Lecture 9: Regularized/penalized regression

Felix Held, Mathematical Sciences

MSA220/MVE440 Statistical Learning for Big Data

15th April 2019

CHALMERS UNIVERSITY OF TECHNOLOGY | UNIVERSITY OF GOTHENBURG

Revisited: Expectation-Maximization (I)

New target function: Maximize

$$\log(p(\mathbf{X}|\theta)) = \mathbb{E}_{q(\mathbf{Z})}\left[\log\frac{p(\mathbf{X},\mathbf{Z}|\theta)}{q(\mathbf{Z})}\right] - \mathbb{E}_{q(\mathbf{Z})}\left[\log\frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})}\right]$$

with respect to $q(\mathbf{Z})$ and θ

Note:

- The left hand side is independent of $q(\mathbf{Z})$
- The difference on the right hand side has always the same value, **irrespective of the chosen $q(\mathbf{Z})$.**

Choosing $q(\mathbf{Z})$ is therefore a trade-off between

$$\mathbb{E}_{q(\mathbf{Z})}\left[\log\frac{p(\mathbf{X},\mathbf{Z}|\theta)}{q(\mathbf{Z})}\right] \quad \text{and} \quad \mathbb{E}_{q(\mathbf{Z})}\left[\log\frac{p(\mathbf{Z}|\mathbf{X},\theta)}{q(\mathbf{Z})}\right]$$

Revisited: Expectation-Maximization (II)

1. **Expectation step:** For given parameters $\theta^{(m)}$ the density $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})$ **minimizes the second term** and thereby **maximizes the first one**. Set
$$Q(\theta, \theta^{(m)}) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\theta^{(m)})}\left[\log \frac{p(\mathbf{X}, \mathbf{Z}|\theta)}{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}\right]$$
2. **Maximization step:** Maximize the first term with
$$\theta^{(m+1)} = \arg\max_{\theta} Q(\theta, \theta^{(m)})$$

Note: Since

$$\mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\theta^{(m)})}\left[\log \frac{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}\right] = 0$$

it follows that

$$\log(p(\mathbf{X}|\theta^{(m)})) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\theta^{(m)})}\left[\log \frac{p(\mathbf{X}, \mathbf{Z}|\theta^{(m)})}{p(\mathbf{Z}|\mathbf{X}, \theta^{(m)})}\right]$$

Regularized/penalized regression

Remember ordinary least-squares (OLS)

Consider the model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

- $\mathbf{y} \in \mathbb{R}^n$ is the **outcome**, $\mathbf{X} \in \mathbb{R}^{n\times(p+1)}$ is the **design matrix**, $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$ are the **regression coefficients**, and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the **additive error**
- **Five basic assumptions** have to be checked

  Underlying relationship is linear (1)

  Zero mean (2), uncorrelated (3) errors with constant variance (4) which are (roughly) normally distributed (5)
- **Centring** ($\frac{1}{n}\sum_{l=1}^{n} x_{lj} = 0$) and **standardisation** ($\frac{1}{n}\sum_{l=1}^{n} x_{lj}^2 = 1$) of predictors simplifies interpretation
- **Centring** the outcome ($\frac{1}{n}\sum_{l=1}^{n} y_l = 0$) and features removes the need to estimate the intercept

Feature selection as motivation

Analytical solution exists when $\mathbf{X}^T\mathbf{X}$ is invertible

$$\hat{\boldsymbol{\beta}}_{\mathrm{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

This can be unstable or fail in case of

- **high correlation** between predictors, or
- if $p > n$.

Solutions: Regularisation or **feature selection**

Filtering for feature selection

- Choose features through pre-processing
  - Features with maximum variance
  - Use only the first k PCA components
- Examples of other useful measures
  - Use a univariate criterion, e.g. **F-score:** Features that correlate most with the response
  - **Mutual Information:** Reduction in uncertainty about $\mathbf{x}$ after observing y
  - **Variable importance:** Determine variable importance with random forests
- **Summary**
  - **Pro:** Fast and easy
  - **Con:** Filtering mostly operates on single features and is not geared towards a certain method
  - Care with cross-validation and multiple testing necessary
- Filtering is often more of a pre-processing step and less of a proper feature selection step

Wrapping for feature selection

- **Idea:** Determine the best set of features by fitting models of different complexity and comparing their performance
- **Best subset selection:** Try all possible (**exponentially many**) subsets of features and compare model performance with e.g. cross-validation
- **Forward selection:** Start with just an intercept and add in each step the variable that improves fit the most (**greedy algorithm**)
- **Backward selection:** Start with all variables included and then remove sequentially the one with the least impact (**greedy algorithm**)
- As discreet procedures, all of these methods **exhibit high variance** (small changes could lead to different predictors being selected, resulting in a potentially very different model)

Embedding for feature selection

- **Embed/include** the feature selection into the model estimation procedure
- Ideally, penalization on the number of included features

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} \mathbb{1}(\beta_j \neq 0)$$

  However, **discrete optimization problems** are hard to solve
- **Softer regularisation methods** can help

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_q^q$$

  where λ is a tuning parameter and $q \geq 1$ or $q = \infty$.

