Introduction to Machine Learning
HPC IA — 2019-20

4. Regularized linear regression

Chloé-Agathe Azencott
Centre for Computational Biology, Mines ParisTech
chloe-agathe.azencott@mines-paristech.fr

Inserm
Institut national
de la santé et de la recherche médicale

institutCurie

PSL
RESEARCH UNIVERSITY PARIS

MINES
ParisTech

Learning objectives

- Understand **regularization** as a means to control model complexity.
- Define **Lasso**, **ridge regression**, **elastic net.**
- Understand the role of the **l1 and l2 norms** in regularization
- Interpret **solution paths** for Lasso and ridge regression.

Regularization

OPTIONAL

Bias-variance tradeoff

- **Bias:** difference between the expected value of the estimator and the true value being estimated.

$$\text{Bias}(f(\boldsymbol{x})) = \mathbb{E}[f(\boldsymbol{x}) - y]$$

  - A simpler model has a higher bias.
  - **High bias can cause underfitting.**

- **Variance:** deviation from the expected value of the estimates.

$$\text{Var}(f(\boldsymbol{x})) = \mathbb{E}[(f(\boldsymbol{x}) - \mathbb{E}(f(\boldsymbol{x}))^2]$$

  - A more complex model has a higher variance.
  - **High variance can cause overfitting.**

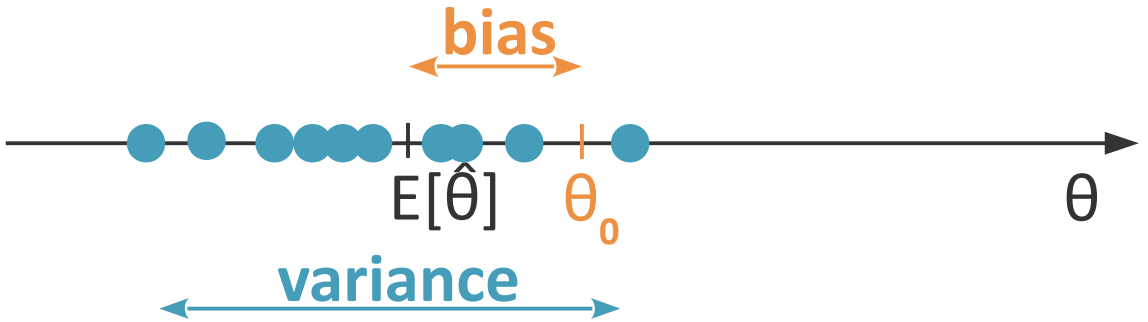
OPTIONAL

Bias-variance tradeoff

Mean squared error of the estimator:

$$\begin{aligned} \mathrm{MSE}(\hat{\theta}) &= \mathbb{E}[(\hat{\theta} - \theta_0)^2] \\ &= \mathrm{Var}(\hat{\theta}) + \mathrm{Bias}^2(\hat{\theta}) \end{aligned}$$

A biased estimator may achieve better MSE than an unbiased one.



Regularization

- **Empirical risk** is a poor estimate of the **expected risk** and of the **ability to generalize.**
- **Regularization: Tweaking** Empirical Risk Minimization to **improve generalization**

$$f^* = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y, f(\boldsymbol{x})) + \lambda \Omega(f).$$

- The **regularization term**
  - trades off errors on the training set for a model that generalizes better.
  - can be seen as a penalization of complex models.
- The **regularization coefficient** λ **must be tuned.**

Ridge regression

Ridge regression

- **Sum-of-squares penalty**

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

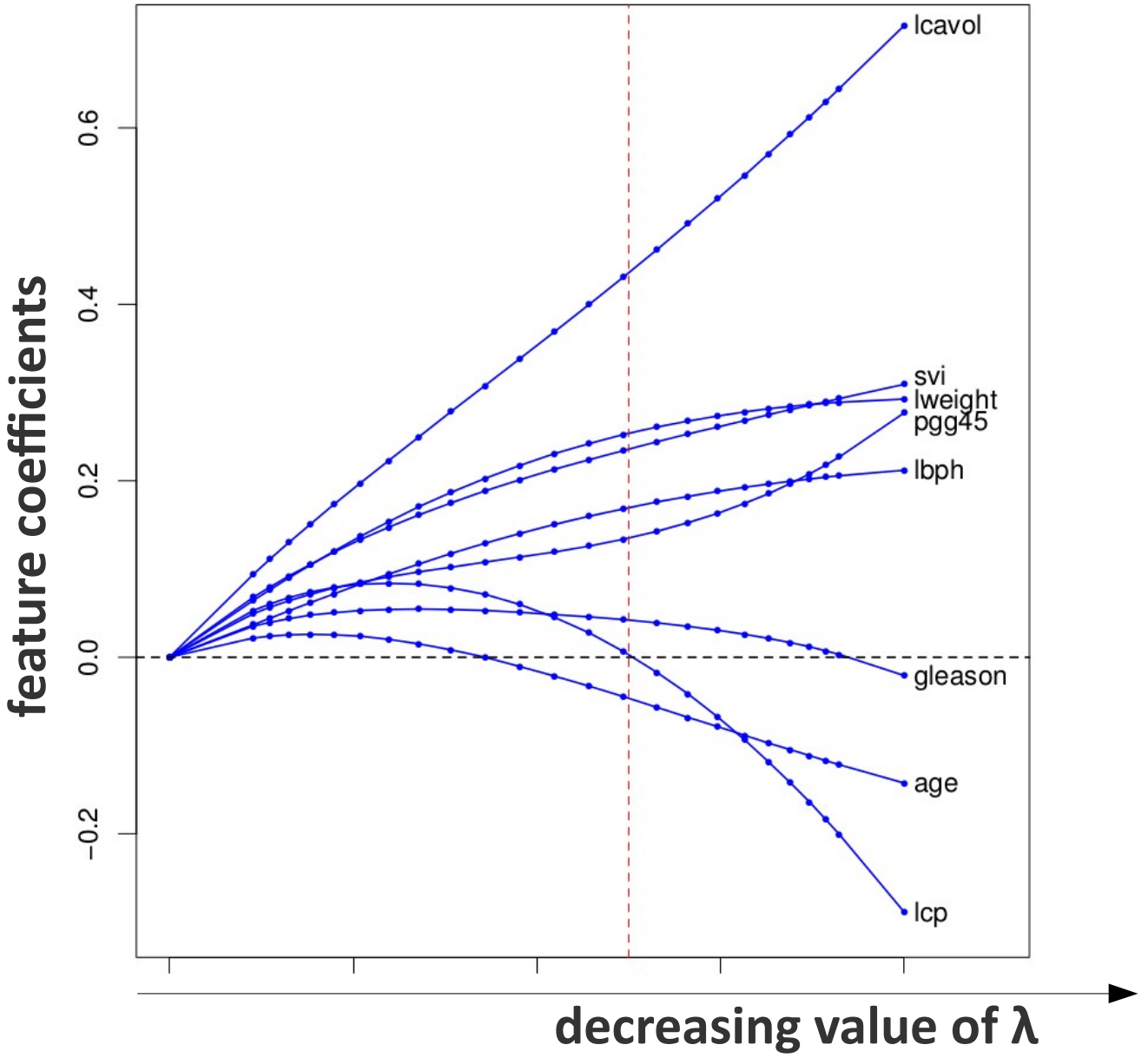
- **Ridge regression estimator:**

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (X^\top X + \lambda I)^{-1} X^\top \boldsymbol{y}$$

$$\text{if } (X^\top X + \lambda I) \text{ invertible.}$$

Always!

