_j$

Proof. We prove this property by first mentioning that if f is a convex function and $\hat{\beta} \in dom(f)$. Then $\hat{\beta} \in \arg\min f$, if and only if $f'(\hat{\beta}, u) \geq 0$, for any u where $f'(\beta, u)$ is the directional derivative of function f in the direction u. Eq. (3.2) can be written as the sum of two components $f(\beta) = L(\beta) + \Omega(\beta)$ where $L(\beta) = \frac{1}{2} \| y - X\beta \|_2^2$. We begin by stating that given $\| X_i - X_j \|_2 < \Delta_w / \| y \|_2$ is satisfied for some pair of columns, then consider some $\hat{\beta}$ such that $\hat{\beta}_i \neq \hat{\beta}_j$ and we can assume that $\hat{\beta}_i > \hat{\beta}_j$. The directional derivative of L at $\hat{\beta}$, in the direction u, where $u_i = -1, u_j = 1$, and $u_k = 0$, for $k \neq i, j$, is.

$$\begin{aligned} L'(\hat{\beta}, u) &= \lim_{\alpha \to 0^+} \frac{\| y - X\hat{\beta} + \alpha(X_i - X_j) \|_2^2}{2\alpha} - \frac{\| y - X\hat{\beta} \|_2^2}{2\alpha} \\ &= g^T (X_i - X_j) \end{aligned} \tag{3.10}$$

where $g = y - X\hat{\beta}$. Similarly, we can compute the directional derivative of Ω at $\hat{\beta}$, in the same direction u

$$\Omega'(\hat{\beta}, u) = \lim_{\alpha \to 0^+} \frac{\Omega(\hat{\beta} + \alpha u) - \Omega(\hat{\beta})}{\alpha} \tag{3.11}$$

In Eq. (3.11), we can use **Lemma 3** and Eq. (3.8) and this can be re-written as

$$\Omega'(\hat{\beta}, u) \leq \lim_{\alpha \to 0^+} \frac{-\Delta_w \alpha}{\alpha} = -\Delta_w \tag{3.12}$$

We can now combine Eq. (3.10) and Eq. (3.12) to obtain the directional derivative of f as

$$\begin{aligned} f'(\hat{\beta}, u) &\leq g^T (X_i - X_j) - \Delta_w \\ &\leq \| g \|_2 \| X_i - X_j \|_2 - \Delta_w \\ &\leq \| y \|_2 \| X_i - X_j \|_2 - \Delta_w < 0. \end{aligned} \tag{3.13}$$

In Eq. (3.13), we used our assumption at the beginning that $\| X_i - X_j \|_2 < \Delta_w / \| y \|_2$ to arrive at $f'(\hat{\beta}, u) < 0$. However, this is a contradiction to our assumption using the convex function property stated earlier that $f'(\hat{\beta}, u) \geq 0$. Hence, using proof by contradiction, we conclude that $\hat{\beta}_i = \hat{\beta}_j$. This implies that Ω assigns coefficients of the same magnitude for similar features essentially grouping them into a cluster.

The second part of this theorem is simply a corollary of the first part which results from swapping the signs of either X_i or X_j and the corresponding coefficient. If two columns are similar, then any $\Delta_w > 0$ is sufficient to guarantee that these two columns (features) will be grouped together, and their corresponding regression coefficient values will have the same magnitude. This completes the proof explaining how the weighted ℓ_1 norm performs exact feature grouping and resolves the misfusion problem . □

CHAPTER 4 EXPERIMENTAL RESULTS

In this chapter, we present the experiments conducted to evaluate the performance of our weighted ℓ_1 approach. We explain the details of the synthetic datasets and also describe the real-world datasets used. We explain the baseline models, evaluation metrics and the implementation details of these methods. We conduct different experiments to assess the recovery of feature groups, goodness of prediction and scalability of the proposed approach.

4.1 Datasets Description

In this section, we describe the datasets considered for evaluating the performance of our weighted ℓ_1 approach. We provide details regarding the synthetic dataset creation which is followed by describing the 20-Newsgroups and breast cancer datasets.

