Multivariate Analysis, I
2nd part

Aldo Solari

INFN School of Statistics

Paestum, June 6, 2019

Important concepts

- The model versus the modeling process
- Ensemble learning: bagging, random forests and boosting
- Regularized regression: ridge and lasso

Outline

1. **The modeling process**
2. Titanic data
3. Ensemble learning
4. Regularized regression

No free lunch

- The **No Free Lunch Theorem** (Wolpert 1996) is the idea that, without any specific knowledge of the problem or data at hand, *no one predictive model can be said to be the best*
- In practice, it is wise to try a number of disparate types of models to probe which ones will work well with your particular data set

The model versus the modeling process

- The modeling technique is a **small part** of the overall process
- The process of developing an effective model is both **iterative** and **heuristic**
- It is difficult to know the needs of any data set prior to working with it
- It is common for many approaches to be evaluated and modified before a model can be finalized

The modeling process

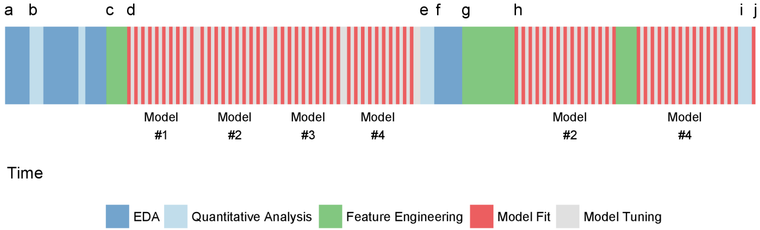


Image from Kuhn & Johnson (2019)

Common steps

Pre-processing and exploratory data analysis

- Handling missing data
- Exploring the relationships among the predictors and between predictors and the response
- Feature engineering
- Etc.

Model building

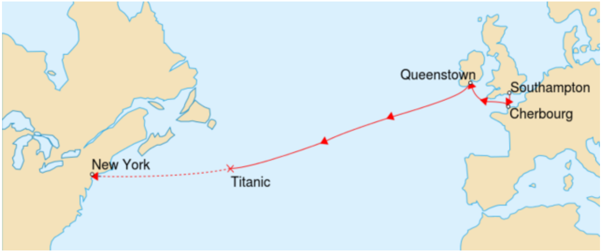
- Evaluating performance
- Parameter tuning
- Feature selection
- Etc.

Outline

1. The modeling process
2. **Titanic data**
3. Ensemble learning
4. Regularized regression

Titanic data

On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 809 out of 1309 passengers



- Training set of $n = 891$ passengers, each with $p = 10$ predictors
- The goal is to predict a 0 or 1 value for the survived variable for the $m = 418$ passengers in the test set

Classification

- Response $Y \in \{0, 1\}$
- Predictors $X = (X_1, \ldots, X_p)^\mathsf{T}$
- (X, Y) have some unknown joint distribution
- The regression function is

$$f(x) = \mathbb{E}(Y|X = x) = \Pr(Y = 1|X = x)$$

- The **Bayes' classification rule** is

$$C(x) = \begin{cases} 1 & \text{if } f(x) > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Bayes error rate

- A **classification rule** is any function $\hat{C} : x \mapsto \{0, 1\}$
- For example, the **plug-in rule**

$$\hat{C}(x) = \begin{cases} 1 & \text{if } \hat{f}(x) > 1/2 \\ 0 & \text{otherwise} \end{cases}$$

  where $\hat{f}$ is an estimate of f based on training data
- The Bayes classification rule is optimal because it has the smallest error rate:

$$\mathbb{E}\left[\Pr(Y \neq C(x))\right] \leq \mathbb{E}\left[\Pr(Y \neq \hat{C}(x))\right] \quad \forall\, \hat{C}$$

  where the expectation averages the probability over all possible values of X
- The Bayes error rate $\mathbb{E}\left[\Pr(Y \neq C(x))\right]$ is analogous to the irreducible error

Missclassification rate and accuracy

- Training set: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Test set: $(x_1^*, y_1^*), (x_2^*, y_2^*), \ldots, (x_m^*, y_m^*)$
- Missclassification rate

$$\mathrm{Err}_{\mathrm{Tr}} = \frac{1}{n} \sum_{i=1}^{n} I\{y_i \neq \hat{c}(x_i)\}$$

$$\mathrm{Err}_{\mathrm{Te}} = \frac{1}{m} \sum_{i=1}^{m} I\{y_i^* \neq \hat{c}(x_i^*)\}$$