Ridge regression solution path



Standardization

- **Multiply x_j by a constant:**
  - For **standard linear regression:**

$$\hat{\beta}_j \rightarrow \frac{1}{c}\hat{\beta}_j$$

  - For **ridge regression:**

    Not so clear, because of the penalization term $\lambda\beta_j^2$

- Need to **standardize** the features

$$\tilde{x}_j^i = \frac{x_j^i}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_j^i - \bar{x}_j)^2}}$$

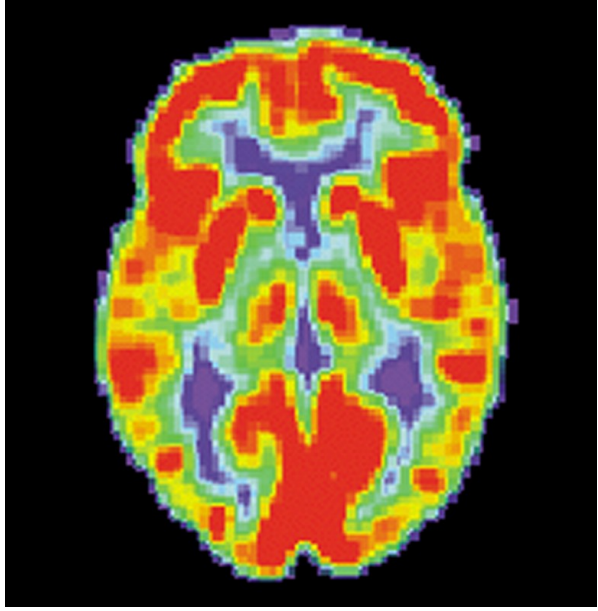
average value of x_j

Ridge regression

- **Grouped selection:**
  - correlated variables get similar weights
  - identical variables get identical weights

Lasso

Large p, small n



- **neuroimaging**

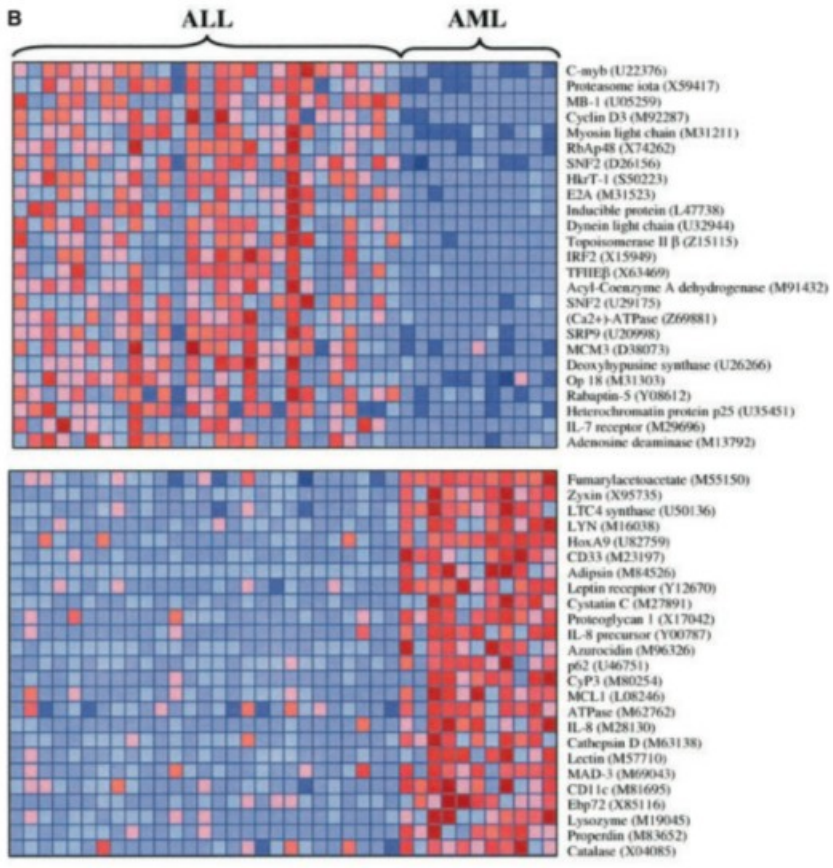
  thousands of brain regions / pixels / voxels

  much fewer patients

- **genetics and genomics**

  thousands of genes, millions of SNPs..

  usually, at best thousands of patients



When p is large

- **p > n:** $X^\top X$ not invertible
  - Use a pseudo-inverse M $(X^\top X)M(X^\top X) = (X^\top X)$
  - Multiple possible solutions
  - High variance of the estimator.
- Large p **reduces interpretability** of the model

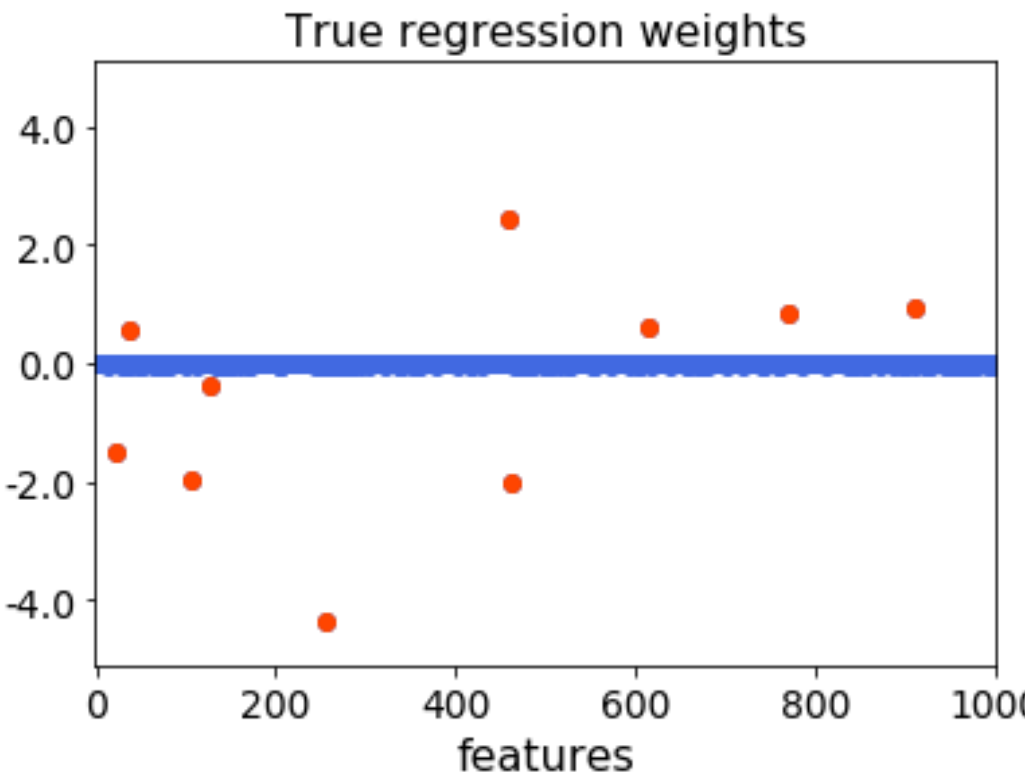
Would prefer a small subset of features with strong effects (= large coefficients).

