

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://rdrr.io/cran/LinRegInter/active/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” symbolizes 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.

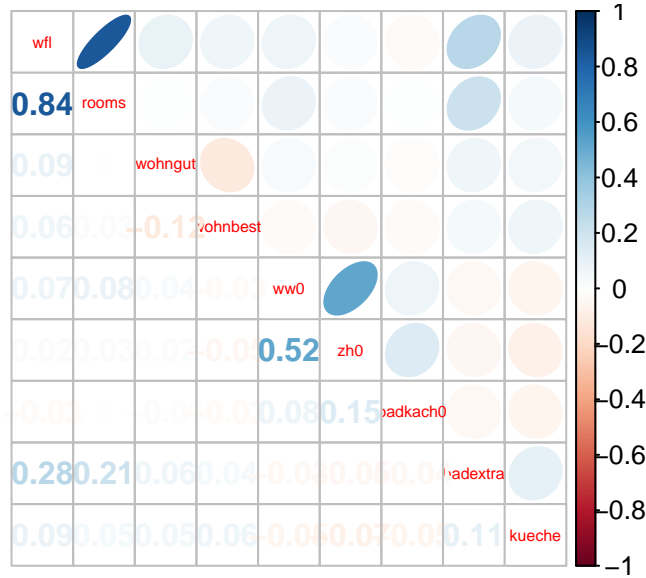


Figure 1: Exploration of multi-collinearity in the data set

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. An extract of the results can be found later, when we summarize the outcomes of all methods.

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.

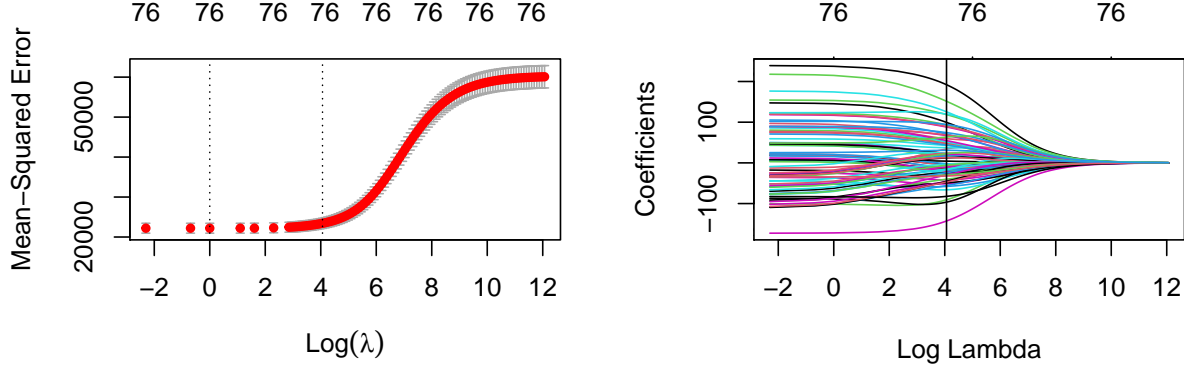


Figure 2: Model selection for ridge shrinkage

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.