4.1.1 Synthetic Datasets

We created three synthetic datasets with moderate dimensionality (*Syn-1*, *Syn-2* and *Syn-3*) and one high-dimensional dataset (*Syn-4*). We include a feature grouping pattern in these datasets which is specified below. This allows to visualize the goodness of feature grouping methods for the moderate dimensionality datasets. The response variable in these datasets is created using the linear regression model which can be written as $y = X\beta^* + \epsilon$ where $\beta^* \in \mathbb{R}^p$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the error term. Features for these datasets are generated as $X \sim \mathcal{N}(0, C)$ where C=$[c_{ij}]$ is a covariance matrix.

1. In *Syn-1*, n=280 and there are 8 predictors; the parameters are generated as follows

$$\beta^* = [3, 2, 1.5, 0, 0, 0, 0, 0]^T$$

   and σ=3, with covariance c_{ij}=$0.7^{|i-j|}$.

2. In *Syn-2*, n=280 and there are 8 predictors; the parameters are generated as follows

$$\beta^* = [3, 0, 0, 1.5, 0, 0, 0, 2]^T$$

   and σ=3, with covariance c_{ij}=$0.7^{|i-j|}$.

3. In *Syn-3*, n=800 and there are 40 predictors; the parameters are generated as follows

$$\beta^* = [\underbrace{0, \cdots, 0}_{10}, \underbrace{2, \cdots, 2}_{10}, \underbrace{0, \cdots, 0}_{10}, \underbrace{2, \ldots, 2}_{10}]^T$$

and σ=15, with covariance c_{ij}=0.5 when $i \neq j$, and 1 otherwise.

4. In *Syn-4*, n=2000 and there are 5000 predictors; the parameters are generated as follows

$$\beta^* = [\underbrace{3, \cdots, 3}_{0.1p}, \underbrace{0, \cdots, 0}_{0.3p}, \underbrace{1.5, \cdots, 1.5}_{0.1p}, \underbrace{0, \cdots, 0}_{0.4p}, \underbrace{2, \cdots, 2}_{0.1p}]^T$$

and σ=3, with covariance c_{ij}=$0.7^{|i-j|}$.

Table 4.1: Description of the datasets used in our experiments.

Dataset	**# Features**	**# Instances**
Syn-1	8	280
Syn-2	8	280
Syn-3	40	800
Syn-4	5000	2000
breast-cancer	8141	295
atheism vs graphics	7943	2000
windows.x vs religion.misc	8442	2000
autos vs motorcycles	7094	2000
baseball vs hockey	7909	2000
forsale vs ms-windows.misc	6678	2000
guns vs mideast	9763	2000
med vs space	8778	2000
pc.hardware vs politics.misc	8001	2000
mac.hardware vs christian	7288	1997
crypt vs electronics	7410	2000

4.1.2 20-Newsgroups Dataset

This dataset is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups[1]. We extract 10 pairs from the 20 different newsgroups to form 10 datasets as given in Table 4.1. We treat each of these 10 pairs as a binary

[1] `http://qwone.com/~jason/20Newsgroups/`

classification problem, wherein we label each document in the dataset with the newsgroup it belongs to. As a part of the preprocessing step, we perform stemming to reduce the redundancy of words and remove the stop words. We only consider words which appear in atleast 4 documents. Subsequently, we build a weight matrix using the TF-IDF method which is commonly used in text analytics to obtain a feature vector based representation.

4.1.3 Breast Cancer Dataset

We use a high-dimensional breast cancer gene expression dataset[2] in our experiments. This dataset contains information about 8,141 genes for 295 breast cancer tumors. These tumor information were collected from 295 women suffering from breast cancer. Out of the 295 tumors, 78 are metastatic (which are labeled as 1) and 217 are non-metastatic (which are labeled as -1). To decrease the class imbalance, we duplicate the metastatic class instances twice before evaluating the performance of the models used here. This helps to obtain unbiased results.

4.2 Performance Evaluation

In this section, we present the metrics used for evaluating our weighted ℓ_1 approach. We use the following metrics to compare the performance of the proposed model with the baseline models: Area Under ROC Curve (AUC) (including standard deviation and p-values), Mean Squared Error (MSE), and the coefficient of determination R-squared (R^2).

