LOGISTIC GROUP LASSO FEATURE IMPORTANCE

HAMZA GOLUBOVIC & KIAN SARAF-POOR

Abstract. In this project, we investigate the group lasso logistic regression problem for feature importance. We define the proximal operator for the group lasso problem, and define the iterative algorithm for computing the proximal gradient descent of the group lasso regression problem. The algorithms presented are implemented in Python and applied to the MovieLens dataset. Feature importance ranking is extracted to determine which movie generes are most important in identifying male or female groups with respect to ratings.

1. Introduction and Overview

Many problems deal with classification for a binary outcome Y given data X. One classical method to tackle this problem is the logistic regression which assumes that the data

$$Y \sim \operatorname{Ber}(\rho(X)) \tag{1}$$

is Bernoulli distributed given X, where the likelihood of the outcome depends linearly on X and a parameter β:

$$\rho(X) = \beta^T X. \tag{2}$$

Then, one can estimate β by maximum likelihood and use that parameter for inference. Thereby, β can have components which significantly predict the outcome and some which do not. Those, typically have an estimate which is close to 0. In order to identify the most important features, one can use the Lasso regularization which uses the loss function

$$\mathcal{L}(\beta) = -\ell(\beta) + \lambda\|\beta\|_1, \tag{3}$$

where ℓ is the loglikelihood function of the logistic regression and $\lambda \geq 0$ is a tuning parameter. Typically, one then obtains a sparse estimate with many components set to 0.

In this paper we deal with a generalized version of the Lasso, called the group Lasso. We motivate it by means of the following example: Suppose one has grouped features and is interested in the significance of a whole group for the outcome. Therefore, one can use the Group Lasso penalty which is the focus of this paper. Thereby, we replace the $\lambda\|\beta\|_1$ by a Group Lasso penalty given by

$$\mathcal{L}(\beta) = -\ell(\beta) + \sum_{j=0}^{J} \lambda_j \|\beta_{(j)}\| \tag{4}$$

with tuning parameters $\lambda_0, \ldots, \lambda_J \geq$. Thereby, the $\beta_{(j)}$'s represent the parameters for the intercept and the J subgroups [2].

In the following we will introduce the notation and theory for this model and define the loss function. Further, we discuss the computation for the estimation by a proximal gradient descent and conclude with a case study.

Date: December 4, 2024.

2. Model Setup

2.1. **Notation.**

Suppose we have a n samples of binary outcomes

$$y = (y^{(1)}, \dots, y^{(n)})^T \in \{0,1\}^n \tag{5}$$

and a $n \times p$ data matrix of features

$$X = \begin{pmatrix} \mathbb{1} & X_1 & \cdots & X_p \end{pmatrix}, \tag{6}$$

where $\mathbb{1} = (1, \dots, 1) \in \mathbb{R}^n$. Thereby, assume that the p features are separated into J subgroups in whose significance for predicting the outcome we are interested in. We denote them as

$$X = \begin{pmatrix} \mathbb{1} & X_{(1)} & \cdots & X_{(J)} \end{pmatrix} \tag{7}$$

where the $X_{(j)}$'s are the $n \times p_j$ submatrices. When referring to the data we use the following notation,

$$X = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(n)} \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_p^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \vdots & & \vdots \\ 1 & x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix} = \begin{pmatrix} 1 & x_{(1)}^{(1)} & \cdots & x_{(J)}^{(1)} \\ 1 & x_{(1)}^{(2)} & \cdots & x_{(J)}^{(2)} \\ \vdots & \vdots & & \vdots \\ 1 & x_{(1)}^{(n)} & \cdots & x_{(J)}^{(n)} \end{pmatrix}. \tag{8}$$

Similarly, we partition the parameter vector $\beta \in \mathbb{R}^p$ as

$$\beta = (\beta_0, \beta_1, \dots, \beta_p) = (\beta_0, \beta_{(1)}, \dots, \beta_{(J)}). \tag{9}$$

In the following we write

$$\|z\| = \sqrt{\sum_{k=1}^n z_k^2} \qquad \|z\|_1 = \sum_{k=1}^n |z_k| \tag{10}$$

for the Euclidian norm and the L^1-norm, respectively.

