2016 IEEE 16th International Conference on Data Mining

Feature Grouping using Weighted ℓ_1 Norm for High-Dimensional Data

Bhanukiran Vinzamuri*, Karthik K. Padthe*, and Chandan K. Reddy†
*Dept. of Computer Science, Wayne State University, Detroit, MI- 48202
Email: {bhanukiranv, karthikp}@wayne.edu
†Dept. of Computer Science, Virginia Tech, Arlington, VA-22203
Email: reddy@cs.vt.edu

***Abstract*—Building effective predictive 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 also called the misfusion problem. In order to solve this problem, in this paper, 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 predictive models. This convex optimization problem is solved using the fast iterative soft-thresholding algorithm (FISTA). We depict how our approach is more effective at resolving the misfusion problem on synthetic datasets compared to existing feature grouping methods such as the elastic net, fused-lasso and oscar. We also evaluate the goodness of the model on real-world breast cancer gene expression and the 20-Newsgroups datasets.**

***Keywords-regression; regularization; feature grouping; high-dimensional data*.**

I. Introduction

Extracting feature groups from high-dimensional data is an extremely important problem in several domains such as bioinformatics, healthcare analytics and general regression analysis. Real-world datasets from these domains have an inbuilt feature grouping structure which is difficult to decipher apriori. Groups of features can be interpreted as clusters where features within each cluster (group) are highly correlated and differ significantly from the features in other groups. However, this task is conceptually different from clustering the features or co-clustering, as these methods are primarily used in an unsupervised setting, whereas feature grouping is done in a supervised setting such as classification or regression.

One of the advantages of developing accurate feature grouping algorithms 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 [1]–[5]. It is also desirable for regression models built on high-dimensional data to recover cohesive and homogenous feature groups with good accuracy, as this reduces the error variance of the model and increases its generalizability.

Existing regularization methods such as lasso are not capable of performing feature grouping. Elastic net, fused-lasso and oscar are popular methods which perform feature grouping, but the elastic net does not promote equality of coefficients among the features in each group [6]. The fused-lasso [7] is not capable of grouping positive and negative variables together even if they share similar magnitude of regression coefficients, and oscar [8] solves a quadratic programming (QP) problem and it's computationally expensive to compute. More importantly, these feature grouping methods are not capable of solving the misfusion problem which is explained below.

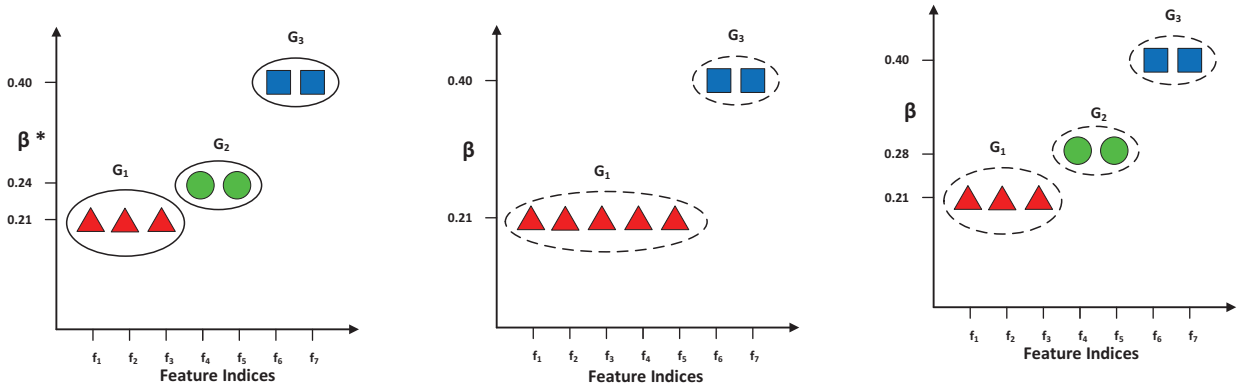
A. The Misfusion Problem

In this section, we present an illustration of the *misfusion* problem on a small synthetic dataset. In Figure 1, we present a scenario of how feature grouping algorithms such as oscar are unable to resolve the misfusion problem [9]. 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 Figure 1. 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=$X\beta^* + \epsilon$ 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 oscar regression model on this dataset and we plot the learned regression coefficient values (β) on the Y-axis in Figure 1(b).