4.3 Implementation Details

Our proposed weighted ℓ_1 norm and its corresponding proximal operator was implemented in R. The isotone R-package is used to implement the PAVA algorithm. The R-package Sparse Modeling Software (*SPAMS*) was used to implement methods such as elastic net, graph-ridge, ℓ_0, ℓ_∞, fused-lasso and trace-lasso algorithms. We use the R-package Feature Grouping and Selection over Undirected Graph (*FGSG*) to implement the graph-based models such as goscar, gflasso, ncFGS, ncTFGS, ncTL, ncTF, and ncTLF. To calculate AUC and R^2 we use R-packages *pROC* and *Metrics*. We calculate the MSE for synthetic datasets

[2] `http://lbbe.univ-lyon1.fr/-Jacob-Laurent-.html?lang=fr`

with a known ground truth β^* using this formula: $MSE = \frac{1}{n}(\beta - \beta^*)^T X^T X(\beta - \beta^*)$, where β is the learned regression coefficient vector after applying the corresponding feature grouping algorithm [36]. These metrics were obtained using five-fold cross validation. Parameter tuning of the regularization parameters was done using a hold out set for all the graph-based convex and non-convex models. The weight vector (w), which follows a pre-specified ordering in our weighted ℓ_1 approach, was generated using a Gaussian Benjamini-Hochberg (BHq) procedure [37].

Table 4.2: MSE (std) values of our weighted ℓ_1 approach compared with other methods for synthetic datasets.

Method	***Syn-1***	***Syn-2***	***Syn-3***
elastic net	1.370 (0.086)	1.382 (0.165)	2.954 (0.325)
fused-lasso	1.032 (0.209)	1.142 (0.137)	2.888 (0.442)
ℓ_∞	1.678 (0.104)	1.750 (0.146)	2.834 (0.344)
graph-ridge	1.575 (0.156)	1.576 (0.178)	2.881 (0.362)
goscar	1.509 (0.156)	1.529 (0.132)	2.918 (0.382)
gflasso	1.593 (0.137)	1.650 (0.178)	2.879 (0.428)
trace-lasso	1.681 (0.192)	1.776 (0.218)	2.888 (0.414)
ncFGS	1.568 (0.118)	1.655 (0.204)	2.832 (0.274)
ncTFGS	1.530 (0.132)	1.632 (0.216)	2.814 (0.413)
ncTF	1.643 (0.145)	1.589 (0.215)	2.880 (0.270)
ncTL	1.652 (0.136)	1.611 (0.236)	2.834 (0.344)
ncTLF	1.606 (0.1502)	1.529 (0.159)	2.822 (0.322)
weighted ℓ_1	**0.543 (0.052)**	**0.454 (0.051)**	**1.762 (0.238)**

4.4 Goodness of Prediction

In this section, we present the results corresponding to the goodness of prediction of our proposed approach. In Table 4.2, we present results obtained using the mean squared error (MSE) and the standard deviation estimated by bootstrapping with 500 resamplings. We observe that our weighted ℓ_1 approach obtains lower MSE values compared to the other competing models. In Table 4.3, we also provide the coefficient of determination (R^2) values. A model is good when it has low MSE and high R^2 values. These results indicate that our method provides the best fit compared to all other methods. This better performance is due to the effective feature grouping ability of our approach which helps in building more

effective and generalizable models.

Table 4.3: R^2 values of our weighted ℓ_1 approach compared with other methods for synthetic datasets.

Method	***Syn-1***	***Syn-2***	***Syn-3***
elastic net	0.305	0.318	0.103
fused-lasso	0.320	0.321	0.105
ℓ_∞	0.288	0.301	0.100
graph-ridge	0.289	0.303	0.104
goscar	0.319	0.312	0.107
gflasso	0.315	0.318	0.104
trace-lasso	0.310	0.305	0.105
ncFGS	0.323	0.323	0.114
ncTFGS	0.320	0.320	0.110
ncTF	0.313	0.318	0.112
ncTL	0.321	0.311	0.110
ncTLF	0.318	0.317	0.111
weighted ℓ_1	**0.354**	**0.345**	**0.377**

