



Data Science Methods - Homework Assignment 2

Group 20: Steffie van Poppel (2031218), Robbie Reyerse (2039047), Mike Weltevrede (1257560)

March 21, 2020

Question 1: Forecasting financial crises

In this exercise, we are asked to apply two methods (we choose to apply a LASSO and a ridge regression) to a dataset on many macroeconomic variables to forecast financial crises. The dataset that we will use was also used by Ward (2017, Journal of Applied Econometrics) where they contrast the performance of one tree with bagging and a random forest against the logit benchmark. We wish to compare our results to those as presented in Table 1.

| Results | | | | | | |
|---------------|----------------------|-------------|------|-----------------|-------------|------|
| Model | Restricted Selection | | | Many Predictors | | |
| | AUC | 95%-CI | N | AUC | 95%-CI | N |
| Single Tree | 0.55 | [0.49,0.6] | 1816 | 0.63 | [0.56,0.69] | 1742 |
| Bagging | 0.77 | [0.73,0.81] | 1816 | 0.87 | [0.84,0.9] | 1742 |
| Random Forest | 0.79 | [0.75,0.83] | 1816 | 0.88 | [0.86,0.91] | 1742 |
| Specification | | | | | | |
| Parameter | Restricted Selection | | | Many Predictors | | |
| | Single | Bagging | RF | Single | Bagging | RF |
| B | 1 | 5000 | 5000 | 1 | 5000 | 5000 |
| J_{try} | 10 | 10 | 3 | 76 | 76 | 9 |
| J | | 10 | | | 76 | |
| # of crises | | 72 | | | 70 | |

Table 1: Results from Ward (2017).

We first need to do data preparation. We follow the same data preparation procedure as Ward. We are,

however, only interested in the case where “many predictors” are used.

```
data_path = "data"
df_data = read.table(paste0(data_path, "/R_class.csv"), sep=",", dec=".",
                     header=TRUE)

ca = grep("ca", names(df_data), value=T)
df_data = df_data[!(names(df_data) %in% c(ca))]

# drop vars not used
stocks = grep("stocks", names(df_data), value=T)
money = grep("money", names(df_data), value=T)
stir = grep("stir", names(df_data), value=T)
assets = grep("assets", names(df_data), value=T)
i = grep("i_", names(df_data), value=T)
ri = grep("ri", names(df_data), value=T)
glo = grep("a_", names(df_data), value=T)

drops = names(df_data) %in% c("year", "ccode", stocks, money, stir, assets, i,
                             ri, glo)
full_om = na.omit(cbind(df_data[glo], df_data[!drops]))
```

