Lecture 11: Data representations

Felix Held, Mathematical Sciences

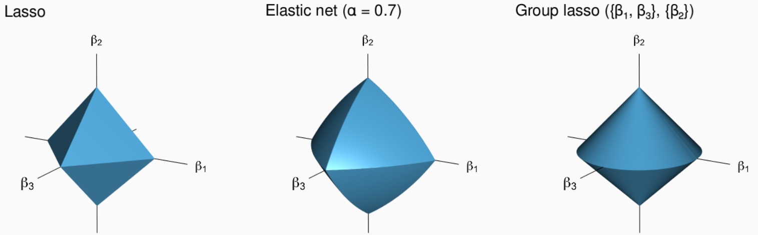
MSA220/MVE440 Statistical Learning for Big Data

3rd May 2019

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Recap: Elnet and group lasso



Lasso

Elastic net ($\alpha = 0.7$)

Group lasso ($\{\beta_1, \beta_3\}, \{\beta_2\}$)

- The lasso sets variables exactly to zero either on a corner (all but one) or along an edge (fewer).
- The elastic net similarly sets variables exactly to zero on a corner or along an edge. In addition, the curved edges encourage coefficients to be closer together.
- The group lasso has actual information about groups of variables. It encourages whole groups to be zero simultaneously. Within a group, it encourages the coefficients to be as similar as possible.

The lasso and bias (I)

One problem with penalisation methods is the **bias** that is introduced by shrinkage.

Remember: For orthogonal predictors, i.e. $\mathbf{X}^T\mathbf{X} = \mathbf{I}_p$ we have

$$\widehat{\beta}_{\text{lasso},j} = \mathrm{ST}(\widehat{\beta}_{\text{OLS},j}, \lambda)$$

The **least squares estimates are unbiased** (i.e. $\mathbb{E}[\boldsymbol{\beta}_{\text{OLS}}] = \boldsymbol{\beta}_{\text{true}}$) and therefore any non-linear transformation (like soft-thresholding) creates biased estimates.

Shrinkage is good for variable/model selection but can decrease predictive performance.

Ideal case: **Oracle procedure** (Fan and Li, 2001) Assume the true subset of non-zero coefficients is $\mathcal{A} = \{j : \beta_{\text{true}} = 0\}$

1. Identifies the right variables, i.e. $\{j : \widehat{\beta}_j \neq 0\} = \mathcal{A}$
2. Optimal estimation rate: $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_{\text{true}}) \xrightarrow{d} N(\mathbf{0}, \boldsymbol{\Sigma})$

The lasso and bias (II)

- We saw that if the **irrepresentable condition** is fulfilled (i.e. correlation between relevant and unrelevant predictors is small), then the lasso does a good job in uncovering the true subset of variables
- The lasso however introduces bias that will not vanish asymptotically. It therefore produces inconsistent estimates (i.e. $\hat{\boldsymbol{\beta}}_{\text{lasso}} \nrightarrow \boldsymbol{\beta}_{\text{true}}$ for $n \to \infty$)
- **Solution: Different penalty function?** An ideal penalty would be
  - singular at zero, leading to sparsity
  - no penalty for large coefficients, leading to unbiased estimates away from zero
  - convex and differentiable
- The **smoothly clipped absolute deviation (SCAD)** penalty combines all these **apart from convexity**

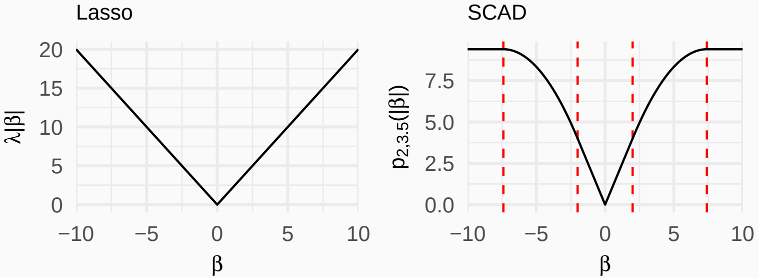
Smoothly clipped absolute deviation (SCAD)