One can clearly observe from Figure 1(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 paper effectively resolves the misfusion problem as can be seen in Figure 1(c), with a minor trade-off being the complete recovery of the ground truth. This misfusion problem

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.83

IEEE computer society



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

Fig. 1: A simple illustration demonstrating the misfusion problem and the results obtained by applying existing methods and our approach.

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 predictive models.

B. Our Contributions

The major contributions of this paper 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 the 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 on high-dimensional real-world datasets, namely, the 20-Newsgroups and breast cancer gene-expression datasets. We also evaluate our approach on three synthetic datasets and visualize the feature groups obtained.

This paper is organized as follows. In Section II, we describe the related work on feature grouping algorithms. In Section III, we present the preliminaries needed to comprehend our approach. In Section IV, we present our proposed weighted ℓ_1 approach by explaining the formulation of the proximal operator and the corresponding algorithm. In Section V, we conduct experiments to evaluate the performance of our approach compared to baseline models on the 20-Newsgroups, breast cancer gene-expression and synthetic datasets.

II. Related Work

In this section, we briefly review existing methods for supervised feature grouping. The elastic net [6] which uses a convex combination of the ℓ_1 and ℓ_2 norms groups correlated features together. However, the regression coefficients of features within a group are not equal which leads to the misfusion problem explained earlier. The kernel elastic net [12] is an extension of the elastic net which can capture feature correlation more effectively using a kernel matrix. It was proven to outperform the elastic net for highly correlated data but it does not address the misfusion problem.

The fused-lasso [7] uses a combination of the ℓ_1 norm and a smoothness term which is used to capture the difference among the regression coefficients of adjacent features. Penalizing this difference promotes equality of coefficients among features which helps to capture feature groups. In this manner, the fused-lasso improves over the elastic net by promoting feature coefficient equality within a group. However, it assumes that such a temporal ordering exists among adjacent features in the real-world data which need not always be observed.

Oscar [8] improves over both the fused-lasso and the elastic net by capturing homogeneous groups and it does not assume any temporal ordering among features. However, the quadratic programming-based solver employed in oscar is not scalable. The alternate direction method of multipliers (ADMM) [11] has been used to accelerate the graph-based oscar regression [10], but this modified approach requires the feature graph to be provided apriori which need not be known in advance for most datasets.

In contrast to the aforementioned methods, the main goal of our weighted ℓ_1 norm-based formulation is to obtain groups of features efficiently by directly resolving the misfusion problem. This is also different from the weighted ℓ_1 norm proposed in [13], where the focus is on learning sparsity efficiently with fewer examples and not feature grouping. In addition, in this approach the weights are optimized over several iterations, whereas our approach uses a fixed set of weights which satisfy a pre-specified ordering scheme which is explained in the next section.

III. Preliminaries

In this section, we present the preliminaries needed to comprehend our weighted ℓ_1 norm-based algorithm for feature

TABLE I: Notations used in this paper.

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.
$\|x\|_\downarrow$	non-increasing sorted $\|x\|$.
$P(x)$	permutation matrix.
$\Omega(\beta)$	weighted ℓ_1 norm.
w	$\mathbb{R}^p$ weight vector.
K_m^+	monotone non-negative cone.

grouping. Table I presents important terms and notations used in this paper. We now explain the interpretation of each of these notations in detail. Lower case 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, respectively. Matrices are written in upper case (such as X) and the i^{th} column vector of X is represented using 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 using $x_{[i]}$. 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 $|x|_{[1]} \geq |x|_{[2]} \geq \ldots \geq |x|_{[p]}$ and 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. 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 norm ball is octagonal. The oscar regularizer is defined as in Eq. (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.

$$h(\beta) = \lambda_1 \parallel \beta \parallel_1 + \lambda_2 \sum_{i<j} max\{|\beta_i|, |\beta_j|\} \tag{1}$$

