

Data Analysis Project 1

MA8701

Group 5 : Yellow Submarine

15 February, 2021

In this project, we analyse a real dataset using shrinkage methods from part 1 of the MA8701 course.

Note on Open Science

To pursue the idea of reproducible research, the chosen dataset as well as the code for our analysis are publicly accessible:

- dataset: <https://data.ub.uni-muenchen.de/2/1/miete03.asc>
- code: <https://github.com/FlorianBeiser/MA8701>

The Data Set

For our project work we use the Munich Rent 2003 data set as described in <https://rdr.io/cran/LinRegInteractive/man/munichrent03.html>. The data set has 12 original covariates, where a brief introduction to these parameters is listed below (in brackets the type of the covariate is explicated), and 2053 observations are available:

- **nmqm**: rent per square meter (double)
- **wfl**: area in square meters (int)
- **rooms**: number of rooms (int)
- **bj**: year of construction (Factor)
- **bez**: district (Factor)
- **wohngut**: quality of location (int)
- **wohnbst**: high quality of location (int)
- **ww0**: hot water supply available (int)
- **zh0**: central heating (int)
- **badkach0**: tiled bathroom (int)
- **badextra**: high-quality bathroom (int)
- **kueche**: upscale kitchen equipment (int)

and the response

- **nm**: rental price (double).

The label “double” naturally stands for numerical values, “int” categorizes parameters with integer values, and “Factor” symbolize parameter taking a certain number of levels - where in contrast to integers a higher level does not necessarily mean an improvement.

Since the price per square meter **nmqm** multiplied with the area **wfl** directly gives the rental price **nm** which we define as the response in the system, it does not make sense to keep both values. Hence, we exclude **nmqm** from the data set to avoid it consuming all the significance in the coming data analysis.

Figure 1 shows the correlation between covariates ignoring the factorials and reveals that the dataset may suffer from a very light multi-collinearity.

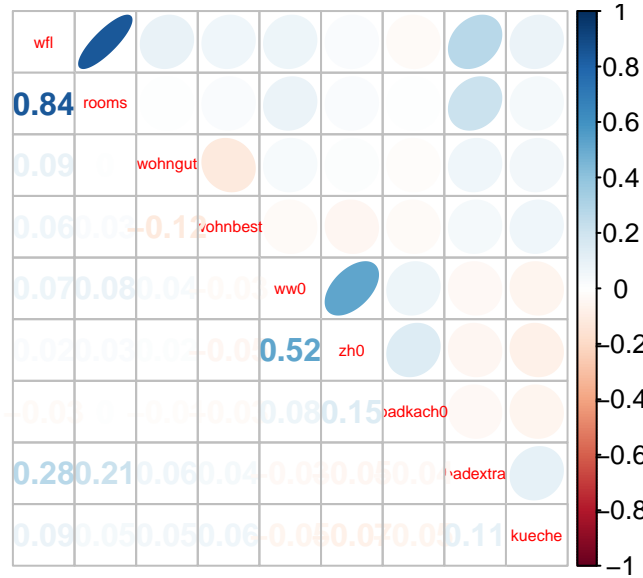


Figure 1: Exploration of multi-collinearity in the data set

However, the factorial variable **bj** and **bez** introduce 44 and 25 levels, respectively, leading to relatively unclear dependencies between the full covariate set (including factorial variables) and the response, which makes this dataset suitable for a regression analysis and the application of shrinkage methods.

Data Analysis

Subsequently, we start with a plain linear regression model as reference such that we can particularly point out the benefits of shrinkage approaches. As shrinkage methods, we employ the ridge, the lasso and the group lasso. The latter approach seems to be very well suited for our data set, as it allows to take the factorial variables as a single unit for shrinkage.

Regression

We start the data analysis with a vanilla LM regression for reference using R's internal `lm` functionality

The regression results show a lot of significant covariates. As maybe expected, the area **wfl** is strongly related to the rent price, however confusingly, the significance of different levels of the years of construction **bj** and districts **bez** varies a lot. From those both observations, it is not possible, to extract clear data analysis results, which also would match our interpretation of the problem.

