Logistic lasso regression

Shu Liu

Introducation

This vignette will provide a basic guidance for the functions fit_logistic_lasso() and predict_logistic_lasso() within a tidymodels workflow.

These functions are built for analyzing data by logstic lasso regression, so the following section will consist of algorithm and foundemental usage of the function.

The example I will use is the dataset in week 9 tutorial without step_intercept(). Here is a glimpse for this dataset:

```
## Rows: 1,000
## Columns: 4
## $ x   <dbl> -3.000000, -2.993994, -2.987988, -2.981982, -2.975976, -2.96997...
## $ w   <dbl> -3.000000, -2.999466, -2.997864, -2.995195, -2.991460, -2.98665...
## $ y   <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, ...
## $ cat <chr> "a", "c", "b", "a", "b", "c", "b", "a", "a", "c", "a", "c", "b"...
```

Logistic lasso regression

The function fit_logistic_lasso() will generate a lasso regression model for the selected dataset in order to predict further. The output for fit_logistic_lasso() will be a list containing intercept, beta and lambda.

The function has several input: data as x, vector of data as y, and maximum iteration number i.

In fit_logistic_lasso, we will firstly normalize input x to be a distribution with mean = 0 and standard deviation = 1. Then we are wishing to discover coefficient β^T and intercept. β is set to be 0 at the beginning. In order to find parameters, we are interested in w and z value. Then we calculate residuals by making difference between prediction and observed point. In the end, we can calculate β, then repeat the above procedures. By iterating i times, we can get beta, intercept.

Here is the application of the function, we set penalty to be 0.3:

```
##         x         w     cat_b     cat_c
## -3.728017  2.294021  0.000000 -0.267606
```

```
## (intercept)
##  -0.1856197
```

The above is part of the output for our logistic lasso regression, then we can use it for prediction.

```
##           Truth
## Prediction   0   1
##          0 113  16
##          1  12 108
```

Now we can compare it with the answer by glm.

```
## # A tibble: 5 x 2
##   term        estimate
##   <chr>          <dbl>
## 1 (Intercept) -0.192
## 2 x           -3.77
## 3 w            2.32
## 4 cat_b        0.00351
## 5 cat_c       -0.275
```

The results are consistent, then we are done for these functions.