Contents

1 LASSO Regression **1**

1.1 The LASSO 1

1.2 Optimization 2

1.3 TODO here 2

1 LASSO Regression

1.1 The LASSO

We want to create a model that has predictive and explanatory power. Predictive power meaning it can predict the precipitation in the Central Amazon Basin, "reasonably well". Explanatory power in the sense of being interpretable, so that we can identify those regions in sea that give us most information about future precipitation. Our problem setting is high dimensional with n « p. The number of predictors is a lot bigger than the number of actual observations. This creates issues with a classic linear model since the linear problem is underdetermined. One possible model for the problem at hand is a LASSO regression model.

In general for the linear model:

$$y_i = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i \tag{1}$$

see (1) Where the y_i refers to the mean precipitation for a given month i and x_i is the vector of sea surface temperature at different locations around the globe. ϵ_i is the residual and we wish to estimate the β´s from the data. As already stated this is not possible with a classic linear model since the number of predictors exceeds the number of observations. We therefore can not estimate a β for every grid point in the sea. From a physical point of view it also seems reasonable that some regions in the ocean have a higher predictive power than others. For example regions that are closer to the Amazon may have more influence on precipitation in the same month. But regions more far away may have more information on the precipitation half a year in the future. We therefore would like to use a model that can find the most important regions in the sea for predicting precipitation for some point in the future. One possible solution for this is a LASSO regression model, as implemented in R by the *glmnet* package (@glmnet-package). This model "automatically" performs model selection, but be aware that because of the time dependencies in our data, normal Cross Validation methods may be unjustified or at least have to be applied with caution. The glmnet package implements the regression problem in the following manner, solving:

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1] \tag{2}$$

This is a lasso regression for $\alpha = 1$ and ridge regression for $\alpha = 0$, α controls the overall strength of regularization or penalty. Intuitively this means we try to find those β´s that minimize the negative log likelihood of our data (this is equal to maximizing the log-likelihood). But at the same time we can not include too many β since this will make the second and third term in the formula grow. As result the algorithm chooses only those predictors that have the most predictive power. How many predictors are included depends on the strength of regularization given by α. *Remark*: Among strongly correlated predictors only one is chosen in the classical lasso model. Ridge regression shrinks the coefficients to zero. Elastic net with $\alpha = 0.5$ tends to either include or drop the entire group together. To specifically choose a group of predictors, variations of the lasso or other models have to be considered.

1.2 Optimization

The glmnet function finds a solution path for the lasso problem via coordinate descent. The implemented algorithm was suggested by @van2007prediction. We can write down the optimization procedure as follows: Given N observation pairs (x_i, y_i) with $Y \in \mathbb{R}$ and $X \in \mathbb{R}^p$, we approximate the regression function with $E(Y|X = x) = \beta_0 + x^T\beta$, Here x_{ij} are considered standardized, so $\sum_{i=1}^N = 0, \frac{1}{N}\sum_{i=1}^N x_{ij}^2 = 1$ for $j = 1, ..., p$. We then solve the problem:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} \big[\frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - x_i^T\beta)^2 + \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1\big] \tag{3}$$

Note that this solves the elastic net problem that also uses a ridge penalty. We follow the elastic net description but in our case $\alpha = 1$, using only the lasso penalty. We consider now a coordinate descent step for solving (3). Given we have estimates $\tilde{\beta}_0$ and $\tilde{\beta}_l$ and we want to partially optimize with respect to β_j, and $i \neq j$. When $\beta_j > 0$,

$$\left.\frac{\partial R_\lambda}{\partial \beta_j}\right|_{\beta=\tilde{\beta}} = -\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \tilde{\beta}_0 - x_i^\top\tilde{\beta}\right) + \lambda(1-\alpha)\beta_j + \lambda\alpha. \tag{4}$$

And similar expressions exist for $\tilde{\beta}_j < 0$. $\tilde{\beta}_j = 0$ is treated separately. The coordinate-wise update has then the form:

$$\tilde{\beta}_j \leftarrow \frac{S\left(\frac{1}{N}\sum_{i=1}^N x_{ij}\left(y_i - \tilde{y}_i^{(j)}\right), \lambda\alpha\right)}{1 + \lambda(1-\alpha)}. \tag{5}$$

with

- $\tilde{y}_i^{(j)} = \tilde{\beta}_0 + \sum_{\ell\neq j} x_{i\ell}\tilde{\beta}_\ell$ standing for fitted value without the contribution from x_{ij}, and therefore $y_i - \tilde{y}_i^{(j)}$ is the partial residual when fitting β_j. Because we applied a standardization, $\frac{1}{N}\sum_{i=1}^N x_{ij}\left(y_i - \tilde{y}_i^{(j)}\right)$ denotes the simple least-squares coefficient for fitting this partial residual to x_{ij}.
- $S(z, \gamma)$ being the soft-thresholding operator. It's value is given by:

$$\text{sign}(z)(|z| - \gamma)_+ = \begin{cases} z - \gamma & \text{if } z > 0 \text{ and } \gamma < |z| \\ z + \gamma & \text{if } z < 0 \text{ and } \gamma < |z| \\ 0 & \text{if } \gamma \geq |z|. \end{cases} \tag{6}$$

So in summary the steps are as follows: Compute the simple least-squares coefficient on the partial residual, then apply soft-thresholding and proportional shrinkage for the lasso and ridge penalty, respectively. Again for our use case, since we use the lasso and $\alpha = 1$, we only apply soft-thresholding and no proportional shrinkage.

The solutions are computed starting from smallest λ_{max} for which all elements in $\hat{\beta} = 0$. For all larger λ the coefficients then stay 0. The smallest λ value λ_{min} is then selected by λ_{min} λ_{max}. The complete searched vector is constructed as sequence of K values, typical values are $\epsilon = 0.001$ and $K = 100$. This procedure is an example of so called *warm starts*. By default they always center the predictor variable. For additional information on other methods how speedup is obtainred refer to Section 2 in @glmnet-package.

1.3 TODO here

maybe drop into with linear model just put lasso formula directly talk about the stuff that is written already note probelms with correlation of predictors and grouping