OPTIONAL

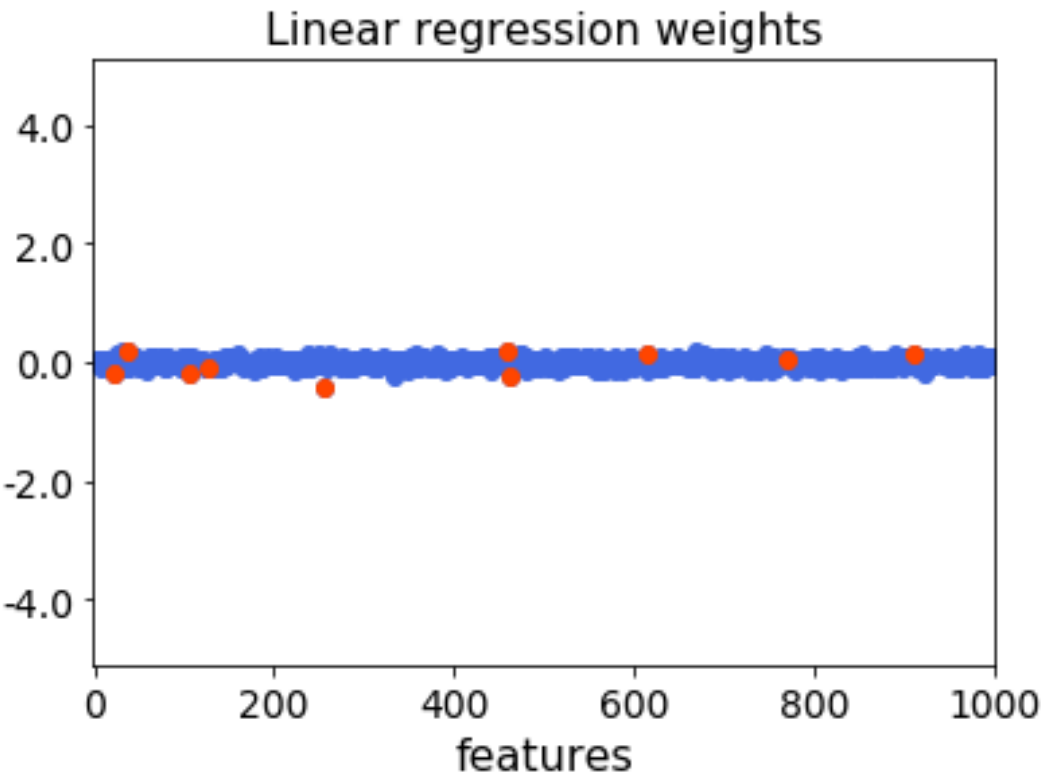
Linear regression when p >> n

Simulated data: p=1000, n=100, 10 causal features

True coefficients



Predicted coefficients

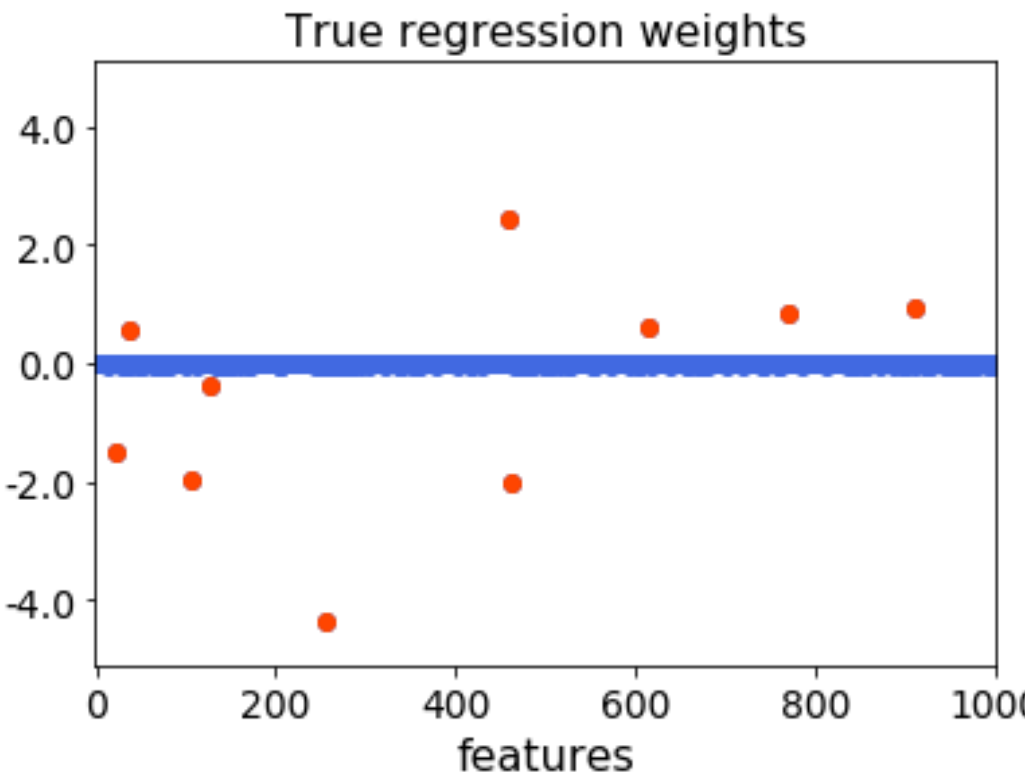


OPTIONAL

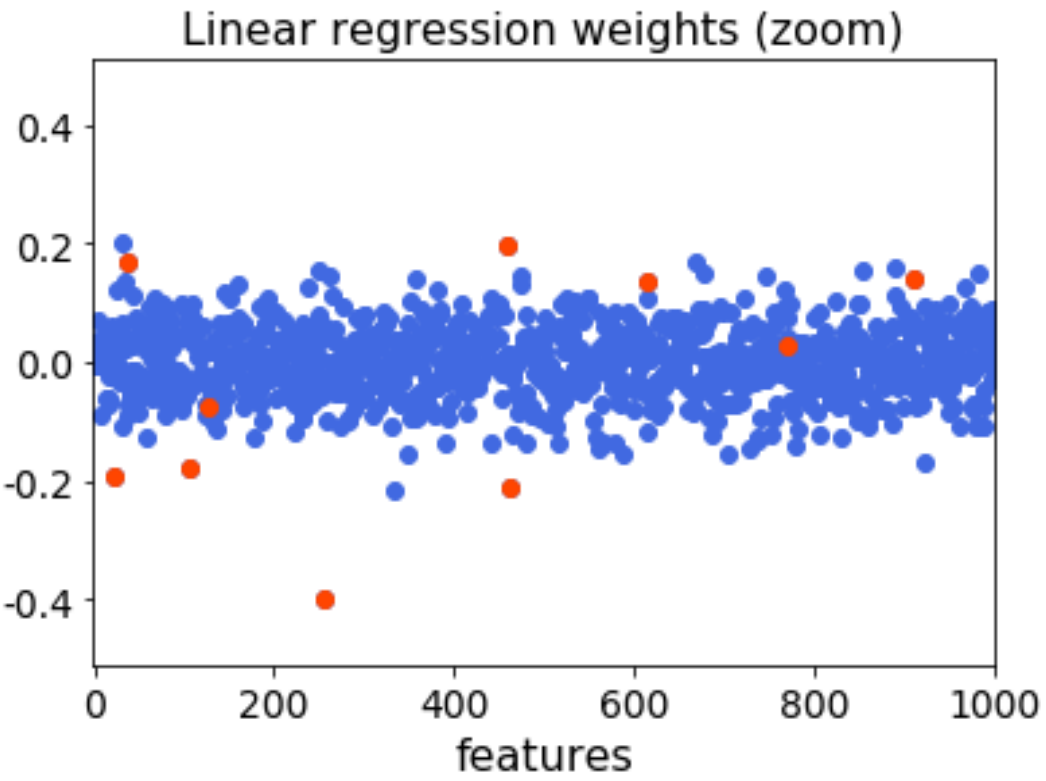
Linear regression when p >> n

Simulated data: p=1000, n=100, 10 causal features

True coefficients



Predicted coefficients



Lasso

$$||\boldsymbol{\beta}||_1 = \sum_{j=1}^{p} |\beta_p|$$

- **L1 penalty**

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

- aka **basis pursuit** (signal processing)
- no closed-form solution
- Equivalent to