We now define the weighted ℓ_1 norm and the regularized linear regression problem in Eq. (2).

$$\begin{aligned} &\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{2} \parallel y - X\beta \parallel_2^2 + \Omega(\beta) \\ &\Omega(\beta) = \parallel w \odot |\beta|_\downarrow \parallel_1 \end{aligned} \tag{2}$$

In this equation, w is a weight vector of non-increasing weights, which is defined as $w = \{w_1 \geq w_2 \geq \ldots \geq w_p \geq 0\}$ 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 [14]. 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. In the next section, we formulate the proximal operator [15] for the weighted ℓ_1 norm and use it within an accelerated proximal gradient (APG) algorithm for solving this problem efficiently.

IV. The Proposed Method

In this section, we present an accelerated proximal gradient FISTA-based 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. We also discuss the complexity of our approach.

A. Proximal operator for Weighted ℓ_1 Norm.

The proximal operator for Ω, which is denoted by $\mathrm{prox}_\Omega(.)$ is defined in Eq. (3) for any $v \in \mathbb{R}^p$ using the standard definition of a proximal operator proposed in [15]. We will now simplify the proximal operator using the steps provided below and explain the procedure for obtaining it.

$$\mathrm{prox}_\Omega(v) = \arg\min_{\beta\in\mathbb{R}^p} \left(\frac{1}{2} \parallel \beta - v \parallel^2 + \Omega(\beta) \right) \tag{3}$$

In Eq. (3), we must estimate $\mathrm{prox}_\Omega(v)$ in order to employ it within the FISTA framework. We use the fact that $w, \beta \in \mathrm{K}_m^+ \subset \mathbb{R}^p$ and mention the steps needed to simplify Eq. (3) further as follows

$$\begin{aligned} \mathrm{prox}_\Omega(v) &= \arg\min_{\beta\in\mathrm{K}_m^+} \frac{1}{2} \parallel \beta - v \parallel_2^2 + w^T\beta \\ &= \arg\min_{\beta\in\mathrm{K}_m^+} \frac{1}{2} \parallel \beta - (v - w) \parallel_2^2 \\ &\text{s.t} \quad \beta_1 \geq \beta_2 \geq \ldots \geq \beta_p \geq 0 \end{aligned} \tag{4}$$

The simplification yields Eq. (4) which needs to be solved to obtain $\mathrm{prox}_\Omega(v)$. 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. (5), and (ii) applying a subsequent projection of this result onto $\mathbb{R}^{p+}$ by clipping the negative values.

$$\begin{aligned} &\arg\min_{\beta\in\mathrm{K}_m} \frac{1}{2} \parallel \beta - (v - w) \parallel_2^2 \\ &\text{s.t} \quad \beta_1 \geq \beta_2 \geq \ldots \geq \beta_p \end{aligned} \tag{5}$$

This projection problem in Eq. (5) has the form as given in Eq. (6) which is also called the isotonic regression problem which is a submodular convex optimization problem [16]. To solve Eq. (5), we use an existing isotonic regression solver

such as the pool adjacent violators algorithm (PAVA).

$$\arg\min_{y\in\mathbb{R}^p} \sum_{i=1}^{p} f_i(y_i) \quad \text{s.t} \quad y_1 \leq y_2 \leq \ldots \leq y_p \tag{6}$$

PAVA [17] is one of the most efficient methods for solving the isotonic regression problem with $O(p\ \log p)$ time complexity. By applying this PAVA algorithm to solve Eq. (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 1, which is the proposed weighted ℓ_1 norm regularized linear regression solver.

B. 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) [18]. FISTA is a variant of the iterative soft-thresholding algorithm (ISTA) which uses the accelerated proximal gradient (APG) method based on Nesterov's technique [19]. First-order optimization methods such as FISTA converge at a rate of $\text{O}(\frac{1}{n^2})$ compared to traditional gradient methods which have a slow convergence rate of $\text{O}(\frac{1}{\sqrt{n}})$.