Ridge Shrinkage

As first shrinkage method, we consider the ridge regression that uses Tikhonov regularisation in the model, where we utilize the `glmnet` library for its implementation in R. Since ridge introduces the additional tuning parameter λ we perform cross validation for the model selection, i.e. for the choice of the optimal λ , where we follow the advice in the ELS to choose λ as the one with minimal CV-error plus one standard deviation of the CV-error.

In Figure 2 on the left, the cross validation for different values of the regularization parameter λ is shown. On the right, the coefficients for the individual covariates are depicted against $\log \lambda$, where the optimal λ -choice is highlighted. As typical for ridge, the coefficients are shrunk towards 0, but all parameters remain positive weights. This makes the outcome still hard to interpret for our practical data set at hand.

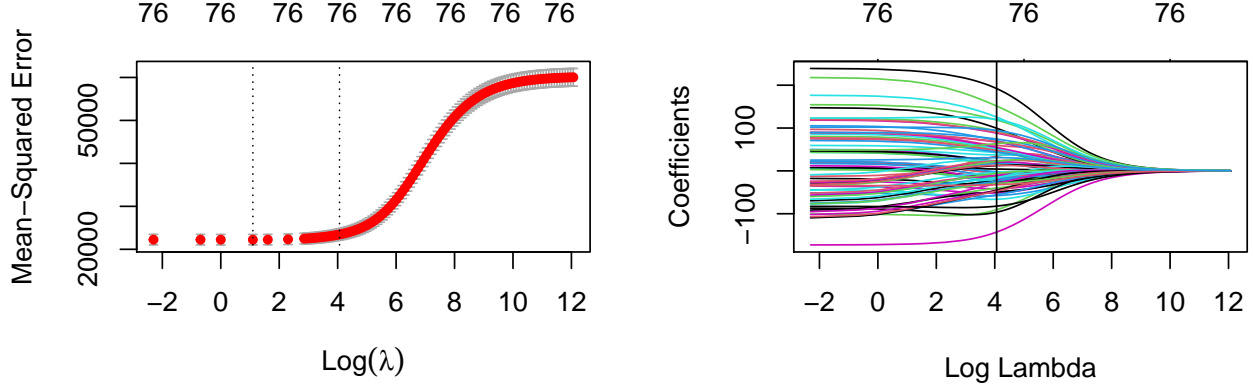


Figure 2: Model selection for ridge shrinkage

Lasso

In contrast to the previous ridge regression, the lasso adds L_1 -regularisation to the regression problem. As before, the implementation in R relies on the `glmnet` library and the hyperparameter λ is tuned as aforementioned.