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \arg\min_{\boldsymbol{\beta}} ||y - X\boldsymbol{\beta}||_2^2 \quad \text{s.t.} \quad ||\boldsymbol{\beta}||_1 \leq t$$

for a unique one-to-one match between t and λ.

OPTIONAL

Geometric interpretation

Minimize f(β) under the constraint g(β) ≤ 0

$$g(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1 - t$$

$$f(\boldsymbol{\beta}) = ||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2$$

OPTIONAL

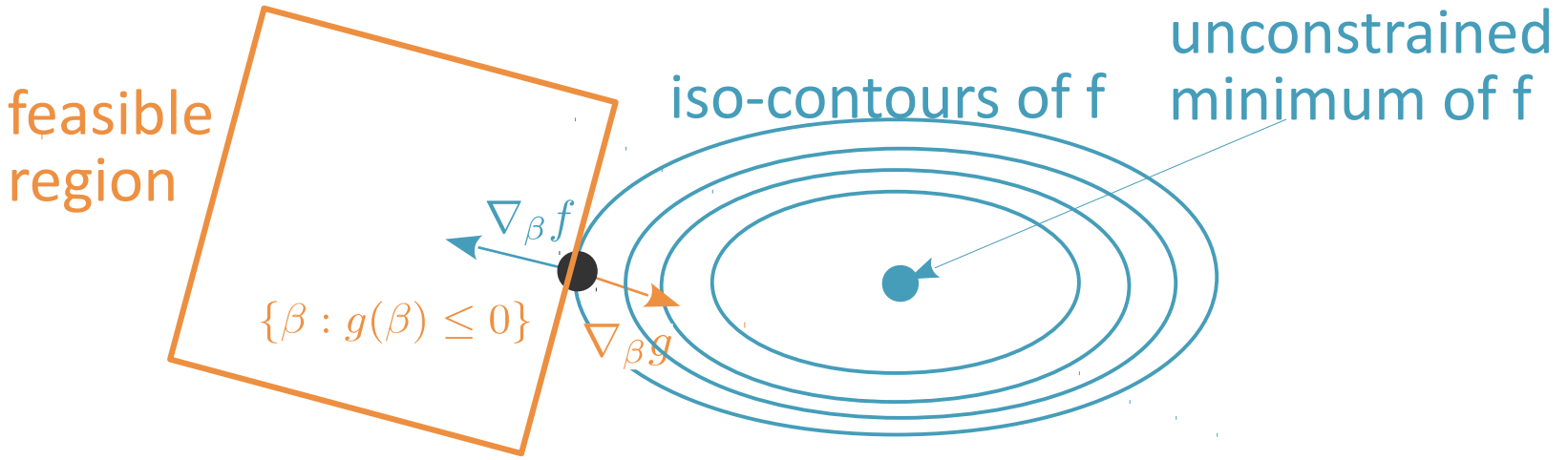
Geometric interpretation

Minimize f(β) under the constraint g(β) ≤ 0

- **Case 1:** the unconstrained minimum lies in the feasible region.
- **Case 2:** it does not.