The penalty is defined by its derivative

$$p'_{\lambda,a}(\theta) = \lambda\left(\mathbb{1}(\theta \le \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda}\,\mathbb{1}(\theta > \lambda)\right)$$

for $\theta > 0$, $\lambda \ge 0$, and $a > 2$. This integrates to

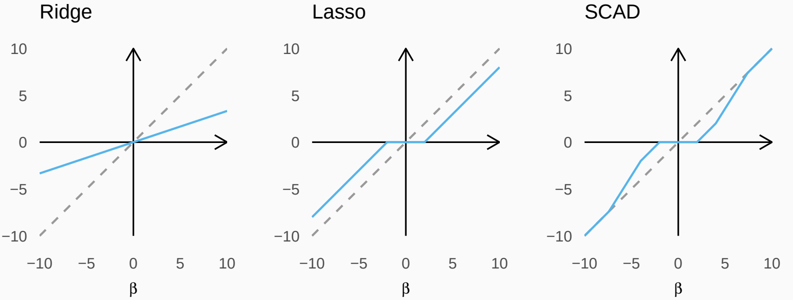
$$p_{\lambda,a}(\theta) = \begin{cases} \lambda\theta & 0 < \theta \le \lambda \\ -\frac{\theta^2 - 2a\lambda\theta + \lambda^2}{2(a-1)} & \lambda < \theta \le a\lambda \\ \frac{(a+1)\lambda^2}{2} & \theta > a\lambda \end{cases}$$



SCAD and bias (I)

The SCAD penalty is applied to each coefficient of $\boldsymbol{\beta}$ and (in case of orthonormal predictors) leads to the estimate

$$\widehat{\beta}_{\mathrm{SCAD},j} = \begin{cases} \mathrm{ST}(\widehat{\beta}_{\mathrm{OLS},j}, \lambda) & |\widehat{\beta}_{\mathrm{OLS},j}| \leq 2\lambda \\ \frac{(a-1)\widehat{\beta}_{\mathrm{OLS},j} - \mathrm{sign}(\widehat{\beta}_{\mathrm{OLS},j}) a\lambda}{a-2} & 2\lambda < |\widehat{\beta}_{\mathrm{OLS},j}| \leq a\lambda \\ \widehat{\beta}_{\mathrm{OLS},j} & |\widehat{\beta}_{\mathrm{OLS},j}| > a\lambda \end{cases}$$



SCAD and bias (II)

- **Good news:** The SCAD penalty gets rid of bias for larger coefficients, but also creates sparsity
- It can be shown theoretically that if $\lambda_n \to 0$ and $\sqrt{n}\lambda_n \to \infty$ when $n \to \infty$, then the SCAD penalty leads to an **oracle procedure**
- **Bad news:** The penalty is not convex and the standard optimization approaches cannot be used. The authors of the method (Fan and Li, 2001) proposed an algorithm based on local approximations.
- Is there a way to stay in the realm of convex functions?

Adaptive Lasso (I)

Consider the weighted lasso problem

$$\hat{\boldsymbol{\beta}}_{\text{ada}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} w_j |\beta_j|$$

where $w_j \geq 0$ for all j. This is called the **adaptive lasso**.

Surprisingly, it turns out that the right choice of weights can turn this procedure into an **oracle procedure**, i.e.

- large coefficients are unbiased
- the convergence rate is optimal
- the right variables are identified

Adaptive Lasso (II)

Let $\boldsymbol{\beta}^*$ be a $\sqrt{n}$-consistent estimate of $\boldsymbol{\beta}_{\text{true}}$, i.e. $\boldsymbol{\beta}^* - \boldsymbol{\beta}_{\text{true}}$ converges (in probability) to $\mathbf{0}$ at rate $n^{-1/2}$, e.g. $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ or $\hat{\boldsymbol{\beta}}_{\text{ridge}}$.