Constrained regression

The optimization problem

$$\arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 \quad \text{subject to} \quad ||\boldsymbol{\beta}||_q^q \leq t$$

for $q > 0$ is equivalent to

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_q^q$$

when $q \geq 1$. This is the **Lagrangian** of the constrained problem.

- Clear when $q > 1$: Convex constraint + target function and both are differentiable
- Harder to prove for $q = 1$, but possible (e.g. with subgradients)

Ridge regression

For $q = 2$ the constrained problem is **ridge regression**

$$\hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda) = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2$$

where $||\boldsymbol{\beta}||_2^2 = \sum_{j=1}^{p} \beta_j^2$.

An **analytical solution** exists if $\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p$ is invertible

$$\hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda) = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{y}$$

If $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$, then

$$\hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda) = \frac{\hat{\boldsymbol{\beta}}_{\mathrm{OLS}}}{1+\lambda},$$

i.e. $\hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda)$ is **biased** but has **lower variance**.

SVD and ridge regression

Recall: The SVD of a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ was

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$$

The analytical solution for ridge regression becomes ($n \geq p$)

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{\text{ridge}}(\lambda) &= (\mathbf{X}^T\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{y} \\ &= (\mathbf{V}\mathbf{D}^2\mathbf{V}^T + \lambda \mathbf{I}_p)^{-1}\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{y} \\ &= \mathbf{V}(\mathbf{D}^2 + \lambda \mathbf{I}_p)^{-1}\mathbf{D}\mathbf{U}^T\mathbf{y} \\ &= \sum_{j=1}^{p} \frac{d_j}{d_j^2 + \lambda}\mathbf{v}_j\mathbf{u}_j^T\mathbf{y} \end{aligned}$$

Ridge regression **acts most** on principal components with **lower eigenvalues**, e.g. in presence of correlation between features.

Effective degrees of freedom

Recall the **hat matrix** $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ in OLS. The trace of $\mathbf{H}$

$$\mathrm{tr}(H) = \mathrm{tr}(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T) = \mathrm{tr}(\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}) = \mathrm{tr}(\mathbf{I}_p) = p$$

is equal to the trace of $\widehat{\mathbf{\Sigma}}$ and the **degrees of freedom** for the regression coefficients.

In analogy define for ridge regression

$$\mathbf{H}(\lambda) := \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T$$

and

$$\mathrm{df}(\lambda) := \mathrm{tr}(\mathbf{H}(\lambda)) = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda},$$

the **effective degrees of freedom**.

Lasso regression

For $q = 1$ the constrained problem is known as the **lasso**

$$\hat{\boldsymbol{\beta}}_{\text{ridge}}(\lambda) = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1$$

- Smallest q in penalty such that constraint is still convex
- Performs **feature selection**

Intuition for the penalties (I)

Assume the OLS solution $\boldsymbol{\beta}_{\text{OLS}}$ exists and set

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}_{\text{OLS}}$$

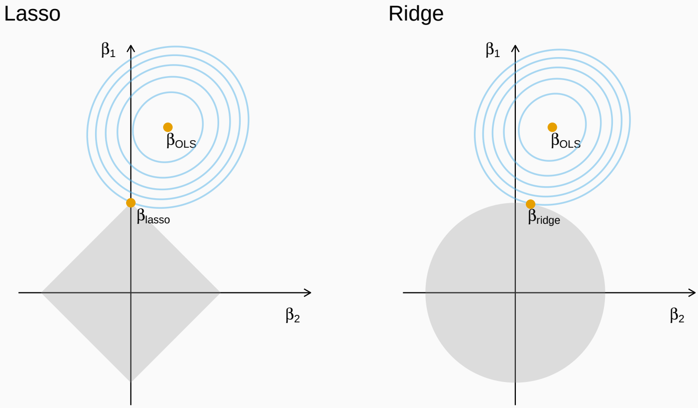
it follows for the **residual sum of squares (RSS)** that