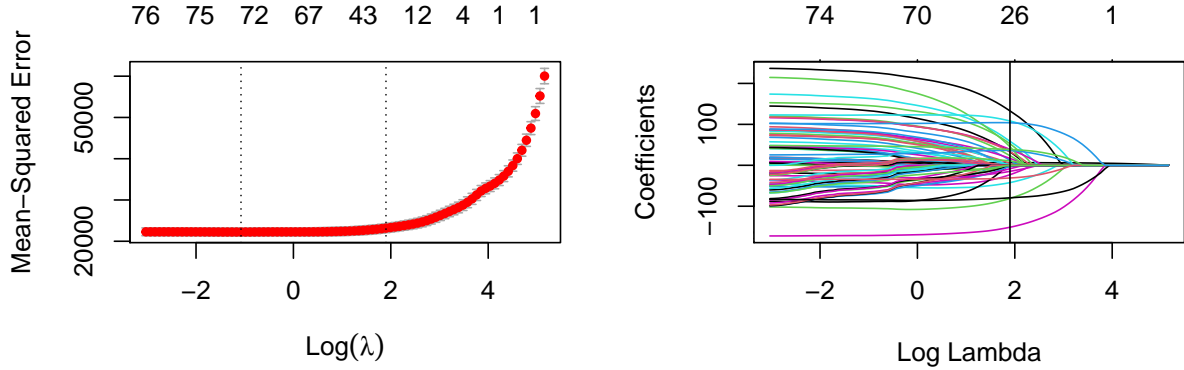


Figure 3: Model selection for lasso shrinkage

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.

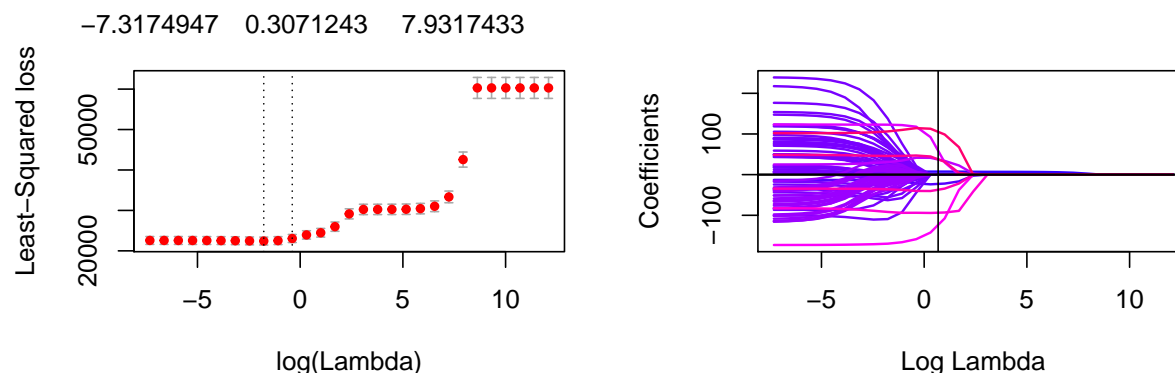


Figure 4: Model selection for group lasso

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

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.

```
## 14 x 4 sparse Matrix of class "dgCMatrix"
##               vanilla LS      ridge general lasso group lasso
## (Intercept) 162.310441 189.426083 120.614323 105.441191
## wfl         6.921638  4.218308  6.372126  7.417871
## rooms      -12.919931 34.176508  .        -23.020781
## bj1998.5    119.079298 88.120257 38.850176  .
## bj1999      47.001514 21.768019  .        .
## bez15       -85.041679 -16.891184  .        .
## bez16      -109.255107 -45.710723 -8.372467  .
## wohngut     24.911148 30.820570 36.283697 37.895305
## wohnbest    123.264686 119.790433 108.250178 64.116043
## ww0        -173.087458 -142.866851 -151.350069 -126.661696
## zh0        -82.624164 -82.303060 -79.472519 -93.773532
## badkach0    -34.489575 -32.295839 -31.000058 -37.658891
## badextra    48.627634 63.433910 38.608276 38.736540
## kueche     101.861941 94.252417 103.866594 106.457053
```

By including two estimated parameters from levels of **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. This might be explained by the fact that there is correlation between the variables, and by shrinking the grouped covariates, the non grouped covariates contain more of the variability of the data set for the optimal lambda.

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. Bootstrapping assigns measures of accuracy to samples of estimates, hence we use it here as a way of validation.

