Data Sciences – CentraleSupelec
Advance Machine Learning
Course II - Linear regression/Linear classification

Emilie Chouzenoux

Center for Visual Computing
CentraleSupelec
emilie.chouzenoux@centralesupelec.fr

Linear regression

Motivations:

- Simple approach (essential to understand more sophisticated ones)
- Interpretable description of the relations inputs $\leftrightarrow$ outputs
- Can outperform nonlinear models, in the case of few training data/high noise/sparse data
- Extended applicability when combined with basis-function methods (see Lab)

Linear regression

Motivations:

- Simple approach (essential to understand more sophisticated ones)
- Interpretable description of the relations inputs $\leftrightarrow$ outputs
- Can outperform nonlinear models, in the case of few training data/high noise/sparse data
- Extended applicability when combined with basis-function methods (see Lab)

Applications: Prediction of

- Sale of products in the future based on past buying behaviour.
- Economic growth of a country or state.
- How much houses it would sell in the coming months and at what price.
- Number of goals a player would score in coming matches based on previous performances.
- Hours of study a student puts in, with respect to the exam results.

Linear model

Training data: $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$

$(\mathbf{x}_i)_{1 \leq i \leq n}$ are inputs / transformed version of inputs (eg, through log) / basis expansions.

Linear model

Training data: $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$
$(\mathbf{x}_i)_{1 \leq i \leq n}$ are inputs / transformed version of inputs (eg, through log) / basis expansions.
Fitting model:

$$y_i \approx f(\mathbf{x}_i) \quad (\forall i = 1, \ldots, n)$$

with, for every $i \in \{1, \ldots, n\}$,

$$f(\mathbf{x}_i) = \beta_0 1 + \beta_1 x_{i1} + \ldots + \beta_d x_{id} = \mathbf{x}_i'^{\top} \boldsymbol{\beta} = [\mathbf{X}\boldsymbol{\beta}]_i$$

with $\mathbf{X} \in \mathbb{R}^{n \times d+1}$ whose i-th line is $\mathbf{x}_i' = [1, x_{i1}, \ldots, x_{id}]$.

Linear model

Training data: $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$
$(\mathbf{x}_i)_{1\le i\le n}$ are inputs / transformed version of inputs (eg, through log) / basis expansions.
Fitting model:

$$y_i \approx f(\mathbf{x}_i) \quad (\forall i = 1, \ldots, n)$$

with, for every $i \in \{1, \ldots, n\}$,

$$f(\mathbf{x}_i) = \beta_0 1 + \beta_1 x_{i1} + \ldots + \beta_d x_{id} = \mathbf{x}_i'^\top \boldsymbol{\beta} = [\mathbf{X}\boldsymbol{\beta}]_i$$

with $\mathbf{X} \in \mathbb{R}^{n\times d+1}$ whose i-th line is $\mathbf{x}_i' = [1, x_{i1}, \ldots, x_{id}]$.
$\rightsquigarrow$ $[\beta_1, \ldots, \beta_d]$ defines a hyperplan in $\mathbb{R}^d$, and β_0 can be viewed as a bias shifting function f perpendicularly to the hyperplan.

Linear model

Training data: $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, \ldots, n$
$(\mathbf{x}_i)_{1 \leq i \leq n}$ are inputs / transformed version of inputs (eg, through log) / basis expansions.
Fitting model:

$$y_i \approx f(\mathbf{x}_i) \quad (\forall i = 1, \ldots, n)$$

with, for every $i \in \{1, \ldots, n\}$,

$$f(\mathbf{x}_i) = \beta_0 1 + \beta_1 x_{i1} + \ldots + \beta_d x_{id} = \mathbf{x}_i'^\top \boldsymbol{\beta} = [\mathbf{X}\boldsymbol{\beta}]_i$$

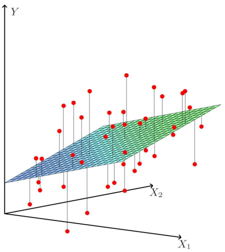
with $\mathbf{X} \in \mathbb{R}^{n \times d+1}$ whose i-th line is $\mathbf{x}_i' = [1, x_{i1}, \ldots, x_{id}]$.
$\rightsquigarrow$ $[\beta_1, \ldots, \beta_d]$ defines a hyperplan in $\mathbb{R}^d$, and β_0 can be viewed as a bias shifting function f perpendicularly to the hyperplan.
Goal: Using the training set, learn the linear function f (parametrized by $\boldsymbol{\beta}$) that predict a real value y from an observation $\mathbf{x}$.

