A Sparse-Group LASSO

University of Toronto Mississauga

Advanced Statistical Learning

Saihiel Bakshi
saihiel.bakshi@mail.utoronto.ca
1003700989

April 20, 2020

Contents

1. Abstract

Modern statistical learning commonly deals with problems in high dimensional setting, specifically when $p >> n$. In such settings, not all the parameters present in a model are equally informative. Additionally, selecting a subset of informative parameters reduces the computational time, while increasing the model's interpretability. Variable selection through shrinkage methods is a popular approach for this. However, for high dimensional supervised learning problems, often using problem specific assumptions leads to a greater accuracy [3]. Previously proposed models are unable to exploit such assumptions directly. Particularly, for problems with grouped covariates, which are believed to have sparsity at both the group and within group level, no prior methods yield sparsity at both levels. While, the Group Lasso proposed the use of an l_2 regularized penalty, and is able to generate sparse solutions at the group level, it is unable to do so at the within-group level [6]. To solve this, Friedman et al [3] proposed a regularized model for linear regression with l_1 and l_2 penalties. They demonstrated that the optimal fit for this model yields the desired effect of group-wise and within group sparsity. Additionally, Friedman et all [3] proposed an algorithm to fit the model via accelerated generalized gradient descent. In this paper, we summarize the Sparse-Group Lasso method, analyse the proposed algorithm, and demonstrate the efficacy of the methodology on simulated data.

Keywords: variable selection, regularize, regression, monte carlo, simulation, nesterov, lasso

---

2. Introduction

In supervised learning problems a statistical learning model attempts to learn the relationship between the observed predictors and response variables. However, in high dimensional supervised learning problems not all the predictors are equally as powerful in predicting the response variables. Consequently, modern statistical learning has concerned itself with developing efficient methodologies for variable selection. Variable selection is the process of selecting subsets of predictors with the highest predictive power. Today, high dimensional data is prolific and can be seen in many areas including genomics. When the number of predictors, p, is substantially larger than n, the computational cost and estimation accuracy become, equally, the priority for any statistical methods.

2.1 Background

In this report we will consider the supervised regression framework. Our data consists of an n response vector y, and an n by p matrix of features, X, where $p >> n$. Additionally, models used are fit without an intercept and the features and standardized before shrinkage methods are applied. The design

matrix, X is divided into m different groupings. A common example of this is when the data represents multiple factor levels of categorical data. We denote the submatrix of X for group l as $X^{(l)}$, such that it contains the columns corresponding to predictors in that grouping. The objective is to estimate the coefficient parameters β. Within β, $\beta^{(l)}$ represents the coefficient vector of group l, with p_l being its length.

Since our design matrix is in dimensions such that $p >> n$, our objective is to find the subset of the design matrix with the highest predictive power. This implies that our coefficient vectors, will be sparse. By setting a value in β to be exactly 0 the effect of the corresponding predictor in X is removed from the model.

2.2 Regularization and Sparsity

Variable selection can be performed through various methods. Two predominant methods includes automatic subset selection and shrinkage. While automatic subset selection methods can be easier to implement in practice, they tend to have higher computational costs and are not guaranteed to converge to the optimum solution in most cases. Additionally, shrinkage methods are continuous processes, more stable, and have lower variability [4].

Regularization methods control model complexity by shrinking coefficient estimates. The model is penalized for have large coefficients and hence shrinks the size of the coefficients close to zero. Particularly, sparsity regularization selects the input variables that best describe the response by setting coefficients to exactly zero. In high-dimensional learning exploiting problem specific assumptions can lead to higher accuracy and hence modern statistical learning methods employ techniques such as structured sparsity regularizers. Structured sparsity regularization uses prior assumptions of predictors, such as groupings, to select optimal parameters in the dataset.

3. Motivation

3.1 LASSO

In high dimensional settings ($p >> n$) the usual linear regression framework fails. To combat this in 1996 Tibshirani [4] proposed a new method by adding regularization and bounding the solution to this problem by the l_1 norm. This approach is famously known as Lasso, and the problem attempts to minimize:

$$\frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

Lasso finds a solution with a few non-zero entries in β [4]. But since our problem contains data with natural groupings, such as gene expression data, or factor level indicators in categorical data. We want to exploit this property to find sparsity at the group level. This is a situation the regular Lasso method does not consider.