Computation:

- For $\gamma > 0$ set $w_j^* = 1/|\beta_j^*|^\gamma$ and $\mathbf{X}^* = \operatorname{diag}(\mathbf{w}^*)^{-1}\mathbf{X}$.
- Solve the unweighted lasso problem for $\mathbf{X}^*$ to get $\boldsymbol{\beta}^*_{\text{lasso}}$.
- Set $\hat{\boldsymbol{\beta}}_{\text{ada}} = \operatorname{diag}(\mathbf{w}^*)^{-1}\boldsymbol{\beta}^*_{\text{lasso}}$, which is the solution.

It can be shown (Zou, 2006), that if $\boldsymbol{\beta}^*$ is a $\sqrt{n}$-consistent estimate, $\lambda_n/\sqrt{n} \to 0$, and $\lambda_n n^{(\gamma-1)/2} \to \infty$, then the adaptive lasso is an **oracle procedure**.

Penalisation in GLMs

Penalisation can also be used in generalised linear models (GLMs), e.g. to perform **sparse logistic regression**.

Given $p(y|\boldsymbol{\beta}, \mathbf{x})$ the log-likelihood of the model is

$$\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) = \sum_{l=1}^{n} \log(p(y_l|\boldsymbol{\beta}, \mathbf{x}_l))$$

Instead of penalising the minimisation of the residual sum of squares (RSS), the **minimisation of the negative log-likelihood is penalized**, i.e.

$$\arg\min_{\boldsymbol{\beta}} -\mathcal{L}(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) + \lambda||\boldsymbol{\beta}||_1$$

Note: If $p(y|\boldsymbol{\beta}, \mathbf{x})$ is Gaussian and the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ is assumed, this is equivalent to RSS minimisation.

Sparse logistic regression

Recall: For logistic regression with $i_l \in \{0, 1\}$ it holds that

$$p(1|\boldsymbol{\beta}, \mathbf{x}) = \frac{\exp(\mathbf{x}^T \boldsymbol{\beta})}{1 + \exp(\mathbf{x}^T \boldsymbol{\beta})} \quad \text{and} \quad p(0|\boldsymbol{\beta}, \mathbf{x}) = \frac{1}{1 + \exp(\mathbf{x}^T \boldsymbol{\beta})}$$

and the penalised minimisation problem becomes

$$\arg\min_{\boldsymbol{\beta}} - \sum_{l=1}^{n} \left(i_l \mathbf{x}_l^T \boldsymbol{\beta} - \log\left(1 + \exp(\mathbf{x}^T \boldsymbol{\beta})\right)\right) + \lambda ||\boldsymbol{\beta}||_1$$

- The minimisation problem is still convex, but non-linear in $\boldsymbol{\beta}$. Iterative quadratic approximations combined with coordinate descent can be used to solve this problem.
- Another way to perform sparse classification (like e.g. nearest shrunken centroids)

Sparse multi-class logistic regression

In multi-class logistic regression with $i_l \in \{1, \dots, K\}$, there is a matrix of coefficients $\mathbf{B} \in \mathbb{R}^{p \times (K-1)}$ and it holds for $i = 1, \dots, K-1$ that

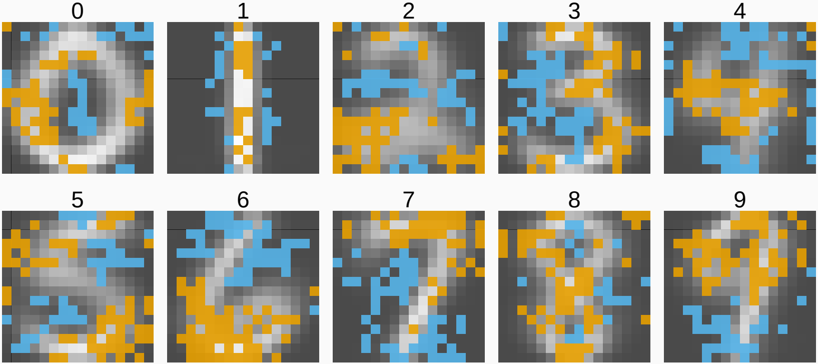
$$p(i|\mathbf{B}, \mathbf{x}) = \frac{\exp(\mathbf{x}^T \boldsymbol{\beta}_i)}{\sum_{j=1}^{K-1} \exp(\mathbf{x}^T \boldsymbol{\beta}_j)} \quad \text{and} \quad p(K|\mathbf{B}, \mathbf{x}) = \frac{1}{\sum_{j=1}^{K-1} \exp(\mathbf{x}^T \boldsymbol{\beta}_j)}$$

- As in two-class case, the absolute value of each entry in $\mathbf{B}$ can be penalised.
- Another possibility is to use the group lasso on all coefficients for one variable, i.e. penalise with $||\mathbf{B}_{j\cdot}||_2$ for $j = 1, \dots, p$.