- Accuracy

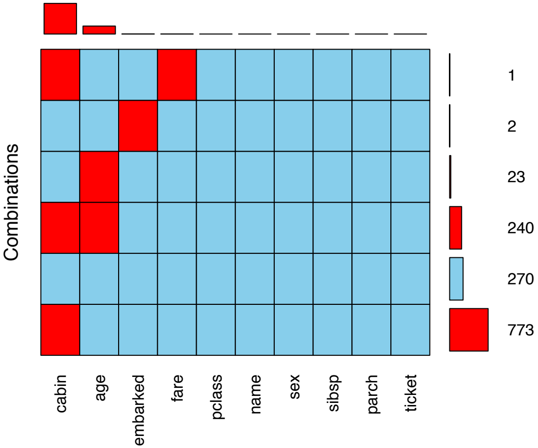
$$\mathrm{Acc}_{\mathrm{Te}} = 1 - \mathrm{Err}_{\mathrm{Te}}$$

Type of variables

pclass	Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
survived	Survival (0 = No; 1 = Yes)
name	Name
sex	Gender (male/female)
age	Age
sibsp	Number of Siblings/Spouses Aboard
parch	Number of Parents/Children Aboard
ticket	Ticket Number
fare	Passenger Fare
cabin	Cabin
embarked	Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Missing values

Predictor	Missing
cabin	1014
age	263
embarked	2
fare	1



Imputing missing values

survived	name	pclass	sex	age	ticket
0	Storey, Mr. Thomas	3	male	60.50	3701

sibsp	parch	fare	cabin	embarked
0	0	?	?	S

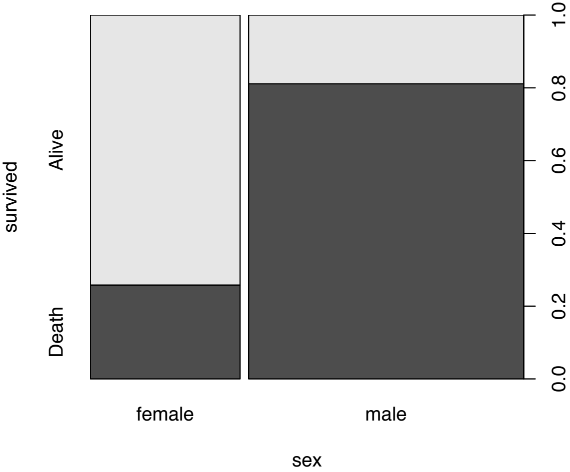
Imputing missing values

survived	name	pclass	sex	age	ticket
0	Storey, Mr. Thomas	3	male	60.50	3701

sibsp	parch	fare	cabin	embarked
0	0	?	?	S

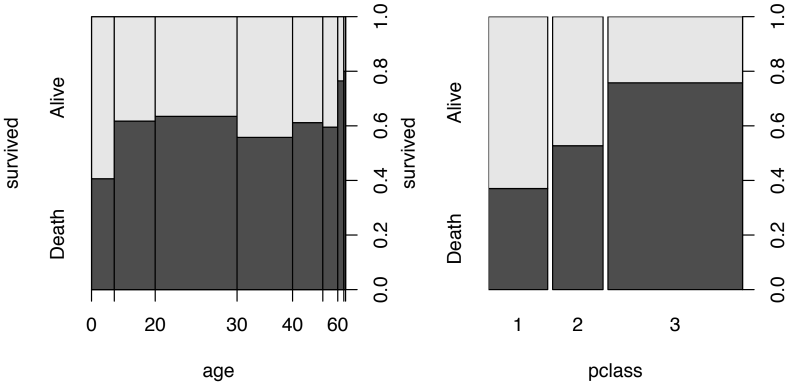
pclass	embarked	median fare
1	C	76.73
2	C	15.31
3	C	7.90
1	Q	90.00
2	Q	12.35
3	Q	7.75
1	S	52.00
2	S	15.38
3	**S**	**8.05**