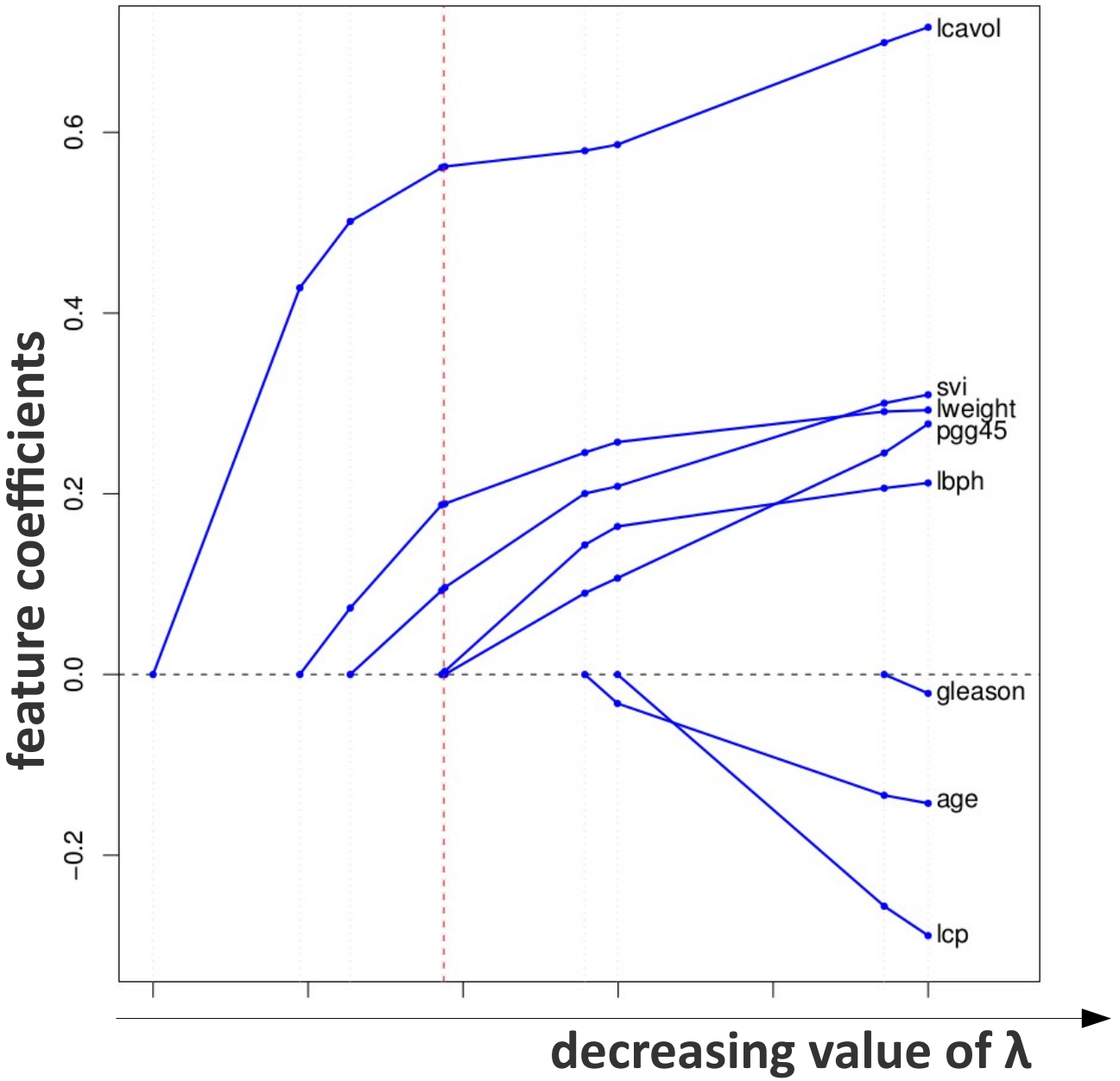
$$g(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_1 - t$$

$$f(\boldsymbol{\beta}) = ||\boldsymbol{y} - X\boldsymbol{\beta}||_2^2$$



The gradient is orthonormal to the iso-contours and points towards the direction of maximum increase.

Lasso solution path



Forward stepwise regression

- Build model **sequentially**, adding one variable at a time
  - Start with the intercept
  - At each step, add the variable that **most improves the fit**
  - **Stop when** $||\beta||_1 \leq t$
- Greedy solution

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.

$$\begin{aligned} \beta_j &\leftarrow \beta_j + \alpha \frac{1}{\sum_{i=1}^n (x_j^i)^2} \sum_{i=1}^n x_j^i r^i \\ &= \beta_j + \alpha (x_j^\top x_j)^{-1} x_j^\top r \\ &= \beta_j + \alpha \langle x_j^\top, x_j \rangle^{-1} \langle x_j, r \rangle \end{aligned}$$

step size

$$r = (y - \bar{y}) - \beta_j x_j$$

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.

4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.

$$r = (y - \bar{y}) - \beta_j x_j - \beta_k x_k$$

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.

4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.

4a. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.

4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.

4a. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.

5. Continue in this way until all p predictors have been entered.

OPTIONAL

Least Angle Regression

At each step, add "only as much of a variable as needed"

1. Standardize the predictors to have mean zero and unit norm. Start with the residual $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, $\beta_1, \beta_2, \ldots, \beta_p = 0$.

2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.

3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.

4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.

4a. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.

5. Continue in this way until all p predictors have been entered.

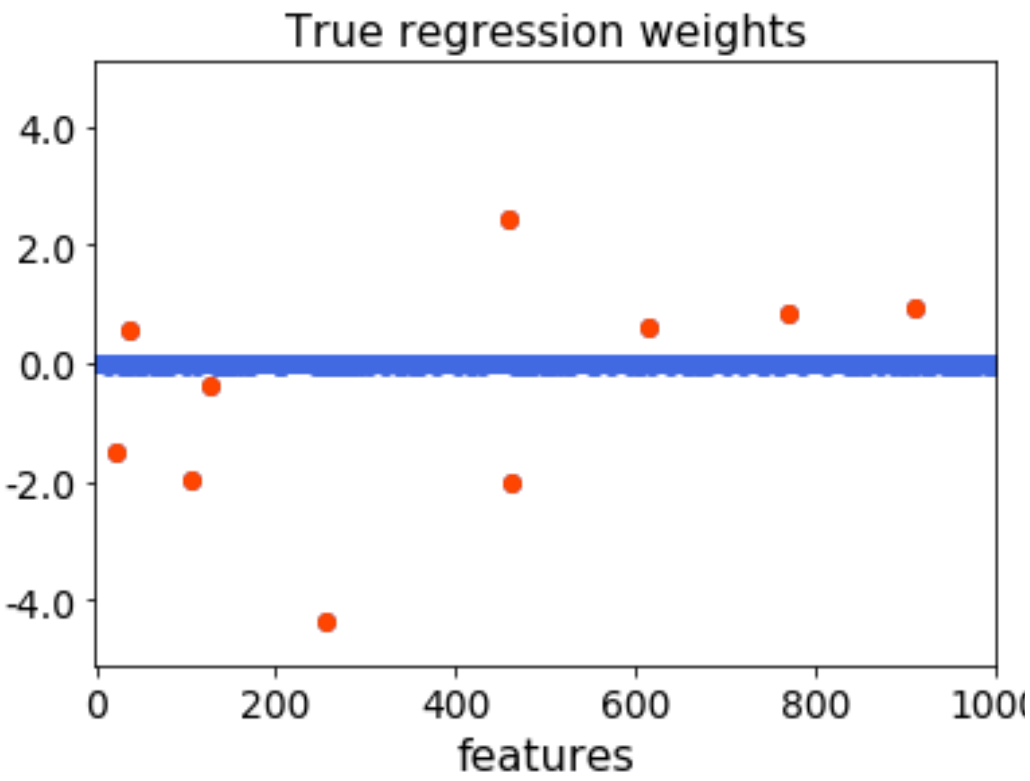
Maximum number of steps: max(n-1, p)