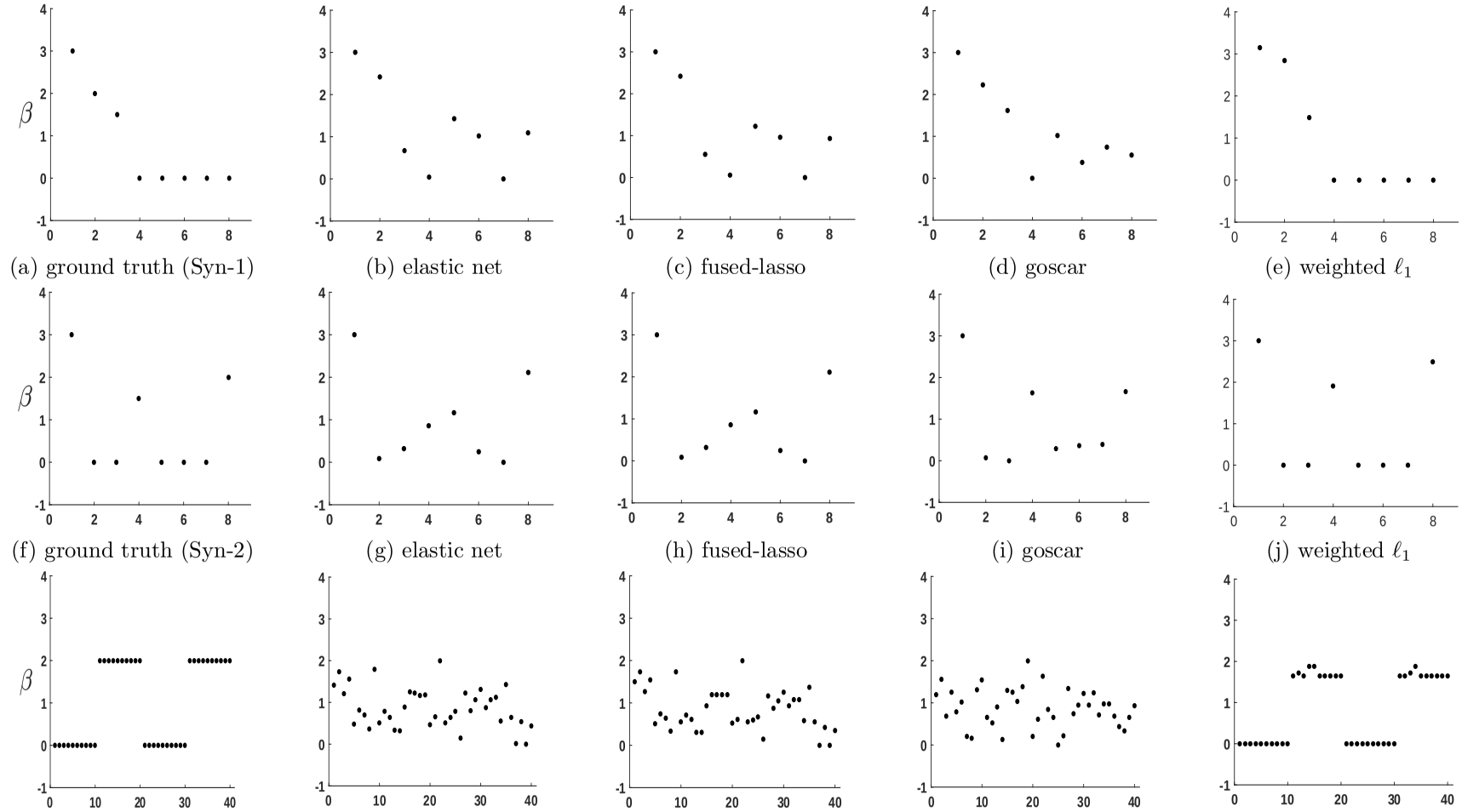
In Table 4.4, we provide the AUC (along with the standard deviation of the result using five-fold cross validation), p-values for our weighted ℓ_1 approach to confirm the performance and the statistical significance of our results. The p-value is calculated using Delong's test for comparing the significance between a pair of AUC values [38]. We compute the p-value by comparing the result obtained after applying our approach with the second best performing model (trace-lasso) for each dataset considered. It should be noted that a result with a p-value of less than 0.05 is considered to be *statistically significant* and is interpreted as being small enough to justify the superiority of our approach over the methods used for comparison.