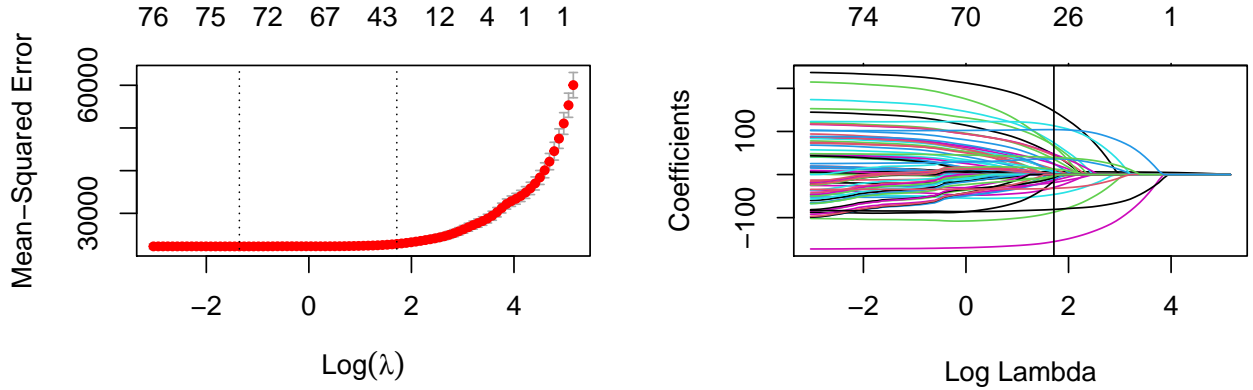


Figure 3: Model selection for lasso shrinkage

In Figure 3 on the left, the cross validation for different values of the regularization parameter λ is shown. On the right, we again see the model coefficients of the covariate set plotted against $\log \lambda$ where the optimal λ chosen by cross validation and the decision rule in ELS is highlighted. Finally and as expected, some coefficients are shrunk to 0. However for the optimal λ , some levels of the construction years `bj` and some of the districts `bez` are shrunk to 0 and other would be still significant. For the practical model problem, this is a non-intuitive behaviour.

Group lasso

The group lasso allows to gather some of the covariates and treat those with the same coefficient jointly in the L_1 -regularised problem. For the implementation in R we utilize the `grplasso` library. Naturally, we group the different levels of the factorial variables, i.e. the years of construction `bj` and the different levels for the districts `bez` together, respectively.

Again we employ the same model selection criterion via cross validation as above, but as `grplasso` does not contain a built in cv function of our knowledge, we implement it in R using the `gglasso` library. In the cross validation procedure, we can observe sudden jumps when a new group is included or shrunk from the model. Since the calculations either include all or none of the levels of the factorial variables, this can lead to a jump in the number of parameters from 9 to 53 in a single step on the lambda grid.

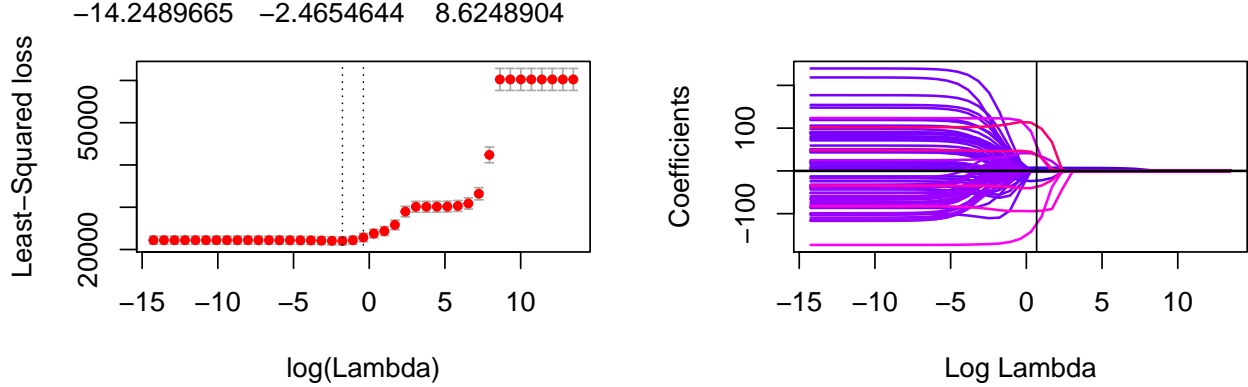


Figure 4: Model selection for group lasso

In Figure 4 on the left, the cross validation for different values of the regularization parameter λ is shown. (NB, in contrast to the previous packages, `gglasso` does not show `nzeros`, the number of active covariates per λ , in the head line above the plot) The jumps when a new group of covariates are included in the model are clearly visible. On the right, we again depict the coefficients of the covariates against the $\log \lambda$ and highlight the optimal hyperparameter. By construction, all levels of a factorial variable are shrunk to zero simultaneously. This means that either all levels remain in the model or all levels are excluded, which corresponds better to the practical interpretation of those variables.