Next, we run a LASSO and a ridge regression model. The function that we use for this is `lasso_ridge_sim`. Using the parameter `alpha`, we can specify whether we want a LASSO regression (`alpha=1`) or a Ridge regression (`alpha=0`).

```
lasso_ridge_sim = function(data, grid_lambda = 10^seq(2, -3, length=100),
                           alpha=1, num_runs=100){

  lambdas = vector("numeric", num_runs)

  if (alpha==1){
    nzeros = vector("numeric", num_runs)
  }

  auc = vector("numeric", num_runs)
  ci95_auc_lo = vector("numeric", num_runs)
  ci95_auc_up = vector("numeric", num_runs)
  precisions = vector("numeric", num_runs)
  recalls = vector("numeric", num_runs)
  f_measures = vector("numeric", num_runs)

  for(j in 1:num_runs) {

    set.seed(j)

    # Select training and test data
    train_labels = sample(1:nrow(data), floor(nrow(data)*0.5))
    train = data[train_labels, ]
    test = data[-train_labels, ]
    train_matrix = model.matrix(b2 ~ ., data=train)
    test_matrix = model.matrix(b2 ~ ., data=test)

    # Train the LASSO/Ridge model
    model = glmnet::cv.glmnet(train_matrix, train[, "b2"], alpha=alpha,
```

```

        lambda=grid_lambda, thresh=1e-12,
        family="binomial")

lambdas[j] = model$lambda.1se

if (alpha==1){
  nzeros[j] = model$nzero[[which(model$lambda == model$lambda.1se)]]
}

# Test the LASSO/Ridge model
prediction = predict(model, newx=test_matrix, s=model$lambda.1se,
                     type="class")

# ROC analysis
r = pROC::roc(test[, "b2"], as.numeric(prediction), ci=T, quiet=T)
auc[j] = as.numeric(r$auc)

ci95_auc_lo[j] = as.numeric(ci.auc(r, conf.level = r$ci[2]))[1]
ci95_auc_up[j] = as.numeric(ci.auc(r, conf.level = r$ci[2]))[3]

# Classification evaluation methods
precisions[j] = caret::precision(factor(prediction, levels=c(0,1)),
                                factor(test$b2, levels=c(0,1)))
recalls[j] = caret::recall(factor(prediction, levels=c(0,1)),
                           factor(test$b2, levels=c(0,1)))
f_measures[j] = caret::F_meas(factor(prediction, levels=c(0,1)),
                              factor(test$b2, levels=c(0,1)))
}

results = list(auc = mean(auc),
              ci95_auc_lo = mean(ci95_auc_lo),
              ci95_auc_up = mean(ci95_auc_up),
              precision = mean(precisions),
              ci95_precision_lo = mean(precisions) - qnorm(0.975)*
                sd(precisions) / sqrt(nrow(test_matrix)),
              ci95_precision_up = mean(precisions) + qnorm(0.975)*
                sd(precisions) / sqrt(nrow(test_matrix)),
              recall = mean(recalls),
              ci95_recall_lo = mean(recalls) - qnorm(0.975)*sd(recalls) /
                sqrt(nrow(test_matrix)),
              ci95_recall_up = mean(recalls) + qnorm(0.975)*sd(recalls) /
                sqrt(nrow(test_matrix)),
              f_measure = mean(f_measures),
              ci95_f_measure_lo = mean(f_measures) - qnorm(0.975)*
                sd(f_measures) / sqrt(nrow(test_matrix)),
              ci95_f_measure_up = mean(f_measures) + qnorm(0.975)*
                sd(f_measures) / sqrt(nrow(test_matrix)),
              lambda = mean(lambdas),
              ci95_lambdas_lo = mean(lambdas) - qnorm(0.975)*sd(lambdas) /
                sqrt(nrow(test_matrix)),
              ci95_lambdas_up = mean(lambdas) + qnorm(0.975)*sd(lambdas) /
                sqrt(nrow(test_matrix))
            )

```

```

if (alpha == 1){
  results[["nzeros"]] = mean(nzeros)
  results[["ci95_nzeros_lo"]] = mean(nzeros) - qnorm(0.975)*sd(nzeros) /
    sqrt(nrow(test_matrix))
  results[["ci95_nzeros_up"]] = mean(nzeros) + qnorm(0.975)*sd(nzeros) /
    sqrt(nrow(test_matrix))
}

return(results)
}

lasso_results = lasso_ridge_sim(full_om)
ridge_results = lasso_ridge_sim(full_om, alpha=0)

```

In addition to the AUC used by Ward, we have also added the precision, recall, and F-measure (AKA the F-Score or the F1-Score) to our analysis. These are commonly used evaluation metrics for categorical response variables. Since our variable is binary, these are applicable. We did not use accuracy as a measure since we have only 140 observations where our variable has value 1 compared to 1602 observations with value 0. As such, the majority class of zeros will overpower the minority class with value 1.

For completeness sake, these are the definitions of the three metrics:

- $Precision = \frac{\#True\ Positives}{\#True\ Positives + \#False\ Positives}$, i.e. if we predict a crisis, how often is this true?
- $Recall = \frac{\#True\ Positives}{\#True\ Positives + \#False\ Negatives}$, i.e. out of the total number of crises, how many do we correctly identify?
- $F\text{-measure} = \frac{2 \times Precision \times Recall}{Precision + Recall}$, i.e. the harmonic mean of the precision and recall scores. This balances the precision and recall measures. It is often used to provide an additional check to values of precision and recall.

Given these definitions, we want a high precision if we want to minimise the number of false positives and a high recall when we want to minimise false negatives. In this case, our intuition would be that we would be most interested in the former. That is, we want to minimise the amount of times that a crisis is coming but that we do not identify this rather than minimising the amount of times where we say that a crisis is coming while this is not true. This is because a crisis can have a grave impact on many people in society and we would rather take too many precautions than too few. However, the other case is also not desirable; therefore, the F -measure is also important to consider.

