# SML: Exercise 2

Thao Le, Finn-Ole Höner, Jason Wang, Ramon Snellen

2022-11-07

## Introduction

There are many procedures for finding a best set of predictor variables and their corresponding coefficients. Of these procedures, some require numerical optimization. The elastic net is such a procedure. To numerically optimize elastic net estimation, different algorithms can be used. This report aims to answer the research question: how does the elastic net estimation of the 'cyclical coordinate descent' algorithm and 'MM' algorithm differ? A cyclical coordinate descent algorithm for the elastic net has already been developed in the `glmnet` package in `R` ("Glmnet: Fit a GLM with Lasso or Elasticnet Regularization" 2021). Therefore, we code the elastic net estimation using an MM algorithm, and compare the estimation output to that of the `glmnet` package.

## Data

The data set contains seven years of store-level data collected at Dominick's Finer Foods by the University of Chicago Booth School of Business. The data can be found at https://www.chicagobooth.edu/research/kilts/datasets/dominicks. The data set contains 50 variables, which stem from:

1. customer count files, which contain information about in-store traffic;
2. a demographics file, which contains store-specific demographic data;
3. number identification files, which contain product information.

Of the fifty variables, `GROCERY_sum` is used as dependent variable. Furthermore, four categorical variables are dropped; `STORE`, `CITY`, `ZIP` and `SPHINDX`. The remaining variables are potential predictor variables.

## Method

The elastic net is a form of penalized regression. In this section, we discuss the elastic net and how we use an MM algorithm to estimate it. In addition, we introduce diagnostics to (1) determine the optimal set of hyperparameters for the elastic net; and (2) compare the performance of the elastic net estimated through an MM algorithm to that of the `glmnet` package.

Let $P(\beta)$ denote a general penalty function. Then, the loss function of the regression equation becomes

$$L(\beta) = (\mathbf{y} - \mathbf{x}\beta)^T(\mathbf{y} - \mathbf{x}\beta) + \lambda P(\beta), \tag{1}$$

where $\lambda$ is the hyperparameter that determines the strength of the penalty. When $P(\beta) = \beta^2$, the regression is called 'ridge' regression. Similarly, when $P(\beta) = |\beta|$, the regression is called 'LASSO' regression. Finally,

any convex combination $P(\beta) = \alpha|\beta| + (1-\alpha)\beta^2$ of the 'ridge' and 'LASSO' penalty, where $\alpha$ denotes the weight on the 'LASSO' penalty, is called 'elastic net' (compare Zou and Hastie 2005). We use the elastic net to find the optimal set of predictor variables, since it exploits the LASSO property of variable selection, and the ridge ...

We use an MM-algorithm to estimate the elasting net. The MM-algorithm uses a majorizing function to find the coefficient vector $\beta$ that minimizes the loss function specified in Equation 1. This is because the minimum is not obtained analytically. Let $\epsilon$ denote the desired level of precision. The majorizing function is specified as

$$L_{MM}(\beta) = \frac{1}{2}\beta^{\mathrm{T}}(\frac{1}{n}\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda(1-\alpha)\mathbf{I} + \lambda\alpha\mathbf{D})\beta - \frac{1}{n}\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{y} + c,$$

where $\mathbf{D}$ is a $p \times p$ diagonal matrix with elements

$$d_{jj} = 1/\max\left(|\beta_j^0|, \epsilon\right).$$

Furthermore,

$$c = \frac{1}{2n}\mathbf{y}^{\mathrm{T}}\mathbf{y} + \frac{1}{2}\lambda\alpha\sum_{i=1}^{p}|\beta_j^0|.$$

To find the optimal penalty strength $\lambda$ and the weight of the elastic net $\alpha$, $K$-fold cross-validation is used. $K$-fold cross-validation starts with splitting the data set into $K$ folds. Subsequently, $K$ iterations are ran as follows: for each iteration $k \in \{1, \ldots, K\}$, fold $k$ is the test set, whereas the remaining $(K-1)$ folds together form the training set. In this report, we use 10 fold cross validations. On the training set, the elastic net is estimated for all 50 values of $\lambda_i \in (10^{-2+i})_{i=0}^{12}$. Moreover, for each $\lambda_i$, all 50 values of $\alpha$ equally spaced between 0 and 1 are estimated, such that there are 2500 combinations of the hyperparameters $\alpha$ and $\lambda$ for each fold. All estimated elastic nets (i.e., all combinations of the hyperparameters), are used to predict the dependent variable in the $k$th fold. When this procedure has been completed for all $K$ folds, we have obtained fitted values for every observations in the data set, for all elastic nets. Consequently, prediction errors can be computed.