Least Squares

Principle: Search for β that minimizes the sum of squares residuals

$$F(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 = \frac{1}{2}\|\mathbf{X}\beta - \mathbf{y}\|^2 = \frac{1}{2}\|\mathbf{e}\|^2$$

with $\mathbf{e} = \mathbf{X}\beta - \mathbf{y}$ the residual vector.



Optimization (reminders?)

We search for a solution to $\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta})$ where $F : \mathbb{R}^{d+1} \to \mathbb{R}$ is convex.
$\hat{\boldsymbol{\beta}}$ is minimizer if and only if $\nabla F(\hat{\boldsymbol{\beta}}) = 0$ where ∇F is the gradient of F, such that

$$[\nabla F(\boldsymbol{\beta})]_j = \frac{\partial F(\boldsymbol{\beta})}{\partial \beta_j} \quad (\forall j = 0, \dots, d).$$

Note that F also reads:

$$F(\boldsymbol{\beta}) = \frac{1}{2}\mathbf{y}^\top \mathbf{y} - \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{y} + \frac{1}{2}\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$$

Its gradient is $\nabla F(\boldsymbol{\beta}) = -\mathbf{X}^\top \mathbf{y} + \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$. Assuming that $\mathbf{X}$ has full column rank, then $\mathbf{X}^\top \mathbf{X}$ is positive definite, the solution is unique and reads:

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

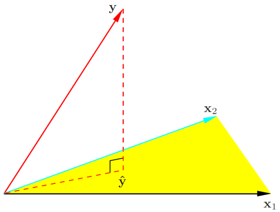
White board

Interpretation

The fitted values at the training inputs are

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} = \mathbf{H}\mathbf{y}$$

where $\mathbf{H}$ is called the “hat matrix”. This matrix computes the orthogonal projection of $\mathbf{y}$ onto the vectorial subspace spanned by the columns of $\mathbf{X}$.



Statistical properties

Variance:

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}^\top \mathbf{X})^{-1}\sigma^2$$

for uncorrelated observations y_i with variance σ^2, and deterministic $\mathbf{x}_i$.
Unbiased estimator:

$$\hat{\sigma}^2 = \frac{1}{n-(d+1)} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Inference properties: Assume that $Y = \beta_0 + \sum_{j=1}^{d} X_j \beta_j + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Then $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}$ are independant and

- $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, (\mathbf{X}^\top \mathbf{X})^{-1}\sigma^2)$
- $(n-(d+1))\hat{\sigma}^2 \sim \sigma^2 \chi^2_{n-(d+1)}$

High dimensional linear regression

Problems with least squares regression if d is large:

- *Accuracy*: The hyperplan fits the data well but predicts (generalizes) badly. (low bias / large variance)
- *Interpretation*: We want to identify a small subset of features important/relevant for predicting the data.

High dimensional linear regression

Problems with least squares regression if d is large:

- *Accuracy*: The hyperplan fits the data well but predicts (generalizes) badly. (low bias / large variance)
- *Interpretation*: We want to identify a small subset of features important/relevant for predicting the data.

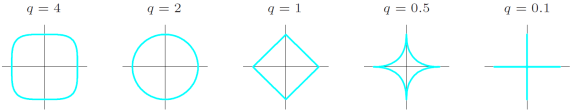
Regularization: $F(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda R(\beta)$

- ridge regression : $R(\beta) = \frac{1}{2}\|\beta\|^2$
- shrinkage : $R(\beta) = \|\beta\|_1$
- subset selection : $R(\beta) = \|\beta\|_0$

$\ast$ Explicit solution in the case of ridge. Otherwise, optimization method is usually needed !

White board

Penalty functions



Contour plots for $\sum_j |\beta_j|^q$

When the columns of $\mathbf{X}$ are orthonormal, the estimators can be deduced from the LS estimator $\hat{\beta}$ according to:

- Ridge : $\hat{\beta}_j/(1+\lambda)$ *weight decay*
- Lasso : $\text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \lambda)_+$ *soft tresholding*
- Best subset : $\hat{\beta}_j \cdot \delta\left(\hat{\beta}_j^2 \geq 2\lambda\right)$ *hard tresholding*

White board

White board

Robust regression

Challenge: Estimation methods insensitive to outliers and possibly high leverage points.