In Algorithm 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. (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, prox_Ω is computed by solving Eq. (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 final converged regression coefficient vector is returned.

Algorithm 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 $L = 2\Lambda_{max}(X^TX)$, Weight vector w, Tolerance parameter tol, max iterations max_iter.

Output: Regression coefficients $\beta \in \mathbb{R}^p$

1 **Initialize**: $\beta_0 \in \mathbb{R}^p, u_1 = \beta_0, t_1 = 1$;
2 **for** *k=1 to max_iter* **do**
3 $\quad \beta_k = \text{prox}_\Omega\Big(u_k - X^T(Xu_k - y)/L\Big)$ using Eq. (4) ;
4 $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;
5 $\quad u_{k+1} = \beta_k + \Big(\frac{t_k - 1}{t_{k+1}}\Big)(\beta_k - \beta_{k-1})$;
6 $\quad$ **if** $\| \beta_k - \beta_{k-1} \|_2 < tol$ **then**
7 $\quad\quad$ break;
8 $\quad$ **end**
9 **end**
10 Return β_k ;

C. Complexity Analysis

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. (5) which has a time complexity of $O(p\ \log p)$ 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 $O\Big(\frac{1}{\sqrt{\epsilon}}(p(n + log p)\Big)$. We observe that for most of the real-world datasets $n \gg \log p$, so the complexity of this algorithm is $O(np/\sqrt{\epsilon})$.

V. Experimental Results

In this section, we present the experiments conducted to evaluate the performance of our weighted ℓ_1 approach. We explain the details pertaining to the synthetic dataset creation and also describe the real-world datasets used. We also explain the implementation details for these methods. We conduct different experiments to assess the goodness of prediction and recovery of feature groups using the proposed approach.

A. Datasets Description

1) Synthetic datasets: We created three synthetic datasets with moderate dimensionality which are *Syn-1*, *Syn-2* and *Syn-3*. We include a feature grouping pattern in these datasets which is specified below. This allows to visualize the goodness of feature grouping methods for moderate dimensionality datasets. The response variable in these datasets is created using the linear regression model which can be written as y=$X\beta^*$+ϵ 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) *Syn-1*: σ=3, c_{ij}=$0.7^{|i-j|}$, p=8, β^*=[3,2,1.5,0,0,0,0,0]^T.
2) *Syn-2*: σ=3, c_{ij}=$0.7^{|i-j|}$, p=8, β^*=[3,0,0,1.5,0,0,0,2]^T.
3) *Syn-3*: σ=15, c_{ij}=0.5 when $i \neq j$, and 1 otherwise, p=40 and β^*=$[\underbrace{0,\cdots,0}_{10},\underbrace{2,\cdots,2}_{10},\underbrace{0,\cdots,0}_{10},\underbrace{2,\ldots,2}_{10}]^T$.

2) 20-Newsgroups dataset: This dataset is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups[1]. We extracted 5 pairs from the 20 different newsgroups to form 5 datasets as given in Table II. In this table, we use short acronyms to represent the names of the datasets concisely. We treat each of these 5 pairs as a binary classification problem, where in we label each document in the dataset with the newsgroup it belongs to. As a part of the preprocessing step, we do stemming to

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

TABLE II: Description of the datasets used in our experiments.

Dataset	# Features	# Instances
Syn-1	8	280
Syn-2	8	280
Syn-3	40	800
breast-cancer	8141	295
ath vs gra	7943	2000
win vs rel	8442	2000
auto vs moto	7094	2000
bb vs hoc	7909	2000
fs vs msw	6678	2000

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.

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 performance of models used here. This helps to obtain unbiased results.

B. Performance evaluation

We use the Area Under ROC Curve (AUC) to compare the performance of the proposed model with the baseline models. 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*) [20] was used to implement algorithms such as the elastic net and fused-lasso. To calculate AUC we use the R package *pROC*. The AUC and standard deviation (std) are obtained using five-fold cross validation. Parameter tuning of the regularization parameters was done using a hold-out set for all the baseline 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 [21]. All codes used for running the baseline models and our weighted ℓ_1 algorithm are available at this link to ensure reproducibility of our work[3].

