Lecture 15: High-dimensional inference

Rebecka Jörnsten, Mathematical Sciences

MSA220/MVE441 Statistical Learning for Big Data

13th May 2022

Lasso - recap

Lasso as a selection mechanism

$$\widehat{\boldsymbol{\beta}}_{\text{lasso}}(\lambda) = \underset{\boldsymbol{\beta}}{\arg\min} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$$

- Smallest q in penalty such that constraint is still convex
- Produces **sparse solutions** (many coefficients exactly equal to zero) and therefore performs **feature selection**

Lasso as a selection mechanism

In lecture we talked about the selection properties of the lasso.

- If X contain correlated features we know that selection can become unstable and pivot between the correlated elements
  - elastic net
  - group lasso
  - filtering
- If the correlations are not too large between the true predictors and the predictors not in the true model, and your penalty is chosen appropriately, lasso can be shown to be consistent in terms of model selection

Lasso as a selection mechanism

Still, as you may already have noticed from your work on **Project 3**, selecting a good penalty factors can be far from trivial.

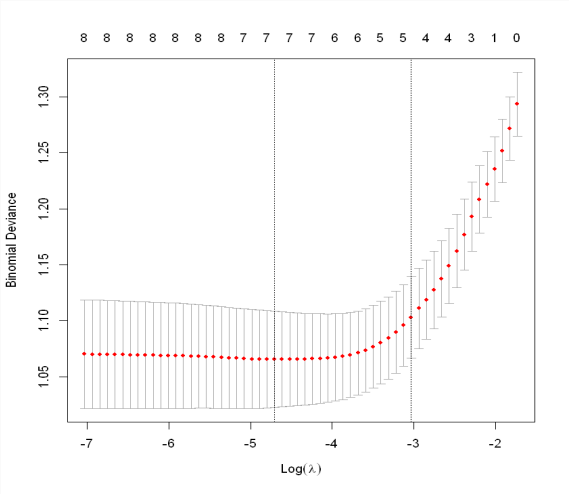
- Cross-validation:
  - Use the λ corresponding to the minimum cross-validation error: λ_{min}
  - Use the λ with cross-validation error within 1 SE of the minimum with the most sparse solutions: λ_{1SE}
- Still, once you apply this penalty to the full data set you arrive at a model with selected features and non-selected features and corresponding estimates
- What if you want p-values and confidence intervals in your analysis report?

Lasso as a selection mechanism

Can't we just refit the selected model to the data with OLS and report those p-values?
No!

- You already used the data to select the model and if you refit using the same data you are subject to **Selection bias**
- Your p-values will be over-optimistic
- Also, you may still have $p_{selected} > n$

Lasso as a selection mechanism



Re-fit of the selected model

Example: the South African heart disease data

glm fit

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.0760913  1.3404862  -5.279  1.3e-07 ***
sbp          0.0065040  0.0057304   1.135 0.256374
tobacco      0.0793764  0.0266028   2.984 0.002847 **
ldl          0.1739239  0.0596617   2.915 0.003555 **
adiposity    0.0185866  0.0292894   0.635 0.525700
famhist      0.9253704  0.2278940   4.061  4.9e-05 ***
typea        0.0395950  0.0123202   3.214 0.001310 **
obesity     -0.0629099  0.0442477  -1.422 0.155095
alcohol      0.0001217  0.0044832   0.027 0.978350
age          0.0452253  0.0121298   3.728 0.000193 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 472.14  on 452  degrees of freedom
AIC: 492.14

Number of Fisher Scoring iterations: 5
```

glm refit on selected

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.35462    0.96804  -7.597 3.02e-14 ***
tobacco      0.08038    0.02588   3.106  0.00190 **
ldl          0.16199    0.05497   2.947  0.00321 **
famhist      0.90818    0.22576   4.023 5.75e-05 ***
typea        0.03712    0.01217   3.051  0.00228 **
age          0.05046    0.01021   4.944 7.65e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 475.69  on 456  degrees of freedom
AIC: 487.69

Number of Fisher Scoring iterations: 5
```

High-dimensional inference

High-dimensional inference

In this lecture we will talk about a few approaches for inference for lasso type methods

- Re-sampling based inference
- Bias-correction

High-dimensional inference

- L1-penalized modeling has become enormously popular for high-dimensional problems
- We get model selection, and as we have seen in previous lectures, pretty good point estimates since the bias is constrained to λ
- But when we do low-dimensional modeling we usually don't feel very satisfied with just point estimates
- We want confidence intervals and p-values!