4.5 Recovering Feature Groups

In this section, we conduct an experiment to visually assess the goodness of our weighted ℓ_1 approach compared to other feature grouping methods for *Syn-1*, *Syn-2* and *Syn-3* datasets. In Figure 4.1, the y-axis represents the feature regression coefficients obtained after fitting four different feature grouping algorithms for all three synthetic datasets and the x-axis represents the feature indices. The first, second and third rows in Figure 4.1 correspond to

Table 4.4: AUC (std) of our weighted ℓ_1 approach compared to other methods for various real-world high-dimensional datasets. The p-values showing the statistical significance of the proposed method compared to the second best model are also reported.

Dataset	**elastic net**	**fused-lasso**	ℓ_∞	ℓ_0	**trace-lasso**	**goscar**	**weighted ℓ_1 (p-value)**
breast cancer	0.734 (0.020)	0.776 (0.025)	0.723 (0.031)	0.717 (0.079)	0.790 (0.052)	0.745 (0.039)	**0.796 (0.066)** (0.007)
atheism vs graphics	0.836 (0.019)	0.820 (0.044)	0.829 (0.024)	0.817 (0.011)	0.847 (0.011)	0.810 (0.016)	**0.955 (0.020)** (7.1e-08)
windows.x vs religion.misc	0.880 (0.023)	0.876 (0.067)	0.874 (0.011)	0.867 (0.014)	0.884 (0.010)	0.870 (0.018)	**0.968 (0.015)** (0.005)
autos vs motorcycles	0.867 (0.007)	0.878 (0.114)	0.869 (0.014)	0.859 (0.010)	0.882 (0.013)	0.841 (0.016)	**0.979 (0.004)** (7.1e-11)
baseball vs hockey	0.872 (0.025)	0.872 (0.056)	0.882 (0.013)	0.877 (0.011)	0.894 (0.023)	0.857 (0.034)	**0.973 (0.012)** (2.42e-08)
forsale vs ms-windows.misc	0.880 (0.017)	0.828 (0.117)	0.877 (0.012)	0.875 (0.018)	0.892 (0.008)	0.854 (0.009)	**0.977 (0.003)** (7.1e-08)
guns vs mideast	0.822 (0.027)	0.805 (0.084)	0.819 (0.009)	0.814 (0.015)	0.830 (0.006)	0.813 (0.014)	**0.945 (0.012)** (3.27e-10)
med vs space	0.863 (0.014)	0.857 (0.097)	0.858 (0.012)	0.852 (0.014)	0.874 (0.004)	0.829 (0.019)	**0.967 (0.005)** (6.35e-08)
pc.hardware vs politics.misc	0.888 (0.024)	0.840 (0.102)	0.868 (0.008)	0.865 (0.014)	0.894 (0.016)	0.867 (0.007)	**0.971 (0.006)** (1.30e-06)
mac.hardware vs christian	0.944 (0.011)	0.862 (0.091)	0.941 (0.015)	0.939 (0.005)	0.951 (0.010)	0.944 (0.013)	**0.973 (0.002)** (0.018)
crypt vs electronics	0.922 (0.012)	0.918 (0.103)	0.919 (0.013)	0.902 (0.016)	0.935 (0.008)	0.849 (0.016)	**0.968 (0.006)** (4.89e-08)



(a) ground truth (Syn-1) (b) elastic net (c) fused-lasso (d) goscar (e) weighted ℓ_1

(f) ground truth (Syn-2) (g) elastic net (h) fused-lasso (i) goscar (j) weighted ℓ_1

(k) ground truth (Syn-3) (l) elastic net (m) fused-lasso (n) goscar (o) weighted ℓ_1