Gender

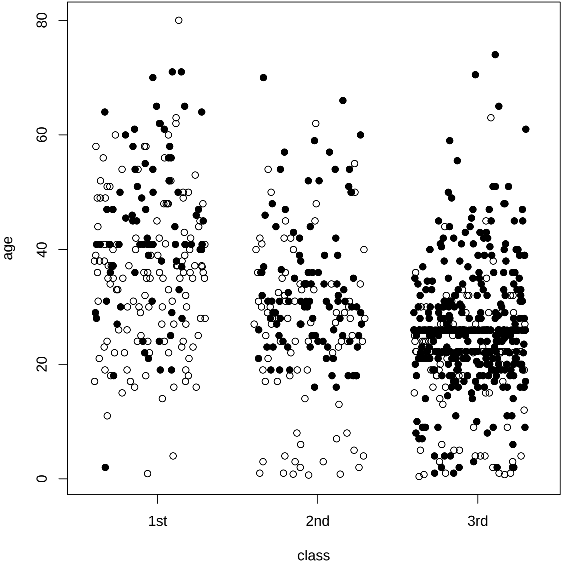


Women first. What about children?

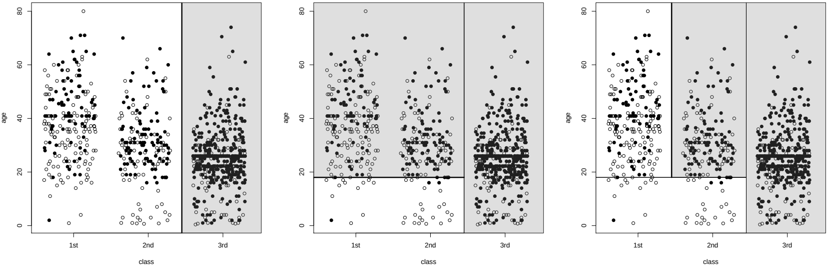
Age and pclass



Age and pclass combined



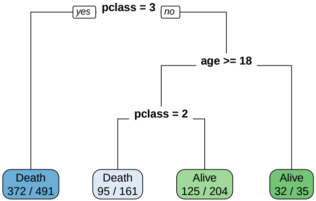
Classification tree



1st split (`pclass`)   2nd split (`age`)   3rd split (`pclass`)

Classification trees recursively partition the sample space into smaller and smaller rectangles

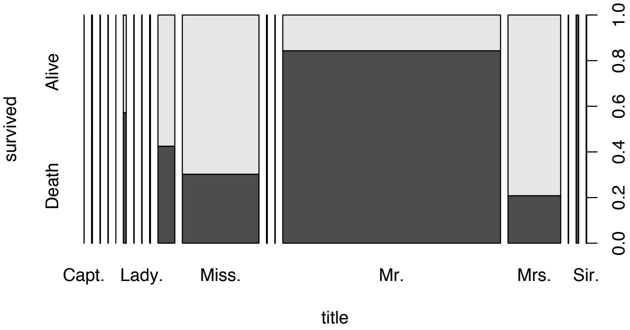
Classification rule



	Pr(Death)	Prediction
Class 3	76%	Death
Class 1-2, younger than 18	9%	Alive
Class 2, older than 18	56%	Death
Class 1, older than 18	39%	Alive

Feature engineering: title

Braund, **Mr.** Owen Harris
Cumings, **Mrs.** John Bradley
Heikkinen, **Miss.** Laina
Palsson, **Master.** Gosta Leonard
. . .



Performance

Predictors	$\mathrm{Acc}_{\mathrm{Tr}}$	$\mathrm{Acc}_{\mathrm{Te}}$
-	61.6%	62.2%
age	61.6%	62.2%
pclass	67.9%	67.2%
sex	78.7%	76.6%
age + pclass	70.9%	67.2%
age + sex	78.7%	76.6%
pclass + sex	78.7%	77.5%
age + pclass + sex	80.2%	76.6%
pclass + title	80.0%	**78.5**%

Titanic: summary

- **Missing values**: `fare` as a function of `pclass` and `embarked`
- **Exploratory data analysis**: `sex`, `age` and `pclass`
- **Feature engineering**: `title` from `name`
- **Performance**: `title` incorporates information about age (many missing values) and gender better