To evaluate which elastic net performs best (that is, the elastic net that maintains the optimal combination of hyperparameters), the Root Mean Square Error (RMSE) is computed for each elastic net; the elastic net that has the lowest RMSE is deemed to be the best model. Let $\hat{\mathbf{y}}_{\mathrm{test}(k)} = \mathbf{X}_{\mathrm{test}(k)}\hat{\beta}_{\mathrm{train(k)}}$ denote the fitted values, and $n_k$ the number of observations in fold $k$. The RMSE is computed as

$$\mathrm{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\mathrm{MSE}_k},$$

where

$$\mathrm{MSE} = \frac{1}{n_k}(\mathbf{y}_{\mathrm{test}(k)} - \hat{\mathbf{y}}_{\mathrm{test}(k)})^{\mathrm{T}}(\mathbf{y}_{\mathrm{test}(k)} - \hat{\mathbf{y}}_{\mathrm{test}(k)}).$$

To validate the results of our MM model, we compare our coefficient estimates with the ones of the established `glmnet` algorithm. We do not expect the coefficients to match exactly, e.g. due to different optimization algorithms. Hence, we need to measure their differences. We do so with the Absolute Error (AE)

$$\mathrm{AE}_i = \left|\hat{\beta}_i^{GLM} - \hat{\beta}_i^{MM}\right|,$$

and the Absolute Percentage Error (APE)

$$\mathrm{APE}_i = \left|\frac{\hat{\beta}_i^{GLM} - \hat{\beta}_i^{MM}}{\hat{\beta}_i^{GLM}}\right|.$$

where $\hat{\beta}_i^{GLM}$ and $\hat{\beta}_i^{MM}$ are the i-th coefficients of the `glmnet` and MM model, respectively.

Indeed, when applying the `glmnet` and MM implementations to the Dominick's Finer Food data set, we get different coeffciients. Figure 1 shows the mean-absolute-percentage differences (MAPE) of these two implementations for the different predictors. For most predictors, the $APE$ is around to 1%.

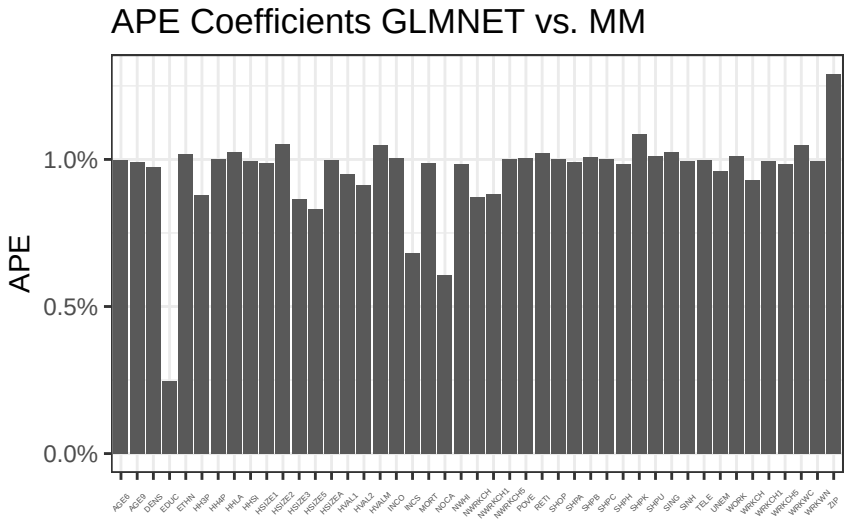


Figure 1: Absolute percentage difference between glmnet and MM coefficients.

# Results

One explanation for this difference could lie in the algorithms' speed of convergence. For a given data set the MM algorithm might not converge fast enough to provide the same estimates as the `glmnet` implementation. In fact, the cyclical coordinate decent used in `glmnet` is a very efficient algorithm according to Friedman, Hastie, and Tibshirani (2010).

We investigate this hypothesis by looking at the estimates' differences for lower convergence thresholds $\epsilon$ of the MM algorithm. Figure 2 shows the zero correlation between the Mean Absolute Error and the parameter $\epsilon$: This precision parameter appears to play no role in the discrepancy between the coefficients of `glmnet` and the MM algorithm.

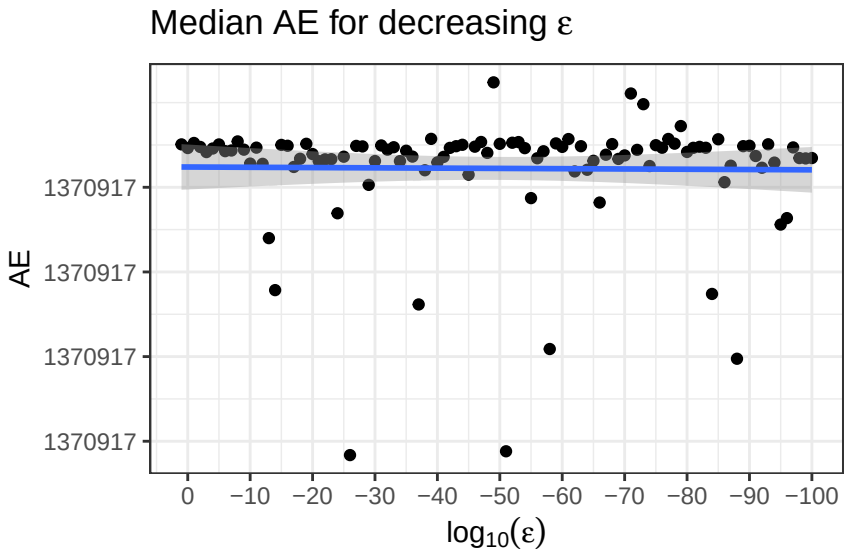


Figure 2: Development of AE fo decreasing precision threshold $\epsilon$.