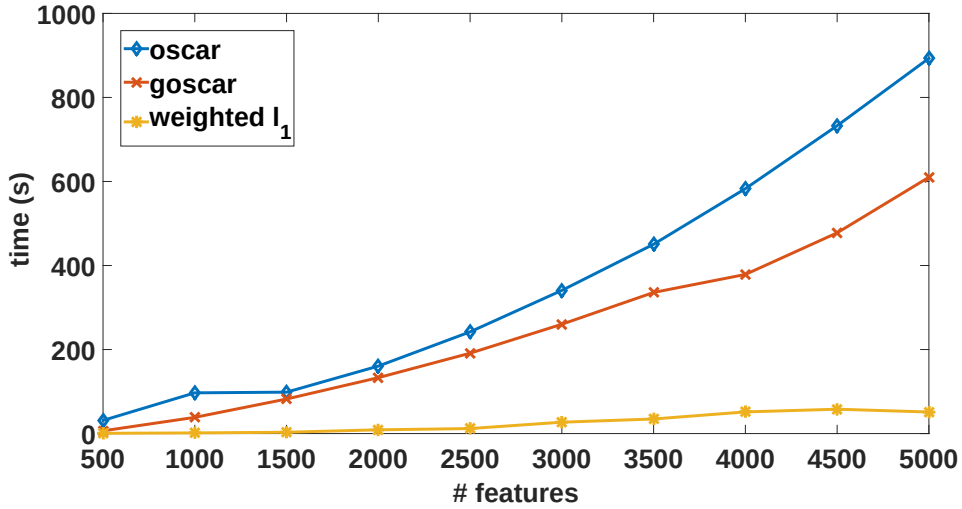
Figure 4.1: Visualizing feature groups obtained on three synthetic datasets by applying four feature grouping algorithms.

Syn-1, *Syn-2* and *Syn-3* datasets, respectively. We can observe that goscar almost retains both the sparsity and the feature grouping structure for *Syn-1* and *Syn-2* datasets, whereas fused lasso and elastic net are not as good as goscar at retaining the grouping structure. Our weighted ℓ_1 approach recovers the ground truth almost completely for *Syn-1* and *Syn-2*. For *Syn-3* one can observe that all competing algorithms perform poorly, but our approach is relatively more effective at recovering the grouping structure, and it avoids misfusing the groups.

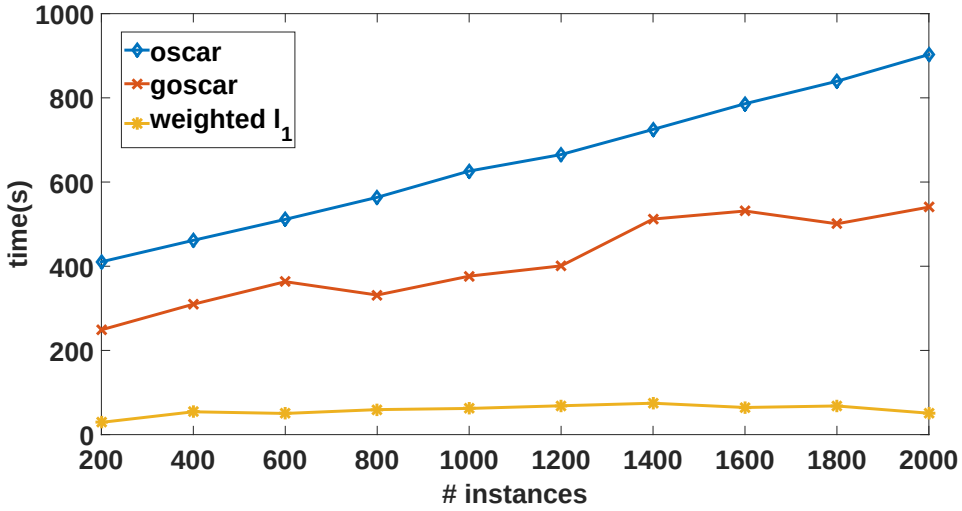
4.6 Scalability Experiments

In this section, we compare the runtime of our weighted ℓ_1 norm regularized linear regression model against oscar and goscar by varying the features and number of instances for a high-dimensional synthetic datasets (*Syn-4*). We choose these two algorithms as baselines for comparison as they are relatively faster compared to other non-convex feature grouping methods used for comparisons in this thesis. This experiment was performed on a machine with 12-GB memory and quad-core CPU.