C. Goodness of Prediction

In Table III we provide the AUC (along with the standard deviation) for six real-world binary classification tasks. We obtain the binary classifier output from the regression-based models by computing the sign of the predicted response variable. We observe that for all the cases our weighted ℓ_1 approach does better compared to the remaining four models. This proves the effectiveness of our approach for real-world classification problems.

TABLE III: AUC (std) of our weighted ℓ_1 approach compared to other methods for various real-world high-dimensional datasets.

Dataset	elastic net	fused-lasso	oscar	weighted ℓ_1
breast-cancer	0.734 (0.020)	0.776 (0.025)	0.745 (0.039)	**0.796 (0.066)**
ath vs gra	0.836 (0.019)	0.820 (0.044)	0.810 (0.016)	**0.955 (0.020)**
win vs rel	0.880 (0.023)	0.876 (0.067)	0.870 (0.018)	**0.968 (0.015)**
auto vs moto	0.867 (0.007)	0.878 (0.114)	0.841 (0.016)	**0.979 (0.004)**
bb vs hoc	0.872 (0.025)	0.872 (0.056)	0.857 (0.034)	**0.973 (0.012)**
fs vs msw	0.880 (0.017)	0.828 (0.117)	0.854 (0.009)	**0.977 (0.003)**

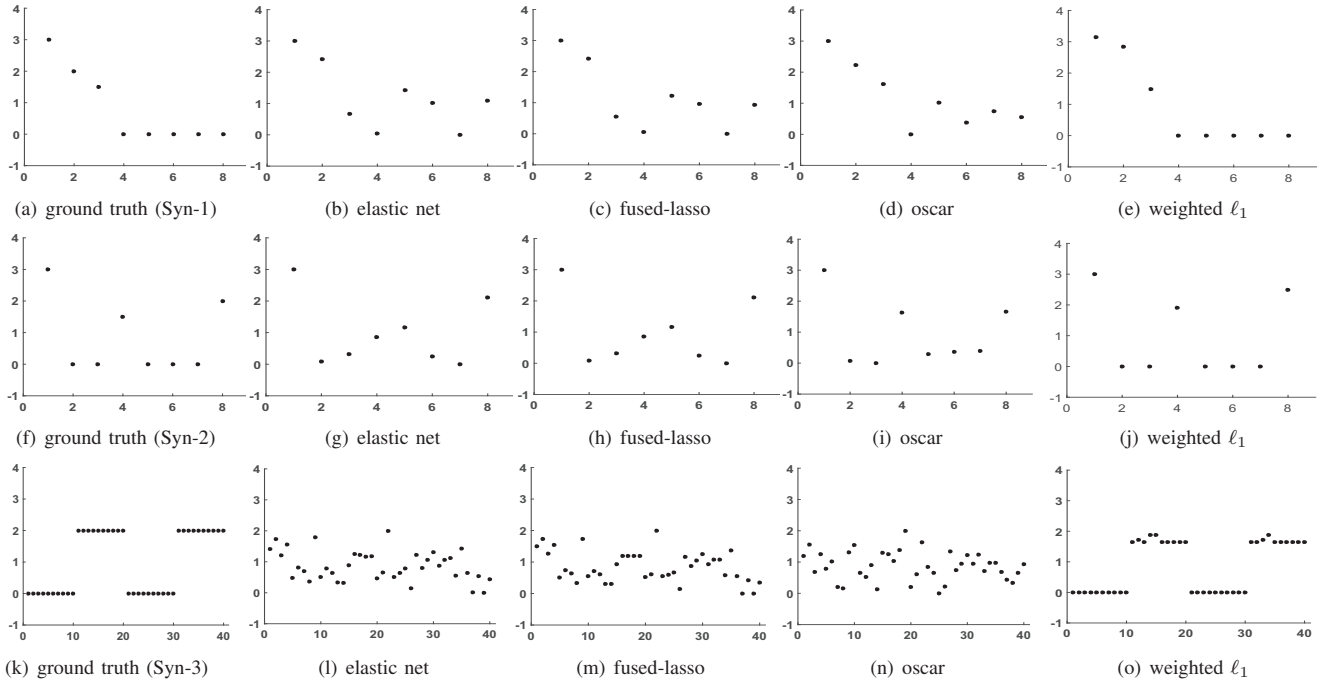
D. Recovering Feature Groups

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 2 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 2 corresponds to *Syn-1*, *Syn-2* and *Syn-3* datasets, respectively. We can observe that oscar infers the feature grouping structure for *Syn-1* and *Syn-2* datasets upto some extent, whereas the fused-lasso and elastic net are not effective at inferring the grouping structure. Our weighted ℓ_1 approach recovers the ground truth almost completely for *Syn-1* and *Syn-2*. For *Syn-3* dataset one can observe that all competing algorithms perform poorly, but our approach is relatively more effective at recovering the grouping structure, and it successfully avoids misfusing the groups which can be seen clearly.

VI. Conclusion

In this paper, we presented a weighted ℓ_1 algorithm for solving the misfusion problem while learning regression models from high-dimensional data with inherent feature groups which are not known apriori. We formulated 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. We conducted experiments on the 20-Newsgroups and breast-cancer gene-expression high-dimensional datasets to assess the goodness

[2]http://lbbe.univ-lyon1.fr/-Jacob-Laurent-.html?lang=fr
[3]https://github.com/Karthikpadthe/ICDM-2016



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

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

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

Fig. 2: Visualizing feature groups obtained on three synthetic datasets by applying four feature grouping algorithms.

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.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation grants IIS-1231742, IIS-1527827 and IIS-1646881.