The results from our analysis using the LASSO and ridge regression models can be found in Table 2.

| Results | | | | |
|---------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| Model | AUC | Precision | Recall | F-measure |
| LASSO | 0.8421 [0.8143, 0.8699] | 0.9739 [0.9726, 0.9752] | 0.9823 [0.9809, 0.9836] | 0.9777 [0.9775, 0.9779] |
| Ridge | 0.8112 [0.7736, 0.8489] | 0.9681 [0.9675, 0.9687] | 0.9922 [0.9918, 0.9926] | 0.9799 [0.9797, 0.9801] |
| Specification | | | | |
| Parameter | LASSO | Ridge | | |
| λ | 0.0162 [0.0158, 0.0166] | 0.0318 [0.031, 0.0325] | | |
| Mean # nonzero parameters | 3.55 [3.3633, 3.7367] | - - | | |

Table 2: Our results (Ranges indicate 95% confidence intervals)

Firstly, we will compare our results to that of Ward using the AUC for the ROC curve (hereafter referred to only as the AUC). After this, we will look into the measures of precision, recall, and the F -measure. We weigh these off to the AUC measure that Ward uses. Lastly, we look into whether these models actually are applicable to the specific data that we are analysing.

To first compare our results with those of Ward, we can only look at the AUC as a quantitative measure since Ward reports no other measures. Ward achieves AUC values of 0.63, 0.87, and 0.88 for the Single Tree, Bagging, and Random Forest methods, respectively. Our analyses using LASSO and ridge regression achieve AUC values of 0.8421 and 0.8112. Because we want to achieve an AUC value as close to 1 as possible, we can see that the LASSO and ridge regressions perform a bit worse than the Bagging and Random Forest methods used by Ward. However, note that AUC is deemed to not be a good metric in case of imbalance in the response as we have explained before. This is because one can achieve a high AUC when the model can identify the majority class well even though it may be very bad at identifying the minority class. Therefore, we think that it would be better to consider precision, recall, and the F -measure.

In that case, we see that the LASSO model achieves the values 0.9739, 0.9823, and 0.9777 for the precision, recall, and F -measure, respectively. The ridge model achieves the values 0.9681, 0.9922, and 0.9799 for the precision, recall, and F -measure, respectively. One can see that the LASSO model has a higher precision while the ridge model achieves a higher recall and F -measure. Following our arguments of before, we should value recall a bit more than precision, though this is not set in stone. Therefore, we can also consider the F -measure as a balance between the two measures. In that case, the ridge model achieves a slightly higher value for the F -measure though not by much. Since the confidence interval of the F -measure for the ridge regression does not include the mean F -measure for the LASSO regression, it is significantly higher for the ridge regression.

We would then say that the choice comes to the final part of our analysis: do these models actually fit the data type? It is important to recognise that the LASSO model sets certain parameters to zero and that the ridge model shrinks them to zero. Although we cannot say that the LASSO model sets all parameters to zero that are truly zero (in exercise 2 we discuss that another model called adaptive LASSO is able to do this for estimation purposes), it seems intuitive that 82 variables are likely too many to describe whether a crisis would occur. The problem comes when we look at how many nonzero parameters LASSO selects, namely only 3.55 (on average). It seems unintuitive that only 3 to 4 parameters can accurately predict whether a crisis will occur. Nonetheless, apparently the LASSO model still does well on the basis of our evaluation metrics so it is not that bad. For the ridge model, some parameters are shrunk towards zero but none are set

equal to zero (with probability 1). As such, all 82 variables are kept in the model but some will have lower coefficients. This makes the model in itself less intuitive to interpret but we are more interested in prediction than estimation so this is not a problem.

All in all, this means that we prefer the ridge model over the LASSO model. Comparing to the models by Ward, we can only use the AUC and our intuition on the applicability of the model to the specific data. On the AUC side, we saw that the AUC values for Ward's models were a bit higher than the LASSO and ridge models. However, also recall that this is not that applicable since the data is imbalanced in the response variable. Therefore, our final conclusion comes to the applicability of the models. Here, we unfortunately do not see a clear reason why we choose one model over the other purely on a theoretical level. As such, we make our conclusion based on the fact that the precision, recall, and F -measure values for the ridge regression model are quite high (also considering the lower bound of its confidence interval); high enough to choose this model over Ward's Random Forest, for example, as we do not have any information on this method's precision, recall, and F -measure.