In Figure 4.2(a), the x-axis represents the number of features and the y-axis represents the time needed for the algorithm execution in seconds. In Figure 4.2(b), the x-axis represents the number of instances and y-axis represents the time. The plots in Figure 4.2 clearly indicate that our algorithm is significantly faster than oscar and goscar. This is because the oscar solver uses a quadratic programming (QP) solver which is slow, and goscar uses an ADMM method based solver, but it requires computing the sparse edgeset graph, which affects its runtime when the number of features are high. In contrast to these algorithms, the FISTA based solver used in our algorithm is much faster because the proximal operator can be computed efficiently with time complexity of $O(plogp)$. In addition, our approach does not explicitly build a feature graph to learn cohesive feature groups, but learns them directly from the data. This also saves the computational time compared to oscar and goscar algorithms.



(a) Scalability w.r.t # features.



(b) Scalability w.r.t # instances.

Figure 4.2: Comparison of runtime (in seconds) for our weighted ℓ_1, oscar and goscar algorithms on *Syn-4* dataset with varying number of features (a) and instances (b).

CHAPTER 5 CONCLUSION AND FUTURE WORK

In this thesis, we presented a weighted ℓ_1 algorithm for solving the misfusion problem while learning regression models from high-dimensional data with inherent feature groupings which are not unknown beforehand. We derived the proximal operator for this weighted ℓ_1 norm and solved the corresponding weighted ℓ_1 norm regularized linear regression problem using the FISTA algorithm. Our approach can automatically learn the feature grouping structure, and it was more effective at resolving the misfusion problem compared to existing methods such as elastic net, fused-lasso and oscar. In addition, our approach was also more scalable compared to oscar and goscar for high-dimensional datasets. We provided exhaustive experimental results on various real-world datasets including the 20-Newsgroups and breast cancer. We also provided results on four synthetic datasets to visually assess the recovery of feature grouping and the scalability of our approach. This work can be extended by developing a more theoretical procedure of providing the optimal weight sequence for the weighted ℓ_1 norm computation.

REFERENCES

[1] Kunal Punera and Joydeep Ghosh. Enhanced hierarchical classification via isotonic smoothing. In *Proceedings of the 17th International Conference on World Wide Web*, pages 151–160. ACM, 2008.

[2] Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 1151–1157. ACM, 2007.

[3] Jiliang Tang, Salem Alelyani, and Huan Liu. Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, page 37, 2014.

[4] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[5] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, 2009.

[6] Chuanren Liu, Fei Wang, Jianying Hu, and Hui Xiong. Temporal phenotyping from longitudinal electronic health records: A graph based framework. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 705–714. ACM, 2015.

[7] Bhanukiran Vinzamuri and Chandan K Reddy. Cox regression with correlation based regularization for electronic health records. In *2013 IEEE 13th International Conference on Data Mining*, pages 757–766. IEEE, 2013.

[8] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[9] Francis R Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems*, pages 118–126, 2010.

[10] Jie Gui, Zhenan Sun, Shuiwang Ji, Dacheng Tao, and Tieniu Tan. Feature selection

based on structured sparsity: A comprehensive study. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–18, 2016.

[11] Sheng Chen and Arindam Banerjee. Structured estimation with atomic norms: General bounds and applications. In *Advances in Neural Information Processing Systems*, pages 2908–2916, 2015.

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

[13] Lei Han and Yu Zhang. Discriminative feature grouping. In *AAAI Conference on Artificial Intelligence*, pages 2631–2637, 2015.

[14] Howard D Bondell and Brian J Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

[15] Sen Yang, Lei Yuan, Ying-Cheng Lai, Xiaotong Shen, Peter Wonka, and Jieping Ye. Feature grouping and selection over an undirected graph. In *Graph Embedding for Pattern Analysis*, pages 27–43. Springer, 2013.

[16] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[17] Seyoung Kim and Eric P Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genet*, 5(8):e1000587, 2009.

[18] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[19] Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Transactions on Signal Processing*, 57(12):4686–4698, 2009.

[20] Yunzhang Zhu, Xiaotong Shen, and Wei Pan. Simultaneous grouping pursuit and feature selection over an undirected graph. *Journal of the American Statistical Association*,

108(502):713–725, 2013.

