VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

Article

Ridge and Lasso logistic regression

Eligijus Bujokas

VILNIUS (2018.12.21)

Logistic Ridge and Lasso regression

Abstract

The aim of this paper is to give a brief introduction to Ridge, Lasso and elastic net binary logistic regression. The feature matrix is constructed from text.

The calculations and visuals are made using the statistical software **R**.

Key words : Logistic regression, Ridge, Lasso, Elastic Net

Contents

Introduction

In the modern world there are many clasification tasks that revolve around clasifying textual data. There are algorythms that based on a user inputed string can talk with a person, after reading lyrics of a song it can label the genre of the song and so on. Typical features (independant variables, regresors) are unique words thus even a small text can have thousands of collumns in the design matrix.

A big design matrix leads to big computational times and often leads to multicolinearity. The reduction of the number of features in a dataset or reweighting the importance of features often leads to a speed up in the computational time and better results. Ridge and Lasso regression are often the tools of choice when dealing with the mentioned problems.

Logistic regression for textual data

Introductory case

Let us assume that our $\mathbb{Y}$ variable is binary. Each class of $\mathbb{Y}$ could indicate whether a string is 'positive' or 'negative' in a semantic sense, 'sincere' or 'not sincere' and so on. Each column in the $\mathbb{X}$ matrix are also binary. Each collumn represent a unique word and each row value represent whether a word was observed in a given string or not.

For example, let us say we have a 'positive' review -'This was delicious'- and a 'negative' one -'The food was awfull'. Then raw document will have two rows:

sentiment	review
1	This was delicious
0	This was awfull

The matrix that is used for computations is often called the document term matrix (dtm for short). The dtm of our raw document would look like this:

delicious	awfull	was	this
1	0	1	1
0	1	1	1

As we can see, the matrix indicates that the first review had words *delicious*, *was* and *this* while the second review had the words *awfull*, *was* and *this*.

We can define a general linear model for this case:

$$log\frac{P(Y=1)}{P(Y=0)} = \beta_0 + \beta_1 \mathbb{1}_{delicious} + \beta_2 \mathbb{1}_{was} + \beta_3 \mathbb{1}_{this} + \beta_4 \mathbb{1}_{awfull}$$

According to the data, the coefficient β_1 should be positive and β_4 should be negative implying positive and negative sentiments respectivelly. The words *was* and *this* do not help in distinguishing between the positive and the negative sentiments thus the coefficients near these features need to be 0 or very small.

General case for binary dependant variable

Let us assume that our data is the set:

$$D_{\mathbb{Y},\mathbb{X}} = \{\mathbb{Y}_i \in \{0,1\}, \mathbb{X} = [1, X_{i1}, X_{i2}, ..., X_{ik}], \forall i \in \{1, ..., n\}\}$$

In most practical cases, the intercept is added to the design matrix.

In matrix form:

$$\mathbb{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \mathbb{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & ... & x_{1k} \\ 1 & x_{21} & x_{22} & ... & x_{2k} \\ ... & ... & ... & ... & \\ 1 & x_{n1} & x_{n2} & ... & x_{nk} \end{bmatrix}$$

In practise, all x_{ij} are either 0 or 1, indicating existance in certain row of the document term matrix, or the count of occurance in row in the document term matrix. The number of columns in the $\mathbb{X}$ matrix is equal to the number of unique words in our document.

The general model is then:

$$log\frac{P(Y=1)}{P(Y=0)} = \sum_{i=0}^{k} \beta_i X_i \, (1)$$

Maximum likelihood for estimating the coefficients

Let us define:

$$\theta := \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}$$

From (1), the probability of 'success' for each observation i can be rewritten as [1 , pp. 3-5]:

$$\pi_i := P(\mathbb{Y}_i = 1) = \frac{e^{\theta^T \mathbb{X}_i}}{1 + e^{\theta^T \mathbb{X}_i}} = \frac{1}{1 + e^{-\theta^T \mathbb{X}_i}}$$

$$P(\mathbb{Y}_i = 0) = 1 - \pi_i$$

In every binary case, the $\mathbb{Y}$ can be encoded as a vector consisting of 0 and 1. Thus, we want θ such that the product:

$$l(\theta) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1 - y_i} \ (2)$$

is the biggest .