Summary

```
## 14 x 4 sparse Matrix of class "dgCMatrix"
##               vanilla LS      ridge general lasso group lasso
## (Intercept)  162.310441  189.426083   119.818077  105.441191
## wfl          6.921638   4.218308    6.395661   7.417871
## rooms       -12.919931  34.176508      .        -23.020781
## bj1998.5     119.079298  88.120257   47.543100      .
## bj1999       47.001514  21.768019      .          .
## bez15        -85.041679 -16.891184      .          .
## bez16       -109.255107 -45.710723  -13.888054      .
## wohngut      24.911148  30.820570   36.368780   37.895305
## wohnbest     123.264686 119.790433  113.137794   64.116043
## ww0         -173.087458 -142.866851 -155.530513 -126.661696
## zh0         -82.624164 -82.303060  -80.904162  -93.773532
## badkach0     -34.489575 -32.295839  -32.094845  -37.658891
## badextra     48.627634  63.433910   39.534779   38.736540
## kueche       101.861941  94.252417   104.214296  106.457053
```

Lastly, we depict a subset of 14 parameter estimates from the methods presented in this report, with most of the grouped covariates for `bez` and `bj` removed for simplicity.

By including two estimated parameters from `bez` and `bj`, we demonstrate the non-intuitive behaviour of the general lasso. We observe that the `bez` and `bj` parameters not shrunk by the general lasso, corresponds to the highly significant covariates in the vanilla LS. Furthermore, there is a continuous shrinkage of the grouped covariates, starting from vanilla LS, shrunk in ridge and then further in general lasso, before finally resulting in zero for the grouped lasso. These trends are repeated for all parameter estimates in `bez` and `bj`, including the ones not listed above.

For the non grouped covariates however, the shrinkage is moving in a bumpy pace, by sometimes being shrunk and other times in fact increased when compared to the Vanilla LS.

@HMO How to explain this last part???? Do we need to include it? I am a true believer of the less you know the less you say

Inference

Can we afford a test set?

The data set is presumably too small to divide it into a training and a test set. We investigate this assumption by pretending an 80/20 training-test split for 100 different random splittings and keeping track of the response mean of the test set.

The variance of the test mean is more than 121, which means that depending on the split huge variations in the test data would be introduced, whereby the test data cannot be used for a proper representation of the original data set. This justifies why we cannot afford a split.

Bootstrapping

As seen before, we cannot afford a split into test and training set for our data set, therefore we use bootstrapping for inference. Bootstrap can be applied here to find the proportion of times each covariate is shrunk to zero. So it is a way of validation. Barplot demonstrates how many percentages of that coefficient being zero inside the bootstrap loop, while the boxplot gives the variability view on how dispersed or concentrated that coefficient can be.

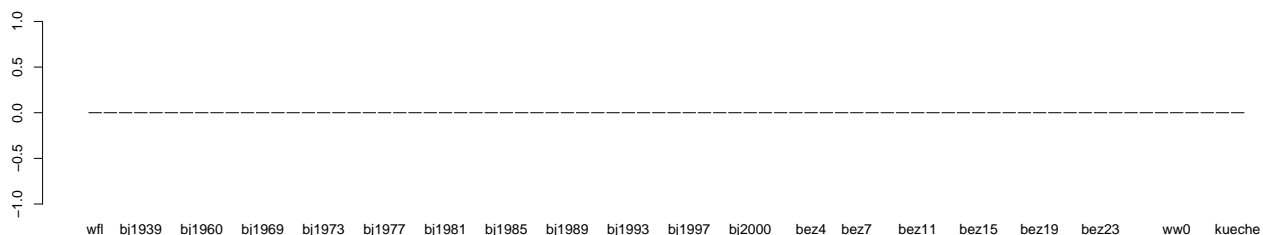


Figure 5: Barplot of the bootstrapped coefficients

From the barplot 5 one can tell that after the grouplasso implementation, the majority of the coefficients for those year covariates `bj` and those factorical covariates `bez` are not remaining zero since they are well considered in the group lasso already.