[21] Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on the complexity of ℓ_p minimization. *Mathematical programming*, 129(2):285–299, 2011.

[22] 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.

[23] Jonathan H Clark. *Locally non-linear learning via feature induction and structured regularization in statistical machine translation*. PhD thesis, Carnegie Mellon University, 2015.

[24] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[25] Edouard Grave, Guillaume R Obozinski, and Francis R Bach. Trace lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems*, pages 2187–2195, 2011.

[26] Neal Parikh and Stephen P Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[27] Nicholas G Polson, James G Scott, and Brandon T Willard. Proximal algorithms in statistics and machine learning. *Statistical Science*, 30(4):559–581, 2015.

[28] Xiangrong Zeng and Mário AT Figueiredo. The ordered weighted ℓ_1 norm: Atomic formulation, projections, and algorithms. *arXiv preprint arXiv:1409.4271*, 2014.

[29] RERE Barlow. Statistical inference under order restrictions; the theory and application of isotonic regression. Technical report, 1972.

[30] Patrick Mair, Kurt Hornik, and Jan de Leeuw. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32(5):1–24, 2009.

[31] Cong Han Lim and Stephen J Wright. Efficient bregman projections onto the permuta-

hedron and related polytopes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1205–1213, 2016.

[32] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[33] Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.

[34] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[35] Mário AT Figueiredo and Robert D Nowak. Ordered weighted ℓ_1 regularized regression with strongly correlated covariates: Theoretical aspects. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 930–938, 2016.

[36] Leon Wenliang Zhong and James T Kwok. Efficient sparse modeling with automatic feature grouping. *IEEE Transactions on Neural Networks and Learning Systems*, 23(9):1436–1447, 2012.

[37] Małgorzata Bogdan, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J Candès. Slope adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9(3):1103–1140, 2015.

[38] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. pROC: an open-source package for R and S+ to analyze and compare roc curves. *BMC bioinformatics*, 12(1):1, 2011.

ABSTRACT

FEATURE GROUPING USING WEIGHTED ℓ_1 NORM FOR HIGH-DIMENSIONAL DATA.

by

KARTHIK KUMAR PADTHE

August 2016

Advisor: Dr. Chandan K. Reddy

Major: Computer Science

Degree: Master of Science

Building effective prediction models from high-dimensional data is an important problem in several domains such as in bioinformatics, healthcare analytics and general regression analysis. Extracting feature groups automatically from such data with several correlated features is necessary, in order to use regularizers such as the group lasso which can exploit this deciphered grouping structure to build effective prediction models. Elastic net, fused-lasso and Octagonal Shrinkage Clustering Algorithm for Regression (oscar) are some of the popular feature grouping methods proposed in the literature which recover both sparsity and feature groups from the data. However, their predictive ability is affected adversely when the regression coefficients of adjacent feature groups are similar, but not exactly equal. This happens as these methods merge such adjacent feature groups erroneously, which is widely known as the misfusion problem. In order to solve this problem, in this thesis, we propose a weighted ℓ_1 norm-based approach which is effective at recovering feature groups, despite the proximity of the coefficients of adjacent feature groups, building extremely accurate prediction models. This convex optimization problem is solved using the fast iterative soft-thresholding algorithm (FISTA). We depict how our approach is more successful than competing feature grouping methods such as the elastic net, fused-lasso and oscar at solving the misfusion problem on synthetic datasets. We also compare the goodness of prediction

of our algorithm against state-of-the-art non-convex feature grouping methods when applied on a real-world breast cancer dataset, the 20-Newsgroups dataset and synthetic datasets.

AUTOBIOGRAPHICAL STATEMENT

Karthik Kumar Padthe was born in Nizamabad, India on April 19, 1991. He completed B.Tech in Computer Science from Jawaharlal Nehru Technological University, Hyderabad (JNTU-H) in 2012. Then he worked as a Software Engineer in product based Internet of Things (IOT) startup M2M innovations LLP located in Bengaluru, India. He joined Wayne State University in Winter 2015 to pursue Masters in Computer Science. His research interests include data mining, machine learning and health analytics.