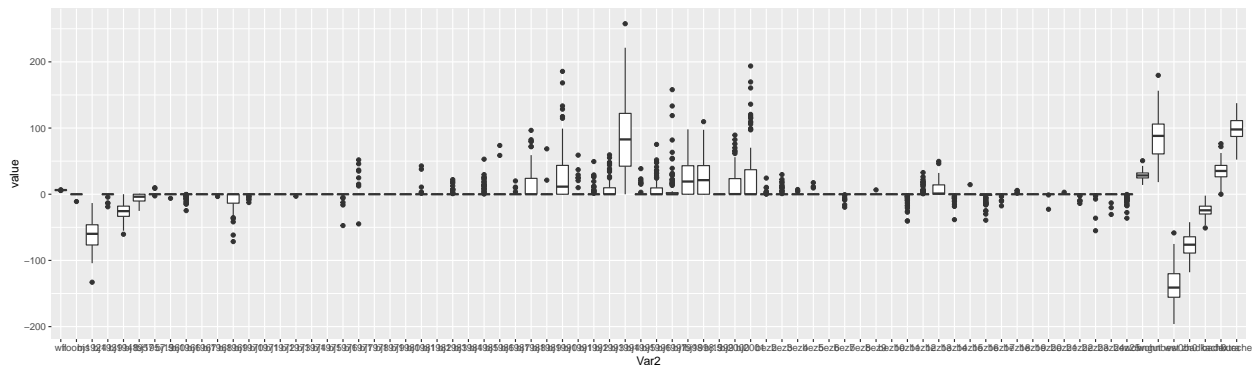


Figure 5: Boxplot of the bootstrapped general lasso coefficients

The general lasso coefficients as depicted in Fig. 5 shows a huge variability for estimated parameters corresponding to the factorial variables for their different levels. Here, we can again see that the split into training and test data would be risky. In contrast to this, the group lasso handles **bez** and **bj** as respective groups and they are always assigned the same parameter. As it can be seen in Fig. 6, the group lasso always shrinks the corresponding coefficients of the factorial variables all to zero. Note that we left out some grouped covariates since they are anyway zeros and thereby, the plot becomes more readable. Moreover, the variability of the parameters for the non-grouped variables is not much increased. It coincides with our previous argument that the group lasso is particularly suited for our data set with its factorial variables.

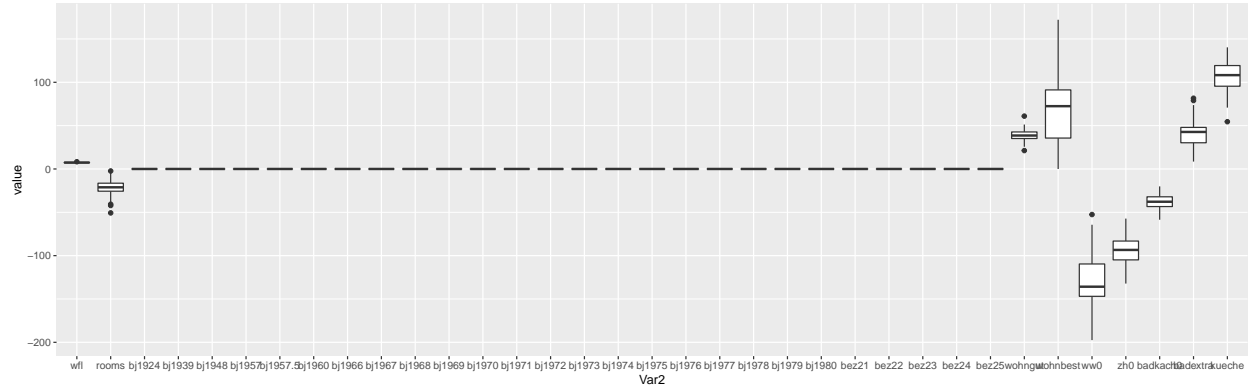


Figure 6: Boxplot of the bootstrapped group lasso selected coefficients

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. The bootstrapping validates our assumption that group lasso the preferred shrinkage method leading to the most interpretable and robust results.