Approach: M-estimation

$$F(\beta) = \sum_{i=1}^{n} \rho(y_i - \mathbf{x}_i'^{\top}\beta)$$

with ρ a potential function satisfying:

- $\rho(e) \geq 0$ and $\rho(0) = 0$
- $\rho(e) = \rho(-e)$
- $\rho(e) \geq \rho(e')$ for $|e| \geq |e'|$

Robust regression

Challenge: Estimation methods insensitive to outliers and possibly high leverage points.

Approach: M-estimation

$$F(\beta) = \sum_{i=1}^{n} \rho(y_i - \mathbf{x}_i'^{\top}\beta)$$

with ρ a potential function satisfying:

- $\rho(e) \geq 0$ and $\rho(0) = 0$
- $\rho(e) = \rho(-e)$
- $\rho(e) \geq \rho(e')$ for $|e| \geq |e'|$

$*$ Minimizer satisfies:

$$\dot{\rho}(y_i - \mathbf{x}_i'^{\top}\hat{\beta})\mathbf{x}_i' = 0, \quad i = 1, \ldots, n$$

$\Rightarrow$ *IRLS algorithm*.

IRLS algorithm

Core idea: Let f be defined as

$$(\forall x \in \mathbb{R}) \qquad \rho(x) = \phi(|x|)$$

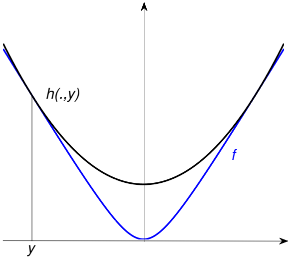
where

(i) ϕ is differentiable on $]0, +\infty[$,

(ii) $\phi(\sqrt{\cdot})$ is concave on $]0, +\infty[$,

(iii) $(\forall x \in [0, +\infty[) \quad \dot{\phi}(x) \geq 0$,

(iv) $\lim_{\substack{x \to 0 \\ x > 0}} \left(\omega(x) := \frac{\dot{\phi}(x)}{x}\right) \in \mathbb{R}$.



Then, for all $y \in \mathbb{R}$,

$$(\forall x \in \mathbb{R}) \quad \rho(x) \leq \rho(y) + \dot{\rho}(y)(x-y) + \frac{1}{2}\omega(|y|)(x-y)^2.$$

Examples of functions ρ

	$\rho(x)$	$\omega(x)$ (exercise)
Convex	$\lvert x\rvert - \delta \log(\lvert x\rvert/\delta + 1)$	
	$\begin{cases} x^2 & \text{if } \lvert x\rvert < \delta \\ 2\delta\lvert x\rvert - \delta^2 & \text{otherwise} \end{cases}$	
	$\log(\cosh(x))$	
	$(1 + x^2/\delta^2)^{\kappa/2} - 1$	
Nonconvex	$1 - \exp(-x^2/(2\delta^2))$	
	$x^2/(2\delta^2 + x^2)$	
	$\begin{cases} 1 - (1 - x^2/(6\delta^2))^3 & \text{if } \lvert x\rvert \leq \sqrt{6}\delta \\ 1 & \text{otherwise} \end{cases}$	
	$\tanh(x^2/(2\delta^2))$	
	$\log(1 + x^2/\delta^2)$	

$(\lambda, \delta) \in]0, +\infty[^2, \ \kappa \in [1, 2]$

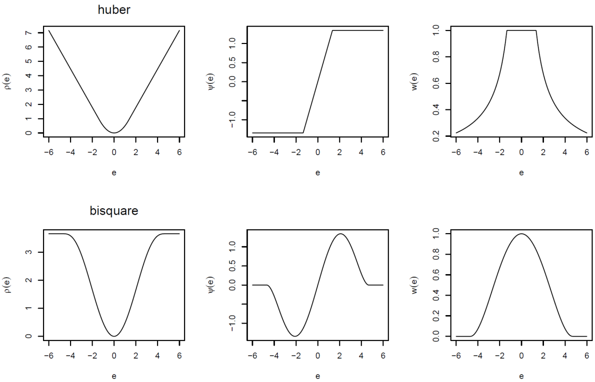
:

White board

IRLS algorithm:

$$(\forall k \in \mathbb{N}) \quad \beta_{k+1} = (\mathbf{X}^\top \mathbf{W}_k \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_k \mathbf{y}.$$

with the IRLS weight matrix $\mathbf{W}_k = \mathrm{Diag}(\omega(\mathbf{y} - \mathbf{X}\beta_k))$.