Outline

1 The modeling process

2 Titanic data

3 **Ensemble learning**

4 Regularized regression

Ensemble of trees

- Classification and regression trees are simple and useful for interpretation
- However they are not competitive with other approaches in terms of prediction accuracy
- **Ensemble methods** such as **bagging**, **random forest** and **boosting** grow multiple trees which are then combined to yield a single prediction
- Combining a large number of trees often result in improved prediction accuracy at the expense of interpretability

Instability of trees

- The primary disadvantage of trees is that they are rather unstable (high variance)
- In other words, a small change in the data often results in a completely different tree
- One major reason for this instability is that if a split changes, all the splits under it change as well, thereby propagating the variability
- Idea: **averaging** a set of variables (trees) reduces the variance: if $T_1, \ldots, T_B$ i.i.d. with $\mathbb{V}\mathrm{ar}(T_i) = \sigma^2$, then

$$\mathbb{V}\mathrm{ar}(\bar{T}) = \frac{\sigma^2}{n}$$

  where $\bar{T} = \frac{1}{B} \sum_{i=1}^{B} T_i$

- Problem: we need B copies of the training data

The bootstrap

- A bootstrap sample of size n from the training data is

$$(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2), \ldots, (\tilde{x}_n, \tilde{y}_n)$$

  where each $(\tilde{x}_i, \tilde{y}_i)$ are drawn from uniformly at random from

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, x_n)$$

  with **replacement**

- Not all of the training points are represented in a bootstrap sample, and some are represented more than once. For large n, the probability for one observation not to be drawn in any of the n draws is

$$\lim_{n \to \infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e} \approx 0.368$$

  We can expect $\approx 1/3$ of the n original observations to be **out-of-bag** (OOB)

Bootstrap aggregation (bagging)

1. Generate B different bootstrapped training sets

$$(\tilde{x}_1^b, \tilde{y}_1^b), (\tilde{x}_2^b, \tilde{y}_2^b), \ldots, (\tilde{x}_n^b, \tilde{y}_n^b), \qquad b = 1, \ldots, B$$

2. Fit a regression tree $\hat{f}^b$ or a classification tree $\hat{c}^b$ for each bootstrapped training set
3. Average all the predictions:

$$\bar{f}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x)$$

for regression trees and

$$\bar{c}(x) = \mathrm{Mode}\{\hat{c}^b(x), b = 1, \ldots, B\}$$

for classification trees (consensus)

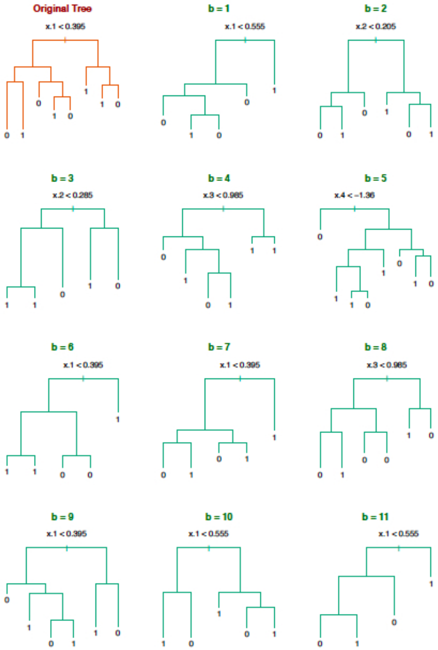


Image from Hastie, Tibshirani and Friedman (2009)