High-dimensional inference

- What are the obstacles for obtaining p-values and confidence intervals?
- Highly non-standard setting when $p > n$
- the distribution of lasso-solutions, by construction, has a point-mass at 0 and this makes bootstrapping to get standard error estimates difficult

Sample-splitting

Wasserman and Roeder (2009) proposed the following approach to obtain p-values

- Split the data in two sets
- Use set 1 to perform modelselection via e.g. lasso
- Use set 2 to evaluate p-values for the non-zero coefficients. This is done by running LS using only the selected variables in the model.
- For the variables not selected in set 1, set p-value to 1.

The p-values are valid because we didn't reuse the same data for selection and p-value computation.
Moreover, if we want to **compute adjusted p-values** that take into account multiple testing we **only have to correct by selected set of variables, not all** p.

Sample-splitting

Example: the South African heart disease data

Result from one 50-50 split
Model summary from the test set

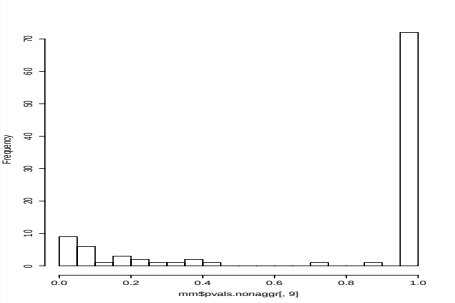
```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.18526    1.42228  -5.755 8.66e-09 ***
tobacco      0.10105    0.03902   2.590   0.0096 **
famhist      1.36506    0.33562   4.067 4.76e-05 ***
typea        0.04408    0.01772   2.487   0.0129 *
age          0.06179    0.01586   3.897 9.75e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 295.44  on 230  degrees of freedom
Residual deviance: 219.26  on 226  degrees of freedom
AIC: 229.26

Number of Fisher Scoring iterations: 5
```

- Price: you are doing model selection on a smaller data set
- Might lead to a smaller model
- What happens if you split the data again?

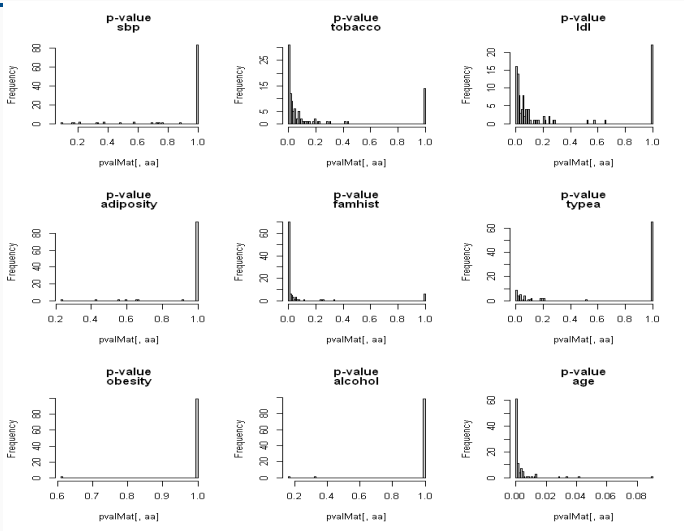
Sample-splitting

Drawback with the procedure

- Sensitive to the split so the pvalues are not reproducible
- ”p-value lottery”
- Different splits leads to widely different p-values!