OPTIONAL

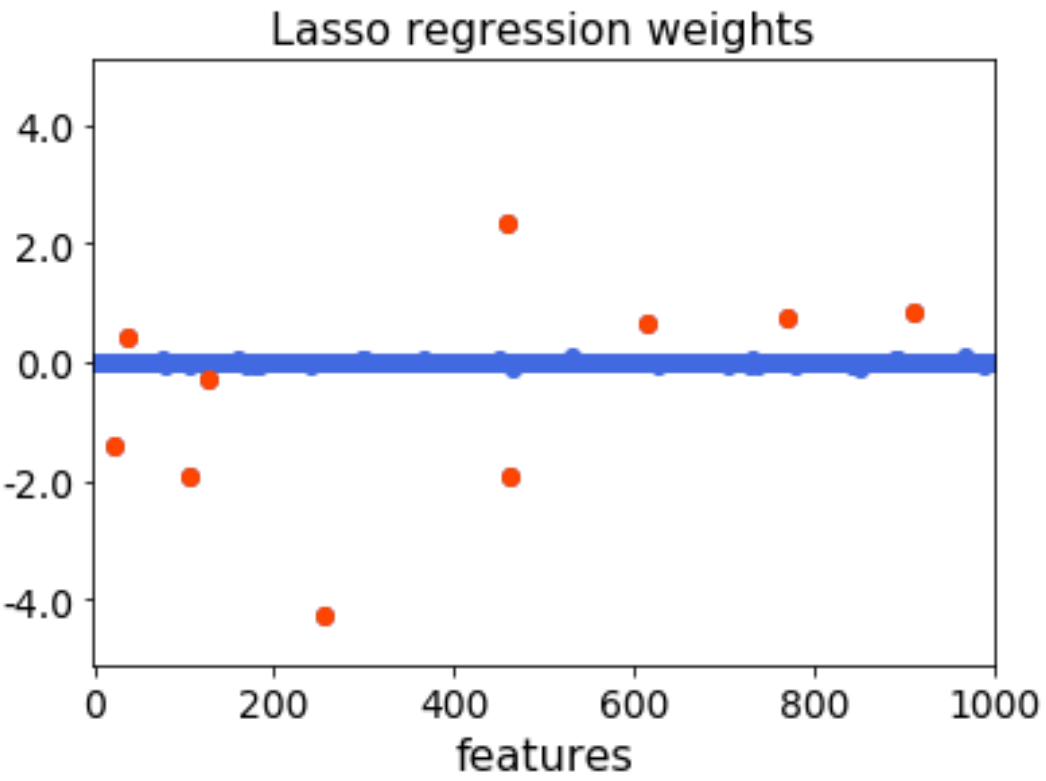
Linear regression when p >> n

Simulated data: p=1000, n=100, 10 causal features

True coefficients



Predicted coefficients



Elastic Net

Elastic Net

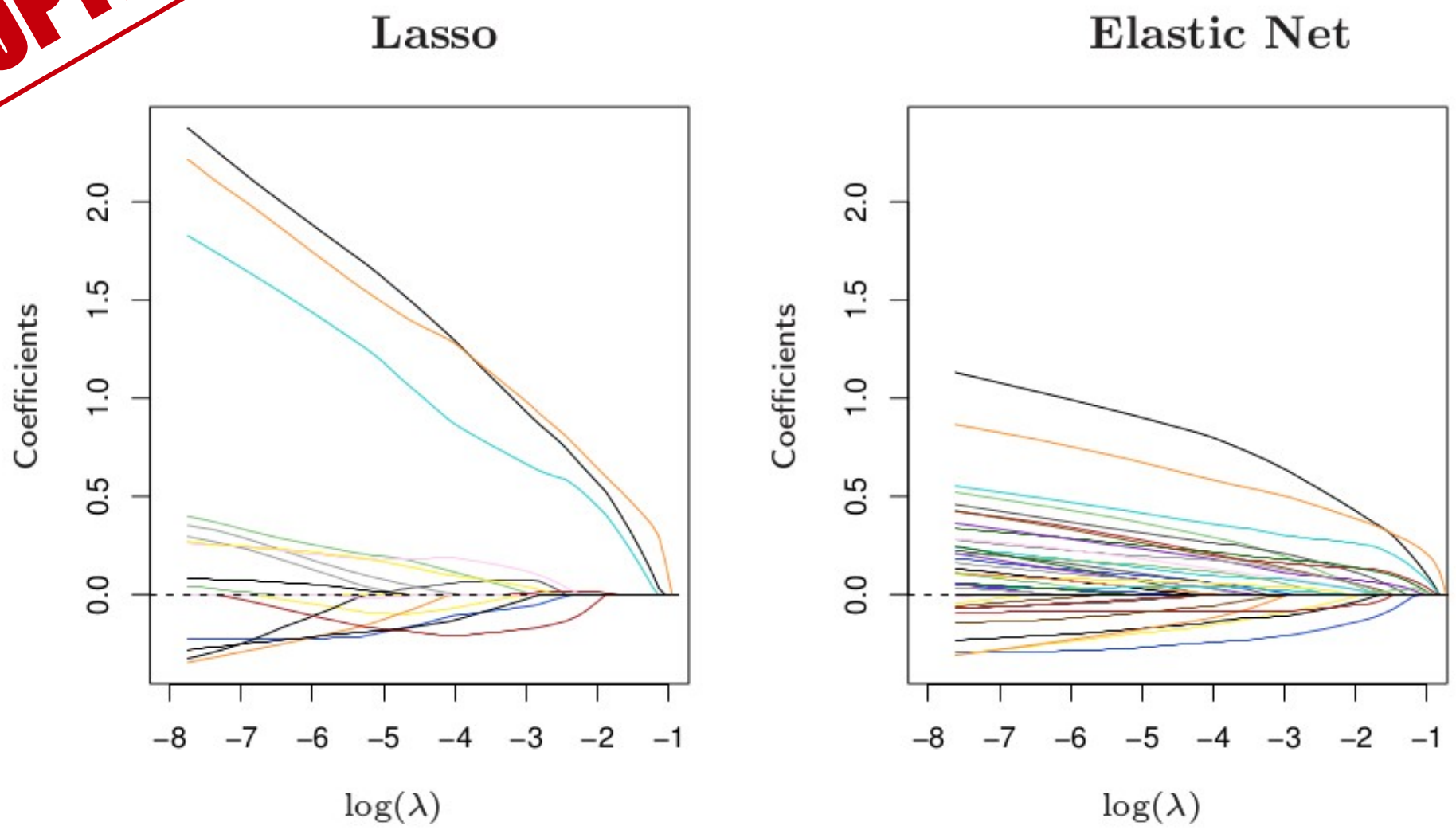
- **Combine lasso** and **ridge regression**

$$\hat{\boldsymbol{\beta}}_{\text{enet}} = \arg\min_{\boldsymbol{\beta}} ||y - X\boldsymbol{\beta}||_2^2 + \lambda \left(\alpha||\boldsymbol{\beta}||_2^2 + (1-\alpha)||\boldsymbol{\beta}||_1\right)$$

  - **Select variables** like the lasso.
  - **Shrinks together coefficients of correlated variables** like the ridge regression.