Question 2: LASSO

Subquestion a: Plain versus adaptive LASSO

The objective functions of plain LASSO is shown in equation (1).

$$\min_{\beta} \left(RSS + \lambda \sum_{j=1}^p |\beta_j| \right), \quad (1)$$

where the residual sum of squares RSS is defined in the usual way as in equation (2):

$$RSS = \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \quad (2)$$

The second part of (1) is the LASSO penalty, where λ is the tuning parameter which controls the strength of the penalty.

Plain LASSO can be used when we assume $y = X\beta + \epsilon$, where ϵ is i.i.d and β is sparse. The last means that a lot of coefficients should be believed to be zero. This is a reasonable thing to assume if the number of predictors, p , grows quickly with n . It will then set some of the true zero parameters to zero asymptotically (as $n \rightarrow \infty$). For prediction it is not a problem that not all zero coefficients are set to zero. Since it safeguards that some coefficients will matter for out of sample while maybe in sample they did not in case of finite samples.

In case of prediction, λ is determined by choosing the lambda is that yields one standard deviation above the minimum cross-validation λ . If $\lambda = 0$ there is no penalization and the plain LASSO solution will be identical to the least squares solution. On the other hand, when $\lambda = \infty$ all penalized coefficients will be zero.

Adaptive LASSO, however, is able to give the all the true zero coefficients, which is essential for estimation (in-sample). The objective of adaptive LASSO is shown in equation (3).

$$\min_{\beta} \left(RSS + \lambda \sum_{j=1}^p |\beta_j| w_j \right), \quad (3)$$

where

$$w_j = \frac{1}{|\hat{\beta}_j|^\gamma} \text{ for } \gamma \geq 1. \quad (4)$$

In fact, when the same assumptions hold as for plain LASSO and in addition $\frac{\lambda_T}{\sqrt{T}} \rightarrow 0$ and $\lambda_T^{\frac{\gamma-1}{2}} \rightarrow \infty$ adaptive LASSO selects the true non-zero coefficients with a probability 1 as $n \rightarrow \infty$.

$\hat{\beta}_j \xrightarrow{P} \beta_j$ holds for $\hat{\beta}_j$ used in the weight of equation (4), in other words the coefficients pre-estimates converge in probability to the true coefficients.

As can be seen in the objective function of adaptive LASSO, the only difference between the plain LASSO is the penalty. Here, it is weighted with w_j (equation (4)). This means that penalization is done proportional to the values of the $\hat{\beta}_j$. So if the pre-estimates are large we penalize less and vice versa. The pre-estimates can be determined by e.g. plain LASSO.

In case we are interested in in-sample estimation, λ can be chosen by BIC. However, there are still some problems because λ is random. In the objective function, λ is treated as only changing with the sample size, so it is not random. But then the β s are nonlinear functions of the random data. There has not yet been a good solution to this.

Subquestion b: Post-adaptive LASSO

When we are interested in estimation, we are usually interested in the effects of covariates on the dependent variable. Hence, there is no need to report zero-valued coefficients unless we want to report parameters that do not have a significant impact on the dependent variable. A main advantage of this is that, when we want to interpret a specific coefficient, we do not have to control for all the covariates for which its coefficient was proven to be zero by adaptive LASSO. When we would directly report the adaptive LASSO results, even though the coefficients were zero, there we had included them in the regression and therefore should be controlled for.

Moreover, although adaptive LASSO will consistently pick all the zero-coefficients, it can still be biased in finite samples. It however aims less estimation biased than plain lass by allowing a relatively higher penalty for zero coefficients and, lower penalty for nonzero coefficients¹. A bias forms major problems for estimation. Post-lasso estimates have been proven, by Windmeijer et al.², to be less biased than LASSO estimates reported on its own. In addition, Belloni et al.³ showed that post-adaptive LASSO can improve inference on the parameter of interest because redundant regressors were penalized away in their procedure.

All in all, for estimation purpose - which is the case when one chooses adaptive LASSO instead of plain LASSO - post-adaptive LASSO will improve interpretation of the the coefficients by decreasing the bias and it makes interpretation a lot easier since it is no longer needed to control for the zero coefficients predictors.

¹Huang, J., Ma, S., & Zhang, C. H. (2008). Adaptive LASSO for sparse high-dimensional regression models. *Statistica Sinica*, 1603-1618.