3.2 Group Lasso

In an attempt to create sparsity at a group level, Yuan and Lin (2007) [6] proposed the group lasso criterion. The problem is of solving the following convex minimization:

$$\min_\beta \frac{1}{2}\left\| y - \sum_{l=1}^{m} X^{(l)}\beta^{(l)} \right\|_2^2 + \lambda \sum_{l=1}^{m} \sqrt{p_l}\left\|\beta^{(l)}\right\|_2$$

This criteria exploits the non-differentiability of $\left\|\beta^{(l)}\right\|_2$ at $\beta^{(l)} = 0$, thus setting groups of coefficients to exactly zero [6]. Akin to Lasso the tuning parameter λ controls the sparsity of the solution. Hence, when the size of each group is exactly one, the solution yields the same values as Lasso.

While the group Lasso gives a sparse set of groups, if it includes a group in the model then all coefficients in the group will be nonzero. Similarly if a group in not included in the model, then all coefficients will be zero.

There are certain problems that even this improved method contains:

1. The method does not yield sparsity within a group.
2. If a group of parameters is non-zero, they will all be non-zero.
3. Yuan & Lin's algorithm assumed that submatrices in each group are always orthonormal [6].
4. If the predictors are not orthonormal, one approach is to othonormalize them before applying group lasso:
   - Generally, this will not provide a solution to the original problem [2].
5. On the other hand, if predictors are orthonormal, then convergence is not guaranteed for their proposed method.

Consequently, in order to combat the aforementioned problems and to gain both sparsity of groups and within each group, the method of Sparse-Group Lasso was proposed.

4. A Sparse Group Lasso

Sparse-Group Lasso gives sparse solutions at both a group level ("groupwise sparsity") and within groups ("within group sparsity"). Friedman et al (2012) [3] proposed the criteria:

$$\min_\beta \frac{1}{2n}\left\| y - \sum_{l=1}^{m} X^{(l)}\beta^{(l)} \right\|_2^2 + (1-\alpha)\lambda \sum_{l=1}^{m} \sqrt{p_l}\left\|\beta^{(l)}\right\|_2 + \alpha\lambda\left\|\beta^{(l)}\right\|_1$$

where $\alpha \in [0, 1]$

4.1 Criterion

The optimization problem proposed by Friedman et al [3] is a convex combination of the lasso and group lasso penalties and hence is convex itself. The method introduces a new tuning parameter α, where

$\alpha = 0$ gives the group lasso fit and $\alpha = 1$ gives the standard lasso fit. The method draws similarities to the elastic net penalty but differs as the $\left\|\beta^{(l)}\right\|_2$ penalty is not differentiable at $\mathbf{0}$ [3]. This allows the possibility of *some* groups to be completely zeroed out. Yet, we also see that within non-zero groups the fit *is the same* as elastic net.
The optimal solution to the proposed criterion is characterized by the subgradient equations. For a group k, $\hat{\beta}^{(k)}$ satisfies

$$\frac{1}{n}X^{(k)T}\left(y - \sum_{l=1}^{m} X^{(l)}\hat{\beta}^{(l)}\right) = (1-\alpha)\lambda\mu + \alpha\lambda v$$

where μ and v are subgradients of $\|\hat{\beta}^{(k)}\|_2$ and $\|\hat{\beta}^{(k)}\|_1$ respectively, evaluated at $\hat{\beta}^{(k)}$. Then, it can be shown that the subgradient equations can be satisfied with $\hat{\beta}^{(k)} = 0$ if

$$\|S(X^{(k)T}r(-k)/n, \alpha\lambda)\|_2 \leq (1-\alpha)\lambda$$

where $r(-k)$ is the partial residual of y, subtracted from all other group fits except group k; and $S(\cdot)$ is the coordinate-wise soft thresholding operator [3].