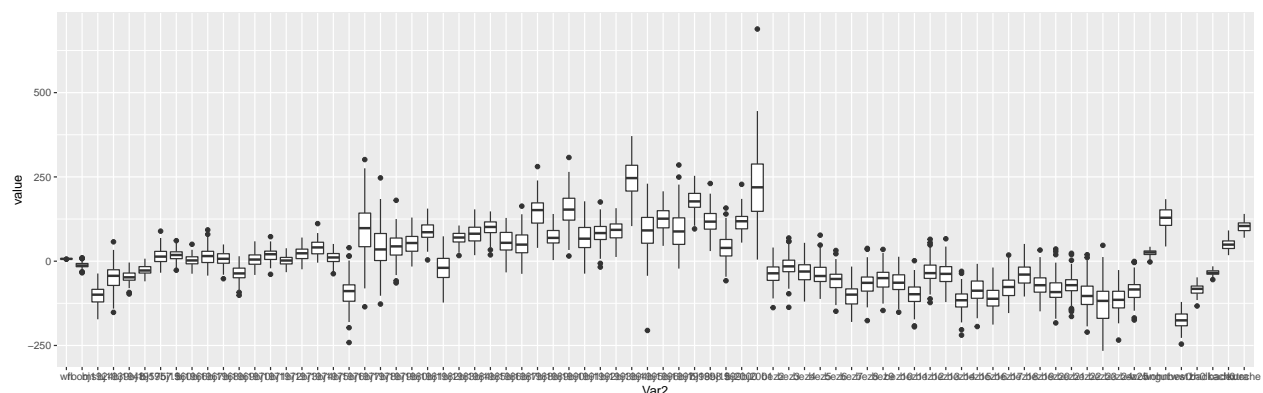


Figure 6: Boxplot of the bootstrapped coefficients

The boxplot 6 reveals the variability of those covariates that used to be zero. It also demonstrates the efficacy of using grouplasso for considering such conditions where many factorial variables are present in the dataset,

normal lasso or ridge methods tend to ignore them by shrinking them to zeros. However, the grouplasso takes into account for their significance as well. The bootstrapped boxplot does prove that the variability of some of coefficients are huge which might imply that the size of the dataset is too small to afford a test set.

So to conclude, one can tell group lasso seems working suitably for this specific case where its categorical covariates have many groups. Ridge regression does not shrink any parameters dramatically while lasso does a bit on the shrinkage, but the most shrinked contribution is from group lasso.

@YG: what happened to the grouped covariates always being shrinked to zero in the histogram you showed last meeting?? This does not make sense, Please explain this tomorrow! What are you validating? The ridge, general lasso or the grouped one? Or are you validating them all at the same time? I'm not a bootstrap expert but I am lost, so please include an explanation!

YG: I am using `lasso`, which is simple, `grplasso` always give the same coefficients, I guess if it is converged, i.e., when we choose the 1se lambda, which shrinks the right part of the coefficients. Therefore, no matter how we change the dataset, it always yields the same coefficients. But I don't know, maybe we should recheck again the implementation from the grouplasso part. I directly used the one in the group lasso section.

@YG: Using only lasso in this section contradicts to the whole narrative of this report!! You can again include it but group lasso has to be there as well!

@YG: Remember to keep the code clean before submission. YG: code is clean, git is not. YG: I think lasso did the work of visualising the purpose of small dataset, i.e., it is too various when looping through, therefore, we have a small dataset. If grouplasso is correct, we can again plot them. But agree on which form to use. But as for validation in the inference part, it suffices to show the variability. As for method validation, I think it is true as what Mette said, we can try to use different lasso variants. From the boxplot, it is not that interesting to see anything, since it is very stable or no variation.

YG: Added grouplasso bootstrap, it coincide with the conclusions now. But only full boxplot can reveal the full picture of the group lasso, thus full boxplot is shown.

Conclusion

In this project, we have chosen a practical data set which contains data on rental prices in Munich (two of the group members are practically familiar with the difficult housing situation in Munich and it was appealing to analyse this statistically). For the data analysis, the factorial variables needed special attention. A plain vanilla and ridge regression were not capable to give explainable outcomes. Likewise, the result of the lasso was contra-intuitive in the unclear handling of the factorial variables. Finally, the group lasso where a factorial variable can be arranged together leads to an interpretable shrinkage conclusion, where all variables except the factorial are selected for the optimal hyperparameter choice. From the bootstrapping validation, it can be concluded that the variability of the coefficients seem to be high enough which hinder the implementation of a test set.