Example for sparse multi-class logistic regression

MNIST-derived zip code digits ($n = 7291$, $p = 256$)

Sparse multi-class logistic regression was applied to the whole data set and the penalisation parameter was selected by 10-fold CV.



Orange tiles show positive coefficients and blue tiles show negative coefficients. The numbers below are the class averages.

The lasso and significance testing

- Calculating p-values or performing significance testing on sparse coefficient vectors is tricky
- **Probabilistic point-of-view:** Sparsity and exact zeros create a point mass at zero, but otherwise coefficients have an approximately Gaussian distribution (**spike-and-slab distribution**)
- In practice: Naive bootstrap is not going to work well since small changes in the data can change a coefficient from exact zero to non-zero
- Some other approaches (Wasserman and Roeder, 2009; Meinshausen et al., 2009) split the data once or multiple times into two subsets, perform variable selection on one part and use the other to perform least squares on the selected variables
- Still an active research topic, but the R package `hdi` contains some recent approaches

Data representations

Goals of data representation

Dimension reduction while retaining important aspects of the data

Goals can be

- Visualisation
- Interpretability/Variable selection
- Data compression
- Finding a representation of the data that is more suitable to the posed question

Let us start with **linear dimension reduction**.

Short re-cap: SVD

The **singular value decomposition (SVD)** of a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, $n \geq p$, is

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

where $\mathbf{U} \in \mathbb{R}^{n \times p}$ and $\mathbf{V} \in \mathbb{R}^{p \times p}$ with

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}_p \quad \text{and} \quad \mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}_p$$

and $\mathbf{D} \in \mathbb{R}^{p \times p}$ is diagonal. Usually

$$d_{11} \geq d_{22} \geq \cdots \geq d_{pp}$$

holds for the diagonal elements of $\mathbf{D}$.

SVD and best rank-q-approximation (I)

Write $\mathbf{u}_j$ and $\mathbf{v}_j$ for the columns of $\mathbf{U}$ and $\mathbf{V}$, respectively. Then

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \sum_{j=1}^{p} d_{jj} \underbrace{\mathbf{u}_j \mathbf{v}_j^T}_{\text{rank-1-matrix}}$$

Best rank-q-approximation: For $q < p$

$$\mathbf{X}_q = \sum_{j=1}^{q} d_{jj} \mathbf{u}_j \mathbf{v}_j^T$$

with **approximation error**

$$\left\|\mathbf{X} - \mathbf{X}_q\right\|_F^2 = \left\|\sum_{j=q+1}^{p} d_{jj} \mathbf{u}_j \mathbf{v}_j^T\right\|_F^2 = \sum_{j=q+1}^{p} d_j^2$$

SVD and best rank-q-approximation (II)

Notes

- $\mathbf{X}_q = \sum_{j=1}^{q} d_{jj}\mathbf{u}_j\mathbf{v}_j^T$ approximates $\mathbf{X}$ as a sum of layers
- This is the best possible rank-q-approximations
- How to choose q? Possibility: Look at singular values and decide a cut-off
- Interpretation is difficult since layers both add and subtract information
- $\mathbf{U}$ and $\mathbf{V}$ are not unique and could be made sparse

Alternative view of best rank-q-approximation

The matrix $\mathbf{X}_q$ is the unique solution to the following minimization problem (see notes on SVD on website)

$$\underset{\text{rank}(\mathbf{M})=q}{\arg\min} \|\mathbf{X} - \mathbf{M}\|_F^2$$

Alternative view:

Assume $\mathbf{X}$ stores samples as columns, i.e. $\mathbf{X} \in \mathbb{R}^{p \times n}$.