2.2. **Theoretical Background.**

For our model setup we assume a logistic model which we introduce next:

Definition 1 (Logistic Model). *Assume that the outcomes $y^{(1)}, \dots, y^{(n)}$ are independent samples, where $y^{(k)} \sim Ber(\rho(x^{(k)}))$ has Bernoulli distribution. Thereby, we assume the canonical logit link function for ρ, which yields that*

$$\rho(x^{(k)}) = {x^{(k)}}^T \beta \tag{11}$$

Therefore, our loglikelihood function is given by

$$\ell(\beta) = \sum_{k=1}^n y^{(k)} {x^{(k)}}^T \beta - \log\left(1 + \exp({x^{(k)}}^T)\beta\right) \tag{12}$$

As we explained in the introduction, our features are grouped and we are interested in the significance of each group for the outcome. The typical way to pursue this problem for not necessarily grouped data is the Lasso regularization, which minimizes the loss function

$$\mathcal{L}(\beta) = -\ell(\beta) + \lambda \|\beta\|_1 \tag{13}$$

with a tuning parameter $\lambda \geq 0$. This usually yields a sparse estimate $\hat{\beta}$ which sets not significant components to 0. As we are interested in the significance of groups of features jointly, we use a more generalized loss function, given by

$$\mathcal{L}(\beta) = -\ell(\beta) + \sum_{j=0}^{J} \lambda_j \|\beta_{(j)}\|, \tag{14}$$

where $\lambda_0, \ldots, \lambda_J \geq 0$ are tuning parameters. Our estimate $\hat{\beta}$ is defined as

$$\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \mathcal{L}(\beta), \tag{15}$$

if a minimizer of the loss function exists. This concludes the theoretical section. In the following we discuss the computation of our estimate.

3. Computation

Observe, the group lasso problem introduces a bias term, which is non-differentiable. Therefore, this problem cannot be solved using traditional gradient descent. For this reason, we must define a new type of optimization algorithm, called proximal gradient descent.

Specifically, we formulate the proximal gradient descent algorithm to minimize the composite function

$$f(x) = g(x) + h(x) \tag{16}$$

where g is convex and differentiable, and h convex and simple, but non-differentiable. For the remainder of this section, we define t to be the learning rate (or step size), as traditionally used in gradient descent.

Definition 2. [3] *For step size t, the proximal mapping is defined as*

$$\mathrm{Prox}_{h,t}(x) = \arg\min_{z} ||x - z||^2 + h(z) \tag{17}$$

When conducting *proximal gradient descent*, we initialize x_0 and repeat the update

$$x_k = \mathrm{Prox}_{h,t_k}(x_{k-1} - t_k \nabla g(x_{k-1}))$$

This algorithm will allow us to find a closed-form solution for special cases of simple h, which are not differentiable. For example, for regular lasso, we have $g(\beta) = \frac{1}{2}||X\beta - y||^2$ and $h(\beta) = \lambda||\beta||_1$, whereby the proximal mapping is given by

$$\begin{aligned} \mathrm{Prox}_{h,t}(\beta) &= \arg\min_{z} \frac{1}{2t}||\beta - z||^2 + \lambda||z||_1 \\ &= \phi_{st}(\beta; \lambda) \end{aligned}$$

Here ϕ_{st} is the soft thresholding operator defined as

$$(\phi_{st}(\beta;\lambda))_i = \begin{cases} \beta_i + \lambda & \beta_i > \lambda \\ 0 & -\lambda \leq \beta_i \leq \lambda \\ \beta_i + \lambda & \beta_i < -\lambda \end{cases}$$

Therefore, the proximal gradient update is given by

$$\beta^{(k+1)} = \phi_{st}(\beta^{(k)} + t\,\underbrace{X^T(y - X\beta^{(k)})}_{\nabla g(\beta)}; \lambda)$$

In our problem formulation, we are dealing with the *group lasso penalty*, with logistic regression. In order to solve this problem, we first derive a proximal operator for the *group penalty*, and then derive the iterative updates for the logistic regression portion.

Theorem 1. [4] *We define the group lasso penalty as* $h(\beta) = \lambda \sum_{j=0}^{J} ||\beta_j||$. *For step size* t, *the proximal operator for the group lasso penalty is given by*

$$\begin{aligned} \text{Prox}_{h,t}(\beta) &= \phi_{bst}(\beta;\lambda) \\ &:= \left[\left(1 - \frac{\lambda t}{||\beta_j||}\right)_+ \beta_j \right]_{1\le j\le J} \end{aligned} \tag{18}$$

Where we define $a_+ = \max\{a, 0\}$, *and bst stands for block soft-threshold.*

Now that we have developed the proximal mapping for the group lasso penalty, we present the proximal gradient update for the group lasso logistic regression problem defined in the Theoretical Background section.

Theorem 2. *The solution to the group lasso logistic regression is iteratively given by the proximal gradient descent update*

$$\begin{aligned} \beta^{(k+1)} &= \text{Prox}_{h,t}(\beta^{(k)} - t\nabla g(\beta^{(k)})) \\ &= \left[\frac{u_j^{(k)}}{||u_j^{(k)}||} \cdot \left(||u_j^{(k)}|| - \lambda t\right)_+ \right]_{1\le j\le J} \end{aligned} \tag{19}$$

Where

$$u_j^{(k)} = \beta_j^{(k)} - t\sum_{i=1}^{n} \left(\frac{X_{j,i}}{1 + \exp(-\beta_0^{(k)} - \sum_{j=1}^{J} X_{j,i}^T \beta_j^{(k)})} - y_i X_{j,i} \right)$$

for $k = 1, 2, \ldots$.