$$\begin{aligned} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 &= ||(\mathbf{X}\boldsymbol{\beta}_{\text{OLS}} + \mathbf{r}) - \mathbf{X}\boldsymbol{\beta}||_2^2 \\ &= ||(\mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{OLS}}) - \mathbf{r}||_2^2 \\ &= (\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{OLS}})^T \mathbf{X}^T \mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{OLS}}) - 2\mathbf{r}^T \mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_{\text{OLS}}) + \mathbf{r}^T \mathbf{r} \end{aligned}$$

which is an **ellipse** (at least in 2D) centred on $\boldsymbol{\beta}_{\text{OLS}}$.

Intuition for the penalties (II)

The least squares RSS is minimized for $\boldsymbol{\beta}_{\mathrm{OLS}}$. If a constraint is added ($\|\boldsymbol{\beta}\|_q^q \leq t$) then the RSS is minimized by the closest $\boldsymbol{\beta}$ possible that fulfills the constraint.



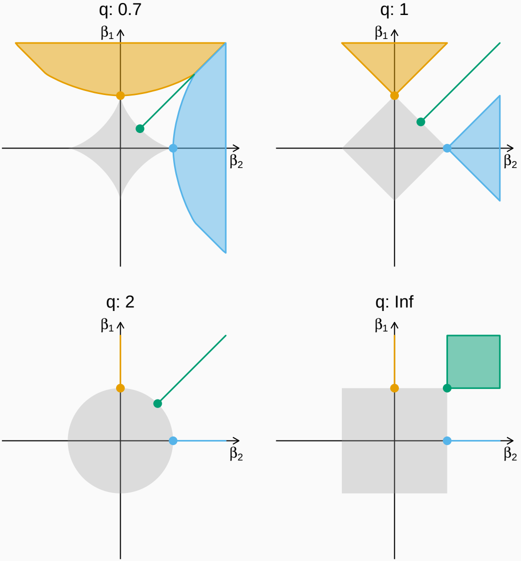
Lasso

Ridge

The blue lines are the contour lines for the RSS.

Intuition for the penalties (III)

Depending on q the different constraints lead to different solutions. If $\boldsymbol{\beta}_{\mathrm{OLS}}$ is in one of the coloured areas or on a line, the constrained solution will be at the corresponding dot.



Computational aspects of the Lasso (I)

What estimates does the lasso produce?

Target function

$$\arg\min_{\boldsymbol{\beta}} \frac{1}{2}||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1$$

Special case: $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$. Then

$$\frac{1}{2}||\mathbf{y}-\mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 = \frac{1}{2}\mathbf{y}^T\mathbf{y} - \underbrace{\mathbf{y}^T\mathbf{X}}_{=\boldsymbol{\beta}_{\mathrm{OLS}}^T}\boldsymbol{\beta} + \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + \lambda||\boldsymbol{\beta}||_1 = \mathrm{g}(\boldsymbol{\beta})$$

How do we find the solution $\hat{\boldsymbol{\beta}}$ in presence of the **non-differentiable** penalisation $||\boldsymbol{\beta}||_1$?

Computational aspects of the Lasso (II)

For $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$ the target function can be written as

$$\arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{p} -\beta_{\mathrm{OLS},j}\beta_j + \frac{1}{2}\beta_j^2 + \lambda|\beta_j|$$

This results in p **uncoupled** optimization problems.

- **If** $\beta_{\mathrm{OLS},j} > 0$, then $\beta_j > 0$ to minimize the target
- **If** $\beta_{\mathrm{OLS},j} \leq 0$, then $\beta_j \leq 0$

Each case results in

$$\widehat{\beta}_j = \mathrm{sign}(\beta_{\mathrm{OLS},j})(|\beta_{\mathrm{OLS},j}| - \lambda)_+ = \mathrm{ST}(\beta_{\mathrm{OLS},j}, \lambda),$$

where

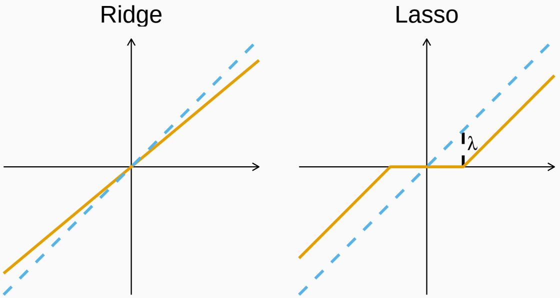
$$x_+ = \begin{cases} x & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

and ST is the **soft-thresholding operator**

Relation to OLS estimates

Both ridge regression and the lasso estimates can be written as functions of $\boldsymbol{\beta}_{\mathrm{OLS}}$ if $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$.

$$\beta_{\mathrm{ridge},j} = \frac{\beta_{\mathrm{OLS},j}}{1+\lambda} \quad \text{and} \quad \widehat{\beta}_j = \mathrm{sign}(\beta_{\mathrm{OLS},j})(|\beta_{\mathrm{OLS},j}| - \lambda)_+$$



Visualisation of the transformations applied to the OLS estimates.