Linear classification

Applications:

- Sentiment analysis from text features
- Handwritten digits recognition
- Gene expression data classification
- Object recognition in images

Linear classification

Applications:

- Sentiment analysis from text features
- Handwritten digits recognition
- Gene expression data classification
- Object recognition in images

Goal: Learn linear functions $f_k(\cdot)$ for dividing the input space into a collection of K regions.

- Map a linear function on $\Pr(G = k|X = x) \sim$ linear regression
- More generally, map a linear function to a transformation of $\Pr(G = k|X = x)$

Logistic regression

Model:

$$\log \frac{\Pr(G=1|X=x)}{\Pr(G=K|X=x)} = \beta_{10} + \boldsymbol{\beta}_1^\top x$$

$$\log \frac{\Pr(G=2|X=x)}{\Pr(G=K|X=x)} = \beta_{20} + \boldsymbol{\beta}_2^\top x$$

$$\vdots$$

$$\log \frac{\Pr(G=K-1|X=x)}{\Pr(G=K|X=x)} = \beta_{(K-1)0} + \boldsymbol{\beta}_{K-1}^\top x$$

Logistic regression

$\Rightarrow$ For every $k = 1, \ldots, K-1$,

$$\Pr(G = k | X = x) = \frac{\exp(\beta_{k0} + \boldsymbol{\beta}_k^\top x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \boldsymbol{\beta}_\ell^\top x)}$$

and

$$\Pr(G = K | X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \boldsymbol{\beta}_\ell^\top x)}$$

Logistic regression

$\Rightarrow$ For every $k = 1, \dots, K-1$,

$$\Pr(G = k | X = x) = \frac{\exp(\beta_{k0} + \boldsymbol{\beta}_k^\top x)}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \boldsymbol{\beta}_\ell^\top x)}$$

and

$$\Pr(G = K | X = x) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\beta_{\ell 0} + \boldsymbol{\beta}_\ell^\top x)}$$

Loss function:

$$F(\Theta) = \sum_{i=1}^{n} -\log \Pr(G = g_i | X = \mathbf{x}_i; \Theta)$$

where Θ gathers the whole parameters set, and g_i the class label associated to entry $\mathbf{x}_i$.

White board

Binary case

- Sign response: $(\forall i = 1, \ldots, n) \quad y_i = -1$ if $g_i = 1$, and $y_i = +1$ if $g_i = 2$.

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \boldsymbol{\beta}^\top \mathbf{x}_i))$$

Binary case

- Sign response: $(\forall i = 1, \ldots, n) \quad y_i = -1$ if $g_i = 1$, and $y_i = +1$ if $g_i = 2$.

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \boldsymbol{\beta}^\top \mathbf{x}_i))$$

  - Function F is convex, differentiable.
  - Useful inequality for $f(x) = \log(1 + e^x)$:

$$(\forall (x, y) \in \mathbb{R}^2) \quad f(x) \leq f(y) + \dot{f}(y)(x - y) + \frac{1}{2}\omega(y)(x - y)^2$$

with $\dot{f}(y) = \frac{e^y}{1+e^y}$ and $\omega(y) = \frac{1}{y}(\frac{1}{1+e^{-y}} - \frac{1}{2}) \Rightarrow$ *IRLS algorithm*.

Binary case

- Sign response: $(\forall i = 1, \ldots, n) \quad y_i = -1$ if $g_i = 1$, and $y_i = +1$ if $g_i = 2$.

$$F(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log(1 + \exp(-y_i \boldsymbol{\beta}^\top \mathbf{x}_i))$$

  - Function F is convex, differentiable.
  - Useful inequality for $f(x) = \log(1 + e^x)$:

$$(\forall (x, y) \in \mathbb{R}^2) \quad f(x) \leq f(y) + \dot{f}(y)(x - y) + \frac{1}{2}\omega(y)(x - y)^2$$

with $\dot{f}(y) = \frac{e^y}{1+e^y}$ and $\omega(y) = \frac{1}{y}(\frac{1}{1+e^{-y}} - \frac{1}{2}) \Rightarrow$ *IRLS algorithm*.

  - For large datasets (i.e. large n) $\rightsquigarrow$ Need for regularization to avoid over-fitting + online minimization technique (see next course!).

White board

White board