4. Case Study

In our case study, we analyze the MovieLens 100K Dataset [1]. In this dataset, there are 943 individuals who rated a variety of movies from 1-5 stars. Specifically, each individual must rate at least 20 movies. Although we may impute missing data, for the sake of demonstrating this algorithm, we keep the matrix sparse by inserting 0 wherever a movie is not rated by a person. This way, we hope the algorithm will ignore these inputs as noise (whether or not the data was imputed). Further, for each movie, we know which genre it belongs to (total of 19 genres), and for each person, we have information about their gender (male or female). For this dataset, one may ask the question: which movie genres are most differently favored among the male and female groups? In other words, which genres are most important in classifying participants as male or female?

We apply the above proximal gradient descent algorithm in Python to the MovieLens data. Movies are grouped according to their genre listings and male/female categories are marked 0/1, respectively.

After testing various hyperparameters, we found that the best are $t = 10\text{e} - 6$ for the learning rate and $\lambda = 30$ for the L-1 penalty. We ran the algorithm for 6000 iterations and see that it converges in around 2500 iterations (Figure 1). In our findings, genres $1 \le j \le J = 19$ with non-zero β_j values ended up being action, musical, sci-fi, and war.

According to this result, we would expect these genres to be favored by one group, but not the other, on average.

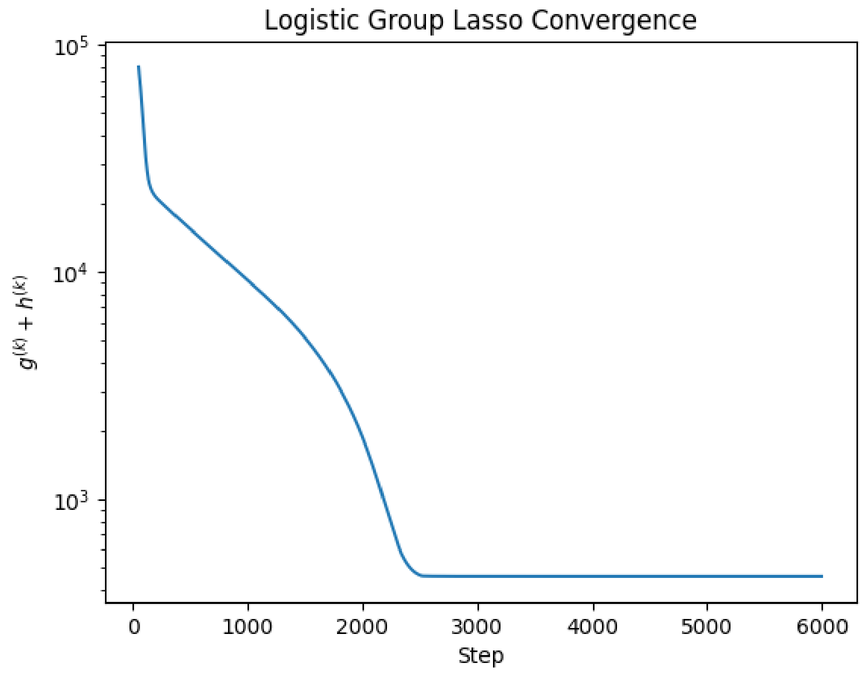

FIGURE 1. Convergence of the proximal gradient descent algorithm on MovieLens data with hyperparameters $t = 10e-6$ and $\lambda = 30$. At step k, error measured as objective function $g^{(k)} + h^{(k)}$ evaluated at the $\beta^{(k)}$ value.

5. Conclusion

In this work, we examined the closed form solution of the proximal gradient descent algorithm for group lasso logistic regression. Using this result, we were able to implement the iterative algorithm in Python and conduct an experiment on the MovieLens data. In the future, this code can be applied to further datasets where sparse groups form naturally within data. Furthermore, in the example of MovieLens, it would be interesting to examine classification of different age groups or even professions with respect to genre preferences.

References

[1] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, December 2015.

[2] L. Meier, S. van de Geer, and P. Buhlmann. The group lasso for logistic regression.

[3] R. Tibshirani, J. Taylor, and T. Hastie. *Statistical Learning with Sparsity: The Lasso and Generalizations.* Chapman and Hall/CRC, Boca Raton, FL, 2014.

[4] Yuxin Chen. ELE 520: Mathematics of Data Science. https://yuxinchen2020.github.io/ele520$_math_data/index.html$.