4.2 Methodology

Friedman et al. proposed *pathwise coordinate gradient descent*, using accelerated generalized descent with backtracking within each group. Since, the criteria is the sum of a convex differential function (the loss), and a separable penalty (between groups) the advantage of this method is that Blockwise Coordinate Gradient Descent is guaranteed to converge to the global optimum.

Additionally, instead of fixing the regularization parameter the authors proposed a method of finding a pathwise solution for various λ values, and by implementing warm-starts along each pathwise iteration the method is made efficient. This helps solve the problem of selecting optimal tuning parameters.

The important techniques employed in the generalized gradient descent procedure to aid in faster convergence are Nesterov's Momentum and Backtracking.

Nesterov's momentum smooths updates by taking a step in the direction of the previous accumulated gradient then corrects the velocity based on this step. The previous gradients are accumulated using an exponential weighted moving average of the previous gradients [1]. This procedure reduces the stochasticity in the gradient update steps by dampening the updates when the optimization space is narrow (non-optimal) and increasing the magnitude of the updates when the conditions are optimal [1]. Backtracking is a method for finding the optimal step size. This optimization method maximizes the step size in the direction of steepest descent [5]. Together these methods help the algorithm converge the optima faster.

4.2.1 Algorithm

The algorithm consists of two loops, one over each of the groups and another over the parameters within each group. Blockwise-descent is used over the groups, and to solve within each group accelerated generalized gradient descent is employed. [3]

The algorithm is outlined as follows:

1. **[Outer loop]** Cyclically iterate over the groups; at each group k to minimize over, consider the other group coefficients as fixed.
2. Check if the group's coefficients are exactly 0. If not, enter inner loop:
3. **[Inner loop]** Start with $\beta^{(k,l)} = \theta^{(k,l)} = \beta_0^{(k)}$, step size $t = 1$, and counter $l = 1$. Unit convergence repeat:
   (a) Update the gradient g by $g = \nabla l(r_{(-k)}, \beta^{(k,l)})$
   (b) Optimize step size by iterating $t = 0.8 * t$ until
   $$l(U(\beta^{(k,l)}, t)) \leq l(\beta^{(k,l)}) + g^T \Delta_{(l,t)} + \frac{1}{2t}\|\Delta_{(l,t)}\|_2^2$$
   (c) Update $\theta^{(k,l)}$ by:
   $$\theta^{(k,l+1)} \leftarrow U(\beta^{(k,l)}, t)$$
   (d) Update the center via a Nesterov step by
   $$\beta^{(k,l+1)} \leftarrow \theta^{(k,l)} + \frac{l}{l+3}(\theta^{(k,l+1)} - \theta^{(k,l)})$$
   (e) Set $l = l + 1$

   Note: $U(\beta_0, t)$ is the update rule, $\Delta_{(l,t)} = U(\beta^{(k,l)}, t) - \beta^{(k,l)}$

In the above algorithm the outer loop is optimizing over the groups using (block) coordinate-wise gradient descent, while the inner loop uses generalised gradient descent within each of the non-zero groupings. Additionally, we see at steps (b) backtracking being implemented, and at step (d) Nesterov's Momentum being applied.

4.2.2 Pathwise Solutions

The algorithm also includes an additional step as mentioned before, it finds the pathwise solutions for a range of λ values. Since, iteratively fitting models over a grid of α and λ values is computationally impractical [3]. The algorithm instead fixes the mixing parameter α, and computes solutions for a path of λ values using warm starts.

The procedure is as follows:

1. Start with large values of λ to set $\hat{\beta} = 0$ and decrease λ from there
2. Use the previous solution for the algorithm at the next λ value along the path. [*Warm-starts*]
3. Since α is fixed, the objective is a piecewise quadratic in λ
4. Find the smallest λ_l for each group that sets that group's coefficients to 0
   - Thus begin the path search with:
   $$\lambda^{\max} = \max_i \lambda_i$$
   - The exact value at which the first coefficient enters the model.
   - Set $\lambda^{\min}$ to be a small fraction of $\lambda^{\max}$ [default 0.1]

This method is efficient for finding optimal λ values for solution, however, optimal α value is still problem specific.