Shrinkage

When λ is fixed, the **shrinkage** of the lasso estimate $\boldsymbol{\beta}_{\text{lasso}}(\lambda)$ compared to the OLS estimate $\boldsymbol{\beta}_{\text{OLS}}$ is defined as

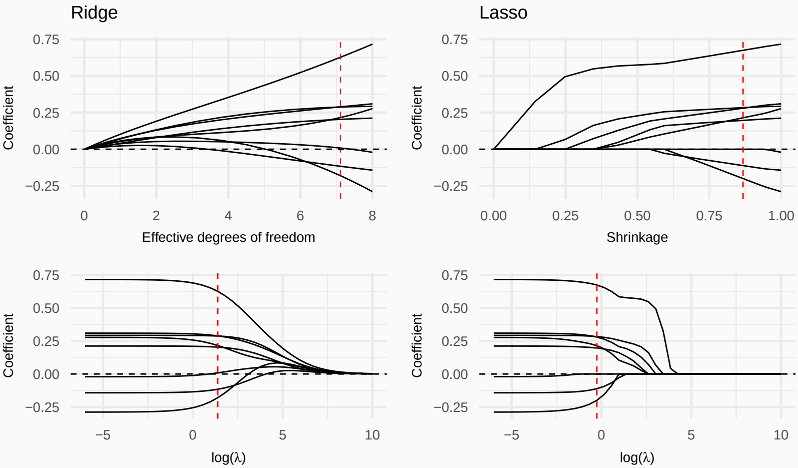
$$s(\lambda) = \frac{||\boldsymbol{\beta}_{\text{lasso}}(\lambda)||_1}{||\boldsymbol{\beta}_{\text{OLS}}||_1}$$

Note: $s(\lambda) \in [0, 1]$ with $s(\lambda) \to 0$ for increasing λ and $s(\lambda) = 1$ if $\lambda = 0$

A regularisation path

Prostate cancer dataset $(n = 67,\ p = 8)$

Red dashed lines indicate the λ selected by cross-validation



Notes on the lasso

- In the general case, i.e. $\mathbf{X}^T\mathbf{X} \neq \mathbf{I}_p$, there is no explicit solution.
- Numerical solution possible, e.g. with **coordinate descent**
- As for ridge regression, **estimates are biased**
- But
    - **Asymptotic consistency:** If $\lambda = o(n)$ then $\boldsymbol{\beta}_{\text{lasso}} \rightarrow \boldsymbol{\beta}_{\text{true}}$ for $n \rightarrow \infty$
    - **Model selection consistency:** If $\lambda \propto n^{1/2}$, then there is a non-zero probability of identifying the true model
    - **Degrees of freedom:** The degrees of freedom are equal to the number of non-zero coefficients

Potential caveats of the lasso (I)

- **Sparsity of the true model:**
  - The lasso only works if the data is generated from a sparse process.
  - However, a dense process with many variables and not enough data or high correlation between predictors can be unidentifiable either way
- **Correlations:** Many non-relevant variables correlated with relevant variables can lead to the selection of the wrong model, even for large n
- **Irrepresentable condition:** Split $\mathbf{X}$ such that $\mathbf{X}_1$ contains all **relevant variables** and $\mathbf{X}_2$ contains all **irrelevant variables**. If

  $$|(\mathbf{X}_2^T\mathbf{X}_1)^{-1}(\mathbf{X}_1^T\mathbf{X}_1)| < 1 - \eta$$

  for some $\eta > 0$ then the lasso is (almost) guaranteed to pick the true model

Potential caveats of the lasso (II)

In practice, both the **sparsity of the true model** and the **irrepresentable condition** cannot be checked.

- Assumptions and domain knowledge have to be used

Take-home message

- Filtering and wrapping methods useful for feature selection in practice but can be unprincipled or have high variance
- Penalisation gives stability to regression
- The lasso performs variable selection and variance stabilisation at the same time