P-value lottery



Multi sample-splitting

To overcome the p-value lottery we perform several random splits of data (Meinhausen et al, 2009)

- Repeat B times: split data into set 1 and set 2
- Use set 1 for selection of variables
- Use set 2 to compute p-values and correct for multiple testing using the number of selected features in set 1
- Aggregate the p-values

Hm? How to we combine B p-values (like those from the histogram above) to one final p-value?

Multi sample-splitting

The p-value estimates are not independent because the data splits overlap.

- We can use the median p-value
- Or any other quantile
- Search for the best quantile - minimum p-values once adjusted for search for the optimal thresshold.
- Note: the sample splits are dependent so you need to adjust for this and the optimal search in the aggregation.
- Implemented in package *hdi()* and see also the class demo.

Stability selection

Stability selection

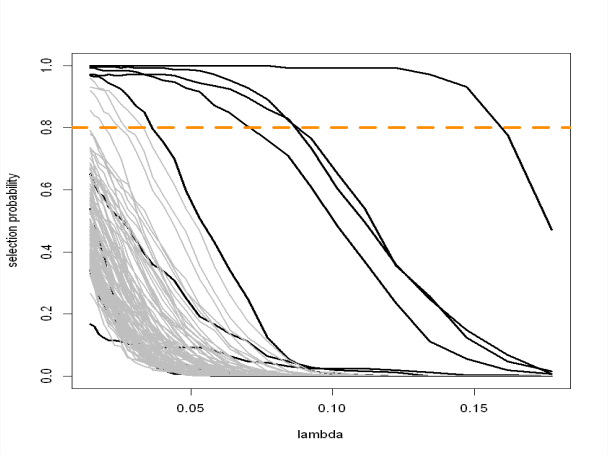
We have noticed that lasso selection can be quite unstable - that is, you obtain very different optimal λ or number and set of selected features of different splits of the data.

Can we explore this to obtain a better selection procedure?

- Consider a range Λ of tuning parameters
- Run lasso selection on multiple splits of data using this range
- For each $\lambda \in \Lambda$, record the selection proportion for each feature j: Π_j^λ - the so-called *stability paths*

Stability selection

Stability paths with a threshold



Stability selection

- For each $\lambda \in \Lambda$, record the selection proportion for each feature j: Π_j^λ - the so-called *stability paths*
- The stable set $S^{stable} = \{j : max_{\lambda \in \Lambda}(\Pi_j^\lambda) \geq \pi_{thr}\}$ for some threshold π_{thr}
- Let q_Λ denote the average model size selected across the multiple runs and range Λ and let V denote the number of falsely selected features.
- Meinshausen and Buhlmann (2010) show that

$$E(V) \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda^2}{p}$$

where p is the number of features we select from.

Stability selection

- Meinshausen and Buhlmann (2010) show that

$$E(V) \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda^2}{p}$$

  where p is the number of features we select from.
- Now you can set a threshold and a tuning parameter range Λ and obtain an estimate of the expected number of false discoveries
- Alternatively, pick a threshold π_{thr} and an acceptable number of false positives and derive the tuning parameter range Λ (size models allowed, e.g. something like $\sim \sqrt{p}$).

Stability selection and extensions thereof are quite popular for network modeling based on lasso-methods.

De-biasing the lasso

De-sparsified lasso

As you saw from the splitting procedures, you do sacrifice some data to find the p-values for sparse estimators.
Zhang and Zhang (2014) proposed the de-sparsified lasso to come up with p-values in a high-dimensional setting.

- We start with the $p < n$ setting
- We are interested in the j-th coefficient estimate
- It turns out we can obtain the LS estimate as follows

$$\hat{\beta}_j^{LS} = \frac{y' Z_j}{X_j' Z_j}$$

  where Z_j is the residual if you run a regression of X_j on all the other Xs!
- Proof: go through the OLS solutions and use matrix block inverses.