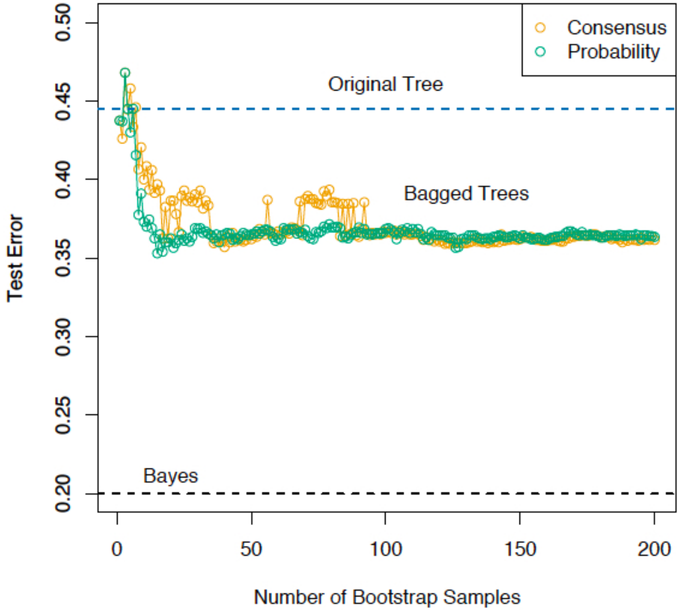


Image from Hastie, Tibshirani and Friedman (2009)

Random forest

- Random forest creates even more variation in individual trees
- Do as bagging, but before each split, select $m < p$ of the predictors **at random** as candidates for splitting
- Typically the tuning parameter m is $\sqrt{p}$ for classification and $p/3$ for regression

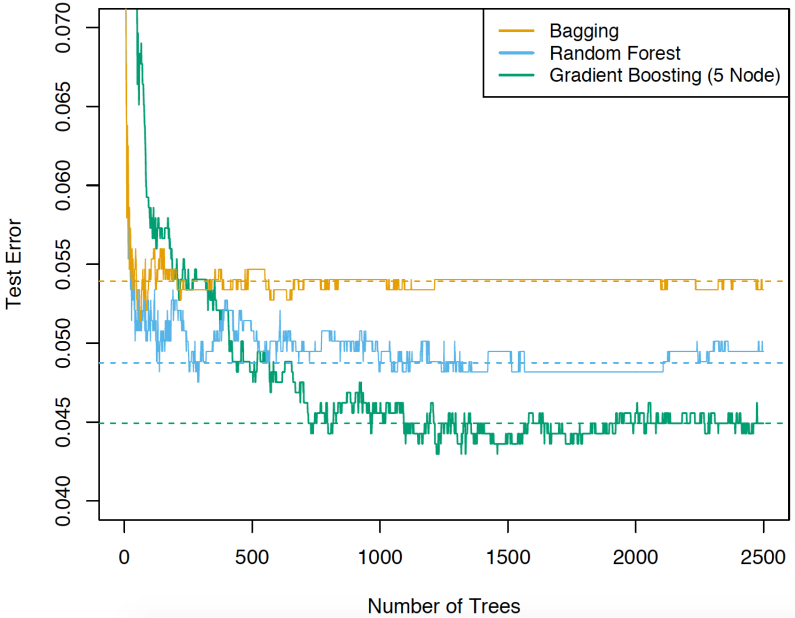


Image from Hastie, Tibshirani and Friedman (2009)

Why random forest works?

- Trees $T_1, \ldots, T_B$ constructed on B bootstrap copies of the training data are correlated
- Random sampling of the predictors **decorrelates** the trees. This reduces the variance when we average the trees
- Given a set of identical distributed (but not necessarily independent) variables $T_1, \ldots, T_B$ with pairwise correlation $\mathbb{C}\mathrm{orr}(T_j, T_l) = \rho$, mean $\mathbb{E}(T_j) = \mu$ and variance $\mathbb{V}\mathrm{ar}(T_j) = \sigma^2$, then

$$\mathbb{V}\mathrm{ar}(\bar{T}) = \rho\sigma^2 + \frac{(1-\rho)}{B}\sigma^2$$

- The idea in random forests is to improve the variance reduction of bagging by reducing the correlation ρ between the trees, without increasing the variance σ^2 too much

Boosting

- 1st algorithm: **adaboost** (Freund and Schapire, 1997) for classification problems
- It starts by fitting a classification tree with a single split (**stump**) to the training data
- Next, the classification tree is re-fitted, but with **more weight** given to misclassified observations
- This process is repeated until some stopping rule is reached