The uniform difference between the coefficients, and the result that this difference does not depend on the precision parameter $\epsilon$. This leads us to believe that there are some structural differences between the `glmnet` and MM algorithm. However, as these discrepancies are low and similar across predictors, both algorithms seem to implement the same model.

### Test on real data using k-fold

First we find the optimal Lambda and Alpha value for our MM-algorithm

```
## Alpha is:  0.5102041 . The minimum lambda is:  0.9102982
##  The minimum RMSE is:  21194047
```

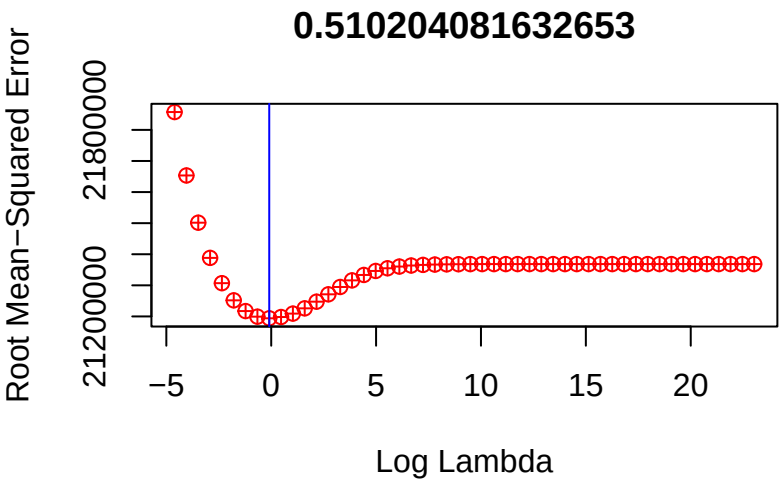


Figure 3: Changes in RMSE for each $\lambda$ using the best cross validated $\alpha$ MM

```
## integer(0)
```

Then, we compare to the optimal lambda and alpha value using the `glmnet` package.

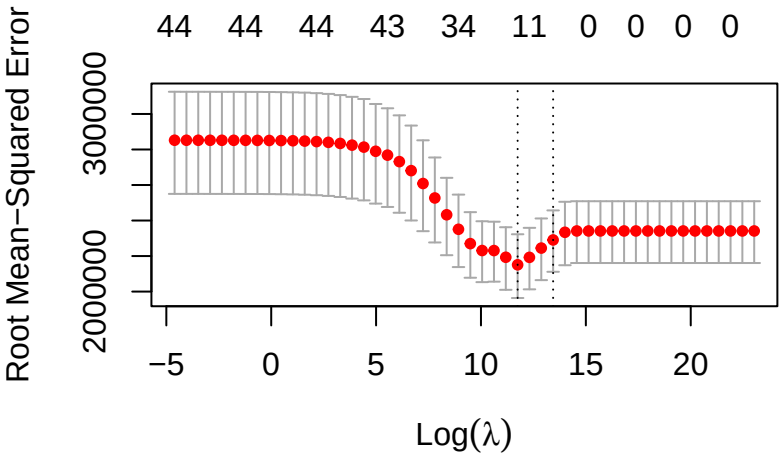


Figure 4: Changes in RMSE for each $\lambda$ using the best cross validated $\alpha$ GLMET

The plot shows that the optimal $\lambda$ using the `glmnet` is approximately $10^{12}$.

## Conclusion and Discussion

To answer our research question, the estimations based on the `glmnet` and MM algorithm differ with respect to the size of the coefficients. In addition, the prediction $RMSE$ of the MM algorithm is higher than the one of the `glmnet` implementation. Regarding to k-fold cross validation results, using the MM algorithm results in higher RMSE than using the `glmnet` package. Moreover, for a fixed value of optimal alpha that we chose for the MM algorithm, the `glmnet` returns a lower RMSE on average (10 times lower than the MM algorithm). Another difference is that for the MM algorithm, a much lower $\lambda$ of 0.9102982 is need, compare to the lambda value of approximately $10^{12}$ using the `glmnet` package. One explanation is that the `glmnet` uses cyclical coordinate descent while our elastic net function uses the MM method. One limitation of our research is that we have not pinpointed the reasons why these two optimization methods differ. Future research can look into this limitation.

## References

## Code


Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1.

"Glmnet: Fit a GLM with Lasso or Elasticnet Regularization." 2021. *RDocumentation*. https://www.rdocumentation.org/packages/glmnet/versions/4.1-4/topics/glmnet.

Zou, Hui, and Trevor Hastie. 2005. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2): 301–20.