De-sparsified lasso

Write out the true model

$$y = \sum_{j=1}^{J} X_j \beta_j^* + \eta$$

where β^* are the true coefficient values

- If we plug this into the estimate $\hat{\beta}_j^{LS} = \frac{y' Z_j}{X_j' Z_j}$ we see

  $$\frac{y' Z_j}{X_j' Z_j} = \beta_j^* + \sum_{k \neq j} \beta_k^* \frac{X_k' Z_j}{X_j' Z_j} + \frac{\eta' Z_j}{X_j' Z_j} When we have run regression with LS the residuals$$

  Z_j are orthogonal to the other variables X_k and so we see that second term on the right hand side is 0.

- What happens when $p > n$?
- Then this doesn't work since residuals Z_j are 0

De-sparsified lasso

Idea (Zhang and Zhang, 2014): Use a *regularized regression* of X_j on the other Xs!

- If we plug this into the estimate $\hat{\beta}_j^{LS} = \frac{y'Z_j}{X_j'Z_j}$ we see

$$\frac{y'Z_j}{X_j'Z_j} = \beta_j^* + \sum_{k \neq j} \beta_k^* \frac{X_k'Z_j}{X_j'Z_j} + \frac{\eta'Z_j}{X_j'Z_j}$$

- Now term 2 does not go away and therefore we now have a biased estimate of β_j^*
- Bias correction

$$\hat{\beta}_j = \frac{y'Z_j}{X_j'Z_j} - \sum_{k \neq j} \hat{\beta}_k \frac{X_k'Z_j}{X_j'Z_j}$$

where we use the lasso-estimates $\hat{\beta}_k$

De-sparsified lasso

Zhang and Zhang (2014) and van de Geer at al (2014) has derived the distribution for the bias-corrected estimate as

$$\sqrt{n}(\hat{\beta} - \beta^*) \sim N_p(0, W)$$

- Since from above we have

$$\sqrt{n}(\hat{\beta}_j - \beta_k^*) = \frac{\sqrt{n}\eta' Z_j}{n^{-1} X_j' Z_j} + R$$

  where R can be shown to be neglible under sparsity assumptions on β^* and structure on X
- Then we can derive the distribution variance W from the term involving η as

$$W_{jk} = \sigma_\eta \frac{Z_j' Z_k}{(X_j' Z_j)(X_k' Z_k)}$$

bias-corrected ridge regression

Another proposal by Buhlmann (2013) uses a bias-corrected ridge estimate

- Here we start with the ridge regression estimate
- Then we perform bias-correction using lasso-estimates
- Buhlmann (2013) derive the sampling distribution for the bias-corrected estimates
- And now we can compute p-values for every β!!!
- Computationally cheaper than the de-sparsified lasso
- Tuning parameters need to selected - CV can be used or other criteria (see journal paper)
- package *hdi()*

Correlated variables

In practice, we often have highly correlated variables in our data sets. This was the motivation for group selection in elastic net or group lasso.
When we have correlated variables this translates to higher estimation variance within the group, wider confidence intervals and lower power of detection.

- Group testing is one solution
- We can group the variables together based on their pairwise correlations, e.g. via hierarchical clustering
- We can then compute p-values for each group.
- How do we we generate group-p-values?
- In de-sparsified lasso and ridge we adjust the individual p-values by the number of tests performed (p) and then use the minimum adjusted p-value within the group for group decisions.

Correlated variables

Meinhausen (2013) proposes a multi-split testing of groups as follows.

- We use multi-sample splitting to construct confidence intervals for the l1-norm of a group.
- If the lower bound of this confidence interval is larger than 0, we reject the null-hypothesis for this group.
- *hdi()* package illustrates the group tests with a hierarchical tree (see paper)

Take-home message

- When $n < p$ we use regularized methods for feature selection
- We have to be careful about how we obtain p-values for selected features - we can't reuse the same data used for selection to compute p-values
- Multi-sample splitting is very intuitive and quite general - drawback that you sacrifice some data in the splitting procedure
- de-sparsified or de-biased methods have been developed for lasso selection and do not require data splits