Toy example

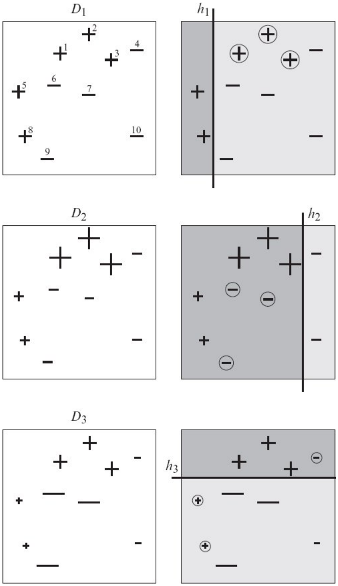


Image from Freund & Schapire

Classification rule

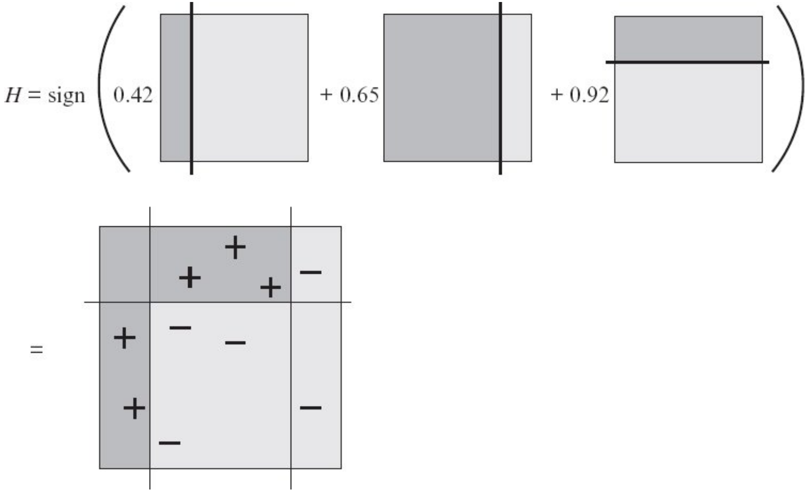


Image from Freund & Schapire

Ensemble learning: summary

- **Idea**: combining multiple trees at the expense of interpretability
- **Bagging**: use bootstrap to construct many trees
- **Random forest**: decorrelate the trees by randomly selecting predictors
- **Boosting**: iterative fitting with more weight to misclassified observations

Outline

1. The modeling process
2. Titanic data
3. Ensemble learning
4. **Regularized regression**

Linear regression

- Training and test data

$$\underset{n\times 1}{\mathbf{y}},\ \underset{n\times p}{\mathbf{X}} \qquad \underset{m\times 1}{\mathbf{y}^*},\ \underset{m\times p}{\mathbf{X}^*}$$

- Least squares problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

- Normal equations: $\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$
- Least squares estimator:

$$\underset{p\times 1}{\hat{\boldsymbol{\beta}}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

- Fitted values: $\underset{n\times 1}{\hat{\mathbf{y}}} = \mathbf{X}\hat{\boldsymbol{\beta}}$
- Prediction on test data: $\underset{m\times 1}{\hat{\mathbf{y}}^*} = \mathbf{X}^*\hat{\boldsymbol{\beta}}$

The failure of least squares in high dimensions

- When $\mathrm{rank}(\mathbf{X}) < p$, e.g. this happens when $p > n$, there are infinitely many solutions in the least square problem
- Suppose $p > n$ and $\mathrm{rank}(\mathbf{X}) = n$. Let $U = \mathrm{span}(\mathbf{X})$ be the n-dimensional space spanned by the columns of $\mathbf{X}$ and $V = U^{\perp}$ the $p - n$ dimensional space orthogonal complement of U, i.e. i.e. the non-trivial null space of $\mathbf{X}$
- Then $\mathbf{X}\mathbf{v} = \mathbf{0}_p$ for all $\mathbf{v} \in V$, and $\mathbf{X}^\top\mathbf{X}\mathbf{v} = \mathbf{X}^\top\mathbf{0}_p = \mathbf{0}_n$, the solution of the normal equations is

$$\underset{p\times 1}{\hat{\boldsymbol{\beta}}} = (\mathbf{X}^\top\mathbf{X})^{-}\mathbf{X}^\top\mathbf{y} + \mathbf{v} \quad \forall\, \mathbf{v} \in V$$

where $\mathbf{A}^{-}$ denotes the Moore-Penrose inverse of $\mathbf{A}$

Regularization

- Least squares:
$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$
- **Penalized** form
$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + P(\boldsymbol{\beta})$$
where $P(\cdot)$ is some (typically convex) penalty function
- **Constrained** form
$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 \quad \text{subject to } \boldsymbol{\beta} \in C$$
where C is some (typically convex) set