REFERENCES

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

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

[3] L. Jacob, G. Obozinski, and J. P. Vert. Group lasso with overlap and graph lasso. *Proceedings of the 26th annual international conference on machine learning*, pages 433–440, 2009.

[4] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan. Feature Selection Based on Structured Sparsity: A Comprehensive Study. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–18, 2016.

[5] S. Chen and A. Banerjee. Structured Estimation with Atomic Norms: General Bounds and Applications. *Advances in Neural Information Processing Systems (NIPS)*, 28, pages 2908–2916, 2015.

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

[7] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1), pages 91–108, 2005.

[8] H. D. Bondell, and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1), pages 115–123, 2008.

[9] L. Han, and Y. Zhang. Discriminative Feature Grouping. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2631–2637, 2015.

[10] S. Yang, L. Yuan, Y. C. Lai, X. Shen, P. Wonka, and J. Ye. Feature grouping and selection over an undirected graph. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 922–930. ACM, 2012.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1) , pages 1–122, 2011.

[12] B. Vinzamuri and C. K. Reddy. Cox regression with correlation based regularization for electronic health records. *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 757–767, 2013.

[13] E. Candes, M. Wakin and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 14 (5), pages 877–905, 2008.

[14] X. Zeng, and M. Figueiredo. The Ordered Weighted ℓ_1 Norm: Atomic Formulation, Projections, and Algorithms. *arXiv preprint arXiv:1409.4271*, 67(1), 2014.

[15] N. Parikh and S. Boyd. Proximal Algorithms. *Foundations and Trends in optimization*, 1(3), pages 127–239, 2014.

[16] R. Barlow, D. Bartholomew, J. Bremner and D. Brunk. Statistical inference under order restrictions: The theory and application of isotonic regression. Publisher Wiley New York, 1972.

[17] P. Mair, K. Hornik and J. D. Leeuw. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of statistical software*, 32(5), pages 1–24, 2009.

[18] A. Beck, and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1), pages 183–202, 2009.

[19] Y. Nesterov. Gradient methods for minimizing composite objective function. *UCL - CORE - Center for Operations Research and Econometrics*, 2007.

[20] J. Mairal, F. R. Bach and J. Ponce. Sparse Modeling for Image and Vision Processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2), pages 85–283. 2014.

[21] M. Bogdan and E. Berg and C. Sabatti and W. Su and E. Candes. SLOPE adaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3), pages 1103-1140, 2015.