OPTIONAL

E.g. Leukemia data



Elastic Net results in more non-zero coefficients than Lasso, but with smaller amplitudes.

Lq-norm regularization

OPTIONAL

Lq-norm regularization

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} ||Y - X\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_q^q \qquad ||\boldsymbol{\beta}||_q = \left(\sum_{j=1}^{p} |\beta_j|^q \right)^{1/q}$$

Equivalently:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} ||Y - X\boldsymbol{\beta}||_2^2 \text{ s. t. } ||\boldsymbol{\beta}||_q^q \leq s$$

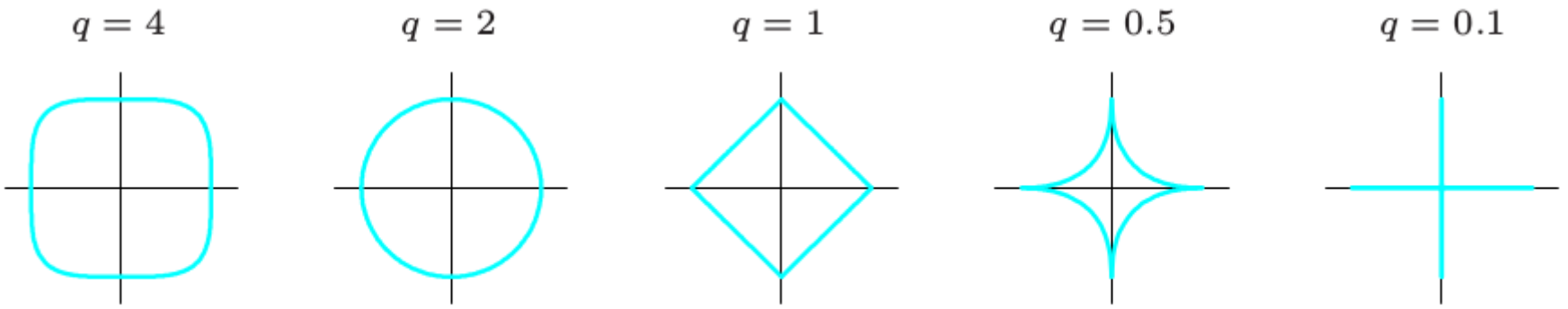


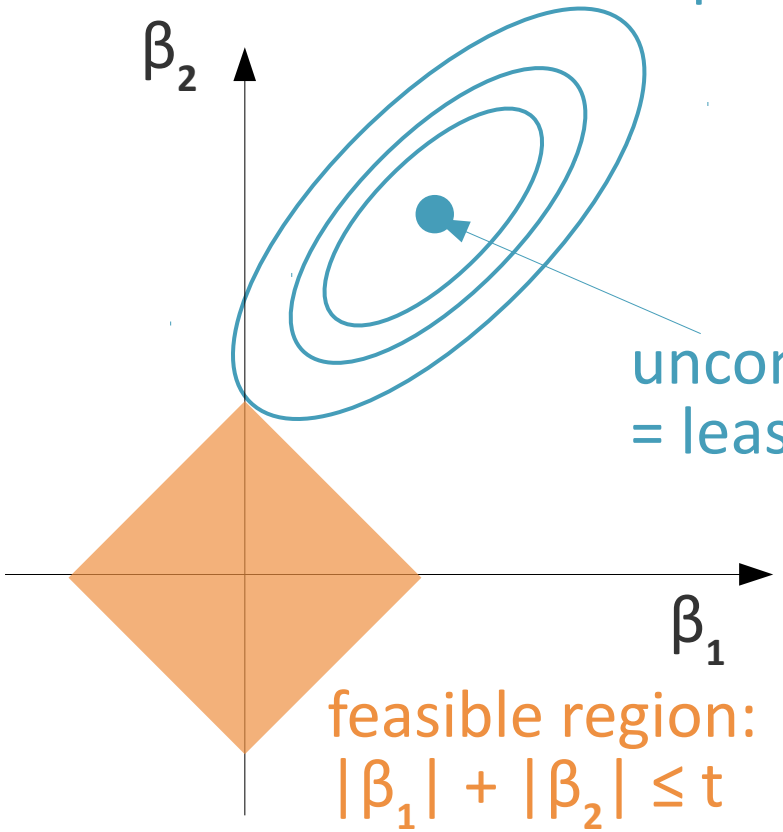
FIGURE 3.12. *Contours of constant value of* $\sum_j |\beta_j|^q$ *for given values of* q.

Lasso vs. ridge

OPTIONAL

L1 norm

iso-contours of the
least-squares error



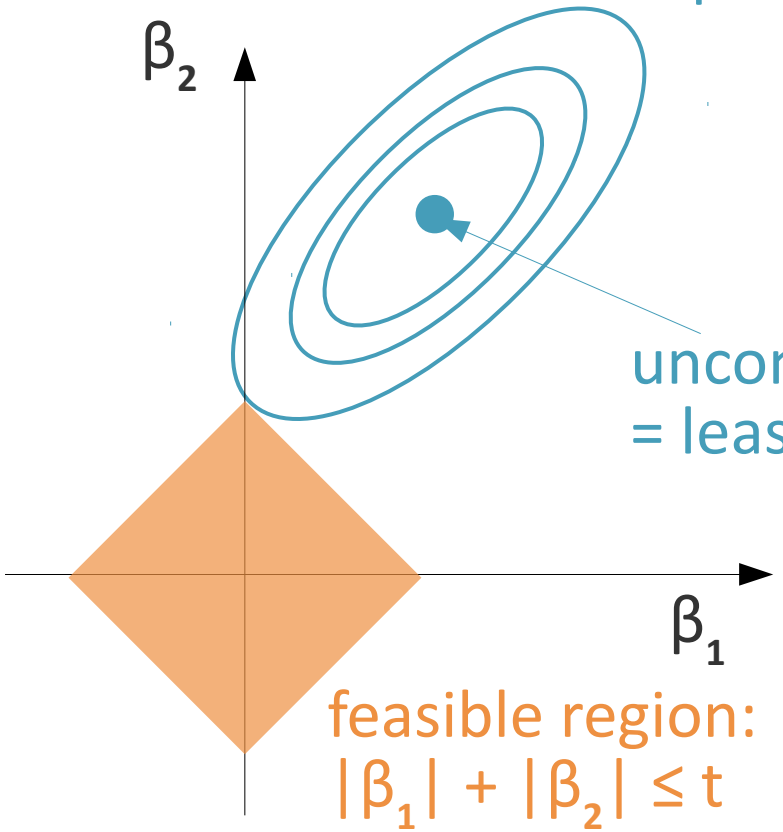
unconstrained min.
= least-squares sol.