4.3 Analysis

The model proposed by the authors [3] has multiple benefits that make it better than previous methods. Particularly, sparse-group Lasso enables sparsity at both group level as well as predictor level, and improves interpretability beyond existing methods such as lasso and group-lasso.

Furthermore, the methodology proposed by the authors is considered significant because it guarantees convergence to global optima, enables optimal tuning of regularization hyperparameter efficiently and can even be used to fit group-lasso with non-orthonormal predictors. These substantial improvements from previous methods are why this new method is so highly accredited.

The sparse-group Lasso method is particularly useful in high-dimensional situations where we have data with many predictors and large number of levels per predictor. Then, it is likely that even for informative predictors, many of the levels may not be informative. Thus, spare-group Lasso takes this into consideration, setting coefficients for many levels to 0, even in nonzero groups. This yields the desired sparsity effect at both a group-level and within-group level.

5. Numerical Analysis & Simulations

The simulations conducted by the authors were repeated and are represented below. The authors conducted simulations to test the accuracy of sparse-group lasso as a variable selection. The sparse-group lasso and lasso methods were compared. In order to follow along a similar procedure as that of the author, the simulated covariate matrix X was constructed as follows.

The columns of X are constructed as iid Gaussian, and the response is constructed as:

$$y = \sum_{l=1}^{g} X^{(l)}\beta^{(l)} + \sigma\epsilon$$

where $\epsilon \sim (0, I), \beta^{(l)} = (1, 2, \ldots, 5, 0, \ldots, 0)$ for $l = 1, \ldots, g$, and σ was set to so the signal to noise ratio was 2. Similar to the simulations conducted by the authors, for both Lasso and Sparse-group Lasso λ was set so the number of nonzero coefficients chosen in the fit matched the true number of nonzero coefficients in the generative model. For sparse-group Lasso α was set to 0.95. Additionally, the number of generative groups, g was varied from 1 to 3 changing the amount of sparsity. However, instead of conducting 10 trial of repeated simulations, only a single trial of simulations was conducted for each parameter settings.

For each of the three generative groups, the parameter settings were as follows:
Note: n is number of observations, p is the number of covariates and m is number of groups

1. Simulation 1: $n = 150, m = 100, p = 5000$
2. Simulation 2: $n = 200, m = 400, p = 40000$

5.1 Results

Table 5.1: Results of simulation

	Number of Groups in Generative Model		
	1 group	2 groups	3 groups
	$n = 150, m = 100, p = 5000$		
SGL	1	0.9	0.69
Lasso	0.8	0.7	0.6
	$n = 200, m = 400, p = 40000$		
SGL	0.8	0.8	0.67
Lasso	0.8	0.6	0.47

The simulation results show that as the number of groups increases Sparse-group Lasso's proportions of correct nonzero coefficient identifications is higher than that of Lasso. The same can be seen when the number of generative groups increases. Additionally, in simulation 2, Sparse-group Lasso's proportion of correct identifications remains constant as the number of generative groups increases from 1 to 2, but Lasso's proportion decreases by 0.2. This goes to show the power of sparse-group Lasso when the data contains natural groupings.

6. Conclusions

The Sparse-Group Lasso method is an effective technique for variable selection when the data contains natural groupings. The method is preferred over previously proposed methods because it yields sparsity at both a group-level and within-group level. Additionally, the efficient algorithm proposed by the authors can be used to fit both Lasso and Group Lasso as well. Employing such a variable selection method also has the advantage of improving the interpretability of models and potentially decreasing computational time. Thus, in high-dimensional setting, having structural knowledge of the dataset can be very helpful in improving the prediction accuracy, as well as, interpretability of models. Particularly, when the data contains natural grouping the Sparse-Group Lasso method is an effective technique to employ.

Bibliography

[1] Aleksandar Botev, Guy Lever, and David Barber. Nesterov's accelerated gradient and momentum as approximations to regularised update descent, 2016.

[2] Jonah Friedman, Trevor J. Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. 2010.

[3] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[4] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[5] Tuyen Trung Truong and Tuan Hang Nguyen. Backtracking gradient descent method for general c^1 functions, with applications to deep learning, 2018.

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