Set $\mathbf{H} := \mathbf{D}_q \mathbf{U}_q^T \in \mathbb{R}^{q \times n}$ and $\mathbf{W} = \mathbf{V}_q \in \mathbb{R}^{p \times q}$, where $\mathbf{D}_q$, $\mathbf{U}_q$, and $\mathbf{V}_q$ contain only the first q columns.

Then $\mathbf{X}_q = \mathbf{W}\mathbf{H}$ is a solution of

$$\underset{\mathbf{W} \in \mathbb{R}^{p \times q}, \mathbf{H} \in \mathbb{R}^{q \times n}}{\arg\min} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2$$

Note: Whereas $\mathbf{X}_q$ is the unique minimizer for the upper minimisation problem, the matrices $\mathbf{W}$ and $\mathbf{H}$ are not unique.

Low-rank matrix factorisation

Let $q < \min(p, n)$

$$\underset{\mathbf{W}\in\mathbb{R}^{p\times q},\mathbf{H}\in\mathbb{R}^{q\times n}}{\arg\min} ||\mathbf{X} - \mathbf{WH}||_F^2$$

Interpretation

- The columns of $\mathbf{W}$ can be seen as **basis vectors** or **coordinates** of a subspace in feature space
- The columns of $\mathbf{H}$ provide **coefficients** that combine the basis vectors in $\mathbf{W}$ to the closest q-dimensional approximation of the respective observation
- In the framework of **factor analysis** the columns of $\mathbf{W}$ are called **factors** and the columns of $\mathbf{H}$ are called **loadings**

Notes on factor analysis

- Originated mostly in psychometrics with the idea that factors could describe unobservable (latent) properties (e.g. intelligence)
- Typically assumes that $\mathbf{W}$ is orthogonal
- Even orthogonality of $\mathbf{W}$ does not ensure identifiability since for a orthogonal matrix $\mathbf{R} \in \mathbb{R}^{q \times q}$

  $$\mathbf{W}'\mathbf{H}' := (\mathbf{W}\mathbf{R})(\mathbf{R}^T\mathbf{H}) = \mathbf{W}\mathbf{H}$$

  and $\mathbf{W}'$ is orthogonal if $\mathbf{W}$ is
- Every orthogonal matrix describes a rotation and when applied to factors and loadings it is called a **factor rotation**
- Can be used to make either factors (**varimax rotation**) or loadings (**quartimax rotation**) sparse

Non-negative Matrix Factorization (NMF)

Idea: We can add constraints to the low-rank matrix factorisation problem.

Non-negative matrix factorisation (NMF): Let $q < \min(p, n)$

$$\underset{\mathbf{W}\in\mathbb{R}^{p\times q},\mathbf{H}\in\mathbb{R}^{q\times n}}{\arg\min} ||\mathbf{X} - \mathbf{WH}||_F^2 \quad \text{such that} \quad \mathbf{W} \geq 0,\ \mathbf{H} \geq 0$$

- $\mathbf{W}$ and $\mathbf{H}$ are again not uniquely identifiable.
- No analytic solution (the numerical problem is actually NP-hard in general)
- Choice of q not as straight-forward as for SVD
- Not directly applicable to data matrices with negative entries (can be solved through translation)
- So, why even bother?

Advantages of NMF

- **Interpretability:** As in the case of truncated SVD we are adding layers, but now all layers are positive and each layer adds information
- **Clustering interpretation:**
  - The columns of $\mathbf{W}$ can be interpreted as cluster centroids
  - Cluster membership of each observation is determined by the columns of $\mathbf{H}$
  - Observation j is assigned to the cluster k such that $H_{kj} > H_{ij}$ for all $i \neq k$

Take-home message

- Bias in lasso estimates can be bad for predictive strength. Using modified penalties can help
- Linear dimension reduction helps to factorise matrices into more interpretable components
- By adding non-negativity constraints to the matrix factorisation problem, NMF creates more interpretable results and can be used for clustering at the same time