Logarythm is a monotone function thus the maximum of $l(\theta)$ is the same as $log(l(\theta))$ [2 , pp. 636 - 637].

$$L(\theta) := log(l(\theta)) = \sum_{i=1}^{n} [y_i log(\pi_i) + (1 - y_i) log(1 - \pi_i)] = \sum_{i=1}^{n} \left[y_i log(\frac{\pi_i}{1 - \pi_i}) + log(1 - \pi_i) \right]$$

$$L(\theta) = \sum_{i}^{n} \left[y_i \theta^T \mathbb{X}_i + log(1 + e^{\theta^T \mathbb{X}_i}) \right] \ (3)$$

The $\widehat{\theta}$ that maximizes the (3) equation will give a weight to every unique word in our text. This, in practise, is ussualy not ideal, because even a small text document can have thousands of unique words.

Ridge logistic regression

Ridgre logistic regression introduces an additional term to the (3) equation - the L2 penalty.

$$L^R(\theta, \lambda) = \sum_i^n \left[y_i \theta^T \mathbb{X}_i + log(1 + e^{\theta^T \mathbb{X}_i}) \right] - \lambda \sum_{j=1}^k \beta_j^2$$

Often in practise, the λ parameter is fixed to a certain value. As the parameter λ increases, the ridge coefficient estimates will tend to approach zero. However, the penalty introduced in the log-likelihood function will shrink all of the coefficients towards zero, but it will not set any of them exactly to zero. Hence, ridge regression has the disadvantage over model selection, of including all the predictors in the final model.

On the other hand, Ridge regression estimates gives us more uniformally distributed weights to all words. Depending on the problem, one may view this as an advantage.

Lasso logistic regression

Lasso logistic regression is very similar to that of Ridge, but the penalty term is L1:

$$L^L(\theta, \lambda) = \sum_i^n \left[y_i \theta^T \mathbb{X}_i + log(1 + e^{\theta^T \mathbb{X}_i}) \right] - \lambda \sum_{j=1}^k |\beta_j|$$

The L1 penalty used in the lasso is used for both variable selection and shrinkage, since it has the effect, when the λ is sufficiently large, of forcing some of the coefficient estimates to be exactly equal to zero [2 , pp.637]. In Lasso regression, the final model may involve only a subset of the predictors, which in turn improves model interpretability.

Depending on the research subject and the problem, having less predictors is beneficial.

Elastic net regression

One is not confined to just using either Ridge or Lasso regression. The elastic net procedure tries to implement both of these methods:

$$L^{EN}(\theta, \lambda) = \sum_i^n \left[y_i \theta^T \mathbb{X}_i + log(1 + e^{\theta^T \mathbb{X}_i}) \right] - (\lambda \sum_{j=1}^k |\beta_j| + (1 - \lambda) \sum_{j=1}^k \beta_j^2)$$

$\lambda \in (0, 1)$.

This approach is particularly useful when the number of predictors is much larger than the number of observations [3].

Quora question example

References

[1] S. A. Czepiel. *Maximum Likelihood Estimation of Logistic Regression Models: Theory and Implementation*. Article. 2017. <URL: https://czep.net/stat/mlelr.pdf>.

[2] A. F. d. S. Jose Manuel Pereira Mario Bastoa. *The logistic lasso and ridge regression in predicting corporate failure*. Article. 2016. <URL: https://www.sciencedirect.com/science/article/pii/S2212567116303100>.

[3] H. Zou and T. Hastie. *Regularization and variable selection via the elastic net*. Article. 2005. <URL: https://web.stanford.edu/~hastie/Papers/B67.2%20(2005)%20301-320%20Zou%20&%20Hastie.pdf}.>