²Windmeijer, F., Farbmacher, H., Davies, N., & Davey Smith, G. (2019). On the use of the LASSO for instrumental variables estimation with some invalid instruments. *Journal of the American Statistical Association*, 114(527), 1339-1350.

³Belloni, A., Chernozhukov, V., & Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2), 608-650.

Question 3: Comparing ridge to OLS

Subquestion a: Bias of the ridge estimator

$$\begin{aligned}
E[\hat{\beta}(\lambda)|X] &= E[(X'X + \lambda I_p)^{-1} X' y | X] \\
&= (X'X + \lambda I_p)^{-1} X' E[(X\beta + \epsilon) | X] \\
&= (X'X + \lambda I_p)^{-1} X' X \beta + (X'X + \lambda I_p)^{-1} X' E[\epsilon | X] \\
&= (X'X + \lambda I_p)^{-1} X' X \beta + (X'X + \lambda I_p)^{-1} X' \times 0 \\
&= (X'X + \lambda I_p)^{-1} X' X \beta \\
&= (X'X + \lambda I_p)^{-1} (X'X + \lambda I_p - \lambda I_p) \beta \\
&= (I_p - \lambda (X'X + \lambda I_p)^{-1}) \beta \\
&= \beta - \lambda (X'X + \lambda I_p)^{-1} \beta
\end{aligned}$$

$$\begin{aligned}
E[\hat{\beta}(\lambda)|X] - \beta &= \beta - \lambda (X'X + \lambda I_p)^{-1} \beta - \beta \\
&= -\lambda (X'X + \lambda I_p)^{-1} \beta \\
&= -\lambda W(\lambda) \beta.
\end{aligned}$$

Subquestion b: Efficiency of the ridge estimator

From the previous exercise, we have that:

$$\begin{aligned}
\hat{\beta}_\lambda &= (X'X + \lambda I_p)^{-1} X' X \hat{\beta} \\
Var[\hat{\beta}_\lambda | X] &= (X'X + \lambda I_p)^{-1} X' X Var[\hat{\beta} | X] (X'X + \lambda I_p)^{-1} \\
&= (X'X + \lambda I_p)^{-1} X' X Var[\hat{\beta} | X] X' X (X'X + \lambda I_p)^{-1} \\
&= (X'X + \lambda I_p)^{-1} X' X \sigma^2 (X'X)^{-1} X' X (X'X + \lambda I_p)^{-1} \\
&= (X'X + \lambda I_p)^{-1} (\sigma^2 X' X) (X'X + \lambda I_p)^{-1} \\
&= W(\lambda) (\sigma^2 X' X) W(\lambda)
\end{aligned}$$

When $\lambda = 0$ we are left with the OLS estimator. So for $Var(\hat{\beta}(0)|X) - Var(\hat{\beta}(\lambda)|X)$ to be positive definite we need $Var(\hat{\beta}(\lambda)|X) < Var(\hat{\beta}(0)|X)$.

$$\begin{aligned}
(X'X)^{-1} - W X' X W &= W W^{-1} (X'X)^{-1} W^{-1} W - W X' X W \\
&= W (W^{-1} (X'X)^{-1} W^{-1} - X' X) W
\end{aligned}$$

Now we will simplify the inside of the brackets.

$$\begin{aligned}
W^{-1} (X'X)^{-1} W^{-1} - X' X &= (X'X + \lambda I) (X'X)^{-1} (X'X + \lambda I)^{-1} - X' X \\
&= X' X (X'X)^{-1} X' X + X' X (X'X)^{-1} \lambda + \lambda (X'X)^{-1} X' X + \lambda^2 (X'X)^{-1} - X' X \\
&= 2\lambda I + \lambda^2 (X'X)^{-1} \\
&= \lambda (2I + \lambda (X'X)^{-1})
\end{aligned}$$

Since $(X'X)^{-1}$ is invertible it is positive definite. Therefore the whole expression is positive definite. Then we can put this back into the original expression to get:

$$\lambda W(2I + \lambda(X'X)^{-1})W$$

Which is positive definite, as the expression inside the brackets is also positive definite. Which shows that $Var(\hat{\beta}(0)|X) - Var(\hat{\beta}(\lambda)|X)$ is positive definite, making the ridge estimator more efficient than OLS estimator.

Subquestion c: Predictive mean-squared error formulation

Subquestion d: The ridge estimator dominates the OLS estimator in PMSE

Subquestion e: Convergence