feasible region:
$|\beta_1| + |\beta_2| \leq t$

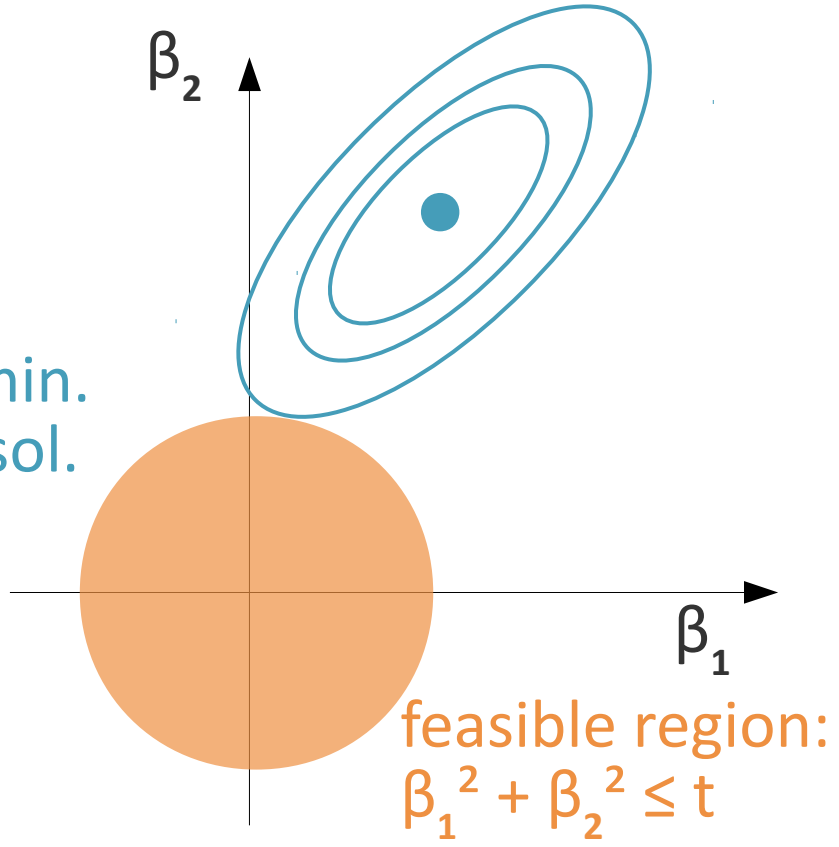
Lasso vs. ridge

OPTIONAL

L1 norm



L2 norm

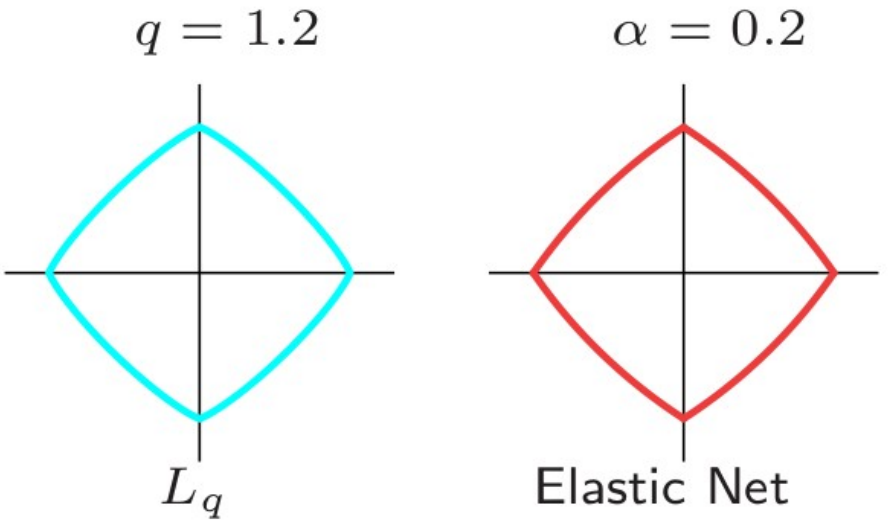


OPTIONAL

Elastic net

Elastic penalty

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||_2^2 + \lambda \left(\alpha||\beta||_2^2 + (1-\alpha)||\beta||_1\right)$$



Structured regularization

OPTIONAL

Group lasso

Use K predefined groups of variables that are known to "work" together and expected to be either all active or all inactive together.

E.g.: genes belonging to the same biological pathway.

$$\hat{\beta} = \arg\min_{\beta} ||y - \sum_{k=1}^{K} X_k \beta_k||_2^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k}\, ||\beta_k||_2$$

Features belonging to group k

Size of group k

Other examples of structured penalties

- **Overlapping groups**

  Jacob et al. (2009). Group lasso with overlap and graph lasso. *ICML.*

- **Graphs**

  Li & Li (2010). Variable selection and regression analysis for graph-structured covariates with an application to genomics. *Ann. App. Stats.*

- **Trees**

  Zhao et al. (2006). Grouped and hierarchical model selection through composite absolute penalties. *Ann. Stat.*

- **Multiple related tasks**

  Obozinski et al. (2006). Multitask feature selection. *Technical Report, UC Berkeley.*

Minimize SSE + λ x regularizer

- **Ridge**
  - reduces error by trading variance for bias
  - not sparse
  - analytical solution
- **Lasso**
  - randomly picks one of several correlated variables
  - sparse
  - LAR algorithm
- **Elastic net**
  - selects variables like the lasso
  - shrinks together the coefficients of correlated variables.
- **Many other regularizers** are possible

  Lp norms, groups, graphs, trees...

OPTIONAL

References

- *A Course in Machine Learning.*
  `http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf`
  - **Regularization:** Chap 7.2–7.3
- *The Elements of Statistical Learning.*
  `http://web.stanford.edu/~hastie/ElemStatLearn/`
  - **Regularization:** Chap 10.12
  - **Ridge regression:** Chap 3.4.1
  - **Lasso:** Chap 3.4.2
  - **LAR:** Chap 3.4.4
  - **Elastic Net:** Chap 4.2