Penalized form

- Ridge regression

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|$$

- Lasso regression

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_{\ell_1}$$

with $\lambda \geq 0$ the tuning parameter (usually chosen by CV) and

$$\|\boldsymbol{\beta}\|_{\ell_1} = \sum_{j=1}^{p} |\beta_j| \qquad \|\boldsymbol{\beta}\| = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

are the ℓ_1 and ℓ_2 norms

Constrained form

- Ridge regression

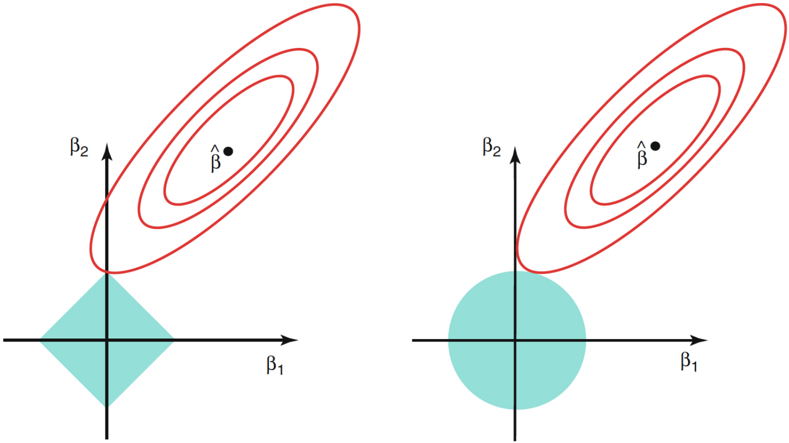
$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 \text{ subject to } \|\boldsymbol{\beta}\| \leq t$$

- Lasso regression

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 \text{ subject to } \|\boldsymbol{\beta}\|_{\ell_1} \leq t$$

with $t \geq 0$ the tuning parameter

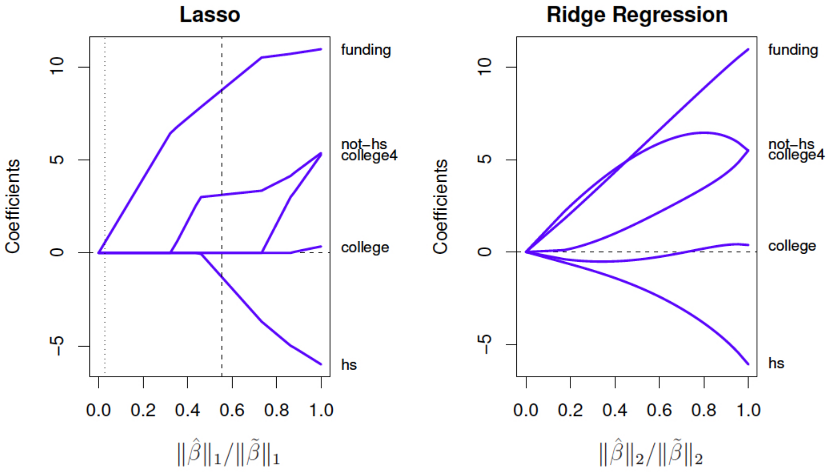
Penalized and constrained problems are equivalent: for any $t \geq 0$ and solution $\hat{\boldsymbol{\beta}}$ of the constrained problem, there is a $\lambda \geq 0$ such that $\hat{\boldsymbol{\beta}}$ also solves the penalized problem, and vice versa



Lasso

Ridge

Image from Hastie, Tibshirani and Friedman (2009)



Lasso $\hat{\boldsymbol{\beta}} = (8, 4, 0, 0, -1)^{\mathsf{T}}$ is **sparse**: many elements are 0

Image from Hastie, Tibshirani and Wainwright (2015)

Regularized regression: summary

- **High-dimensional data**: infinitely many solutions for $\hat{\beta}$
- **Modified least squares**: add penalty or constraint
- **L2/L1 norm**: ridge/lasso
- **Lasso**: sparse estimates