

Machine/Statistical Learning

Linear models: regression and logistic regression

SICOM, M2 Sigma

2022-23

Model based approaches

Reminder on Supervised Learning

- ▶ input data $X \in \mathbb{R}^p$
- ▶ response Y to be predicted
- ▶ training set $(X_1, Y_1), \dots, (X_n, Y_n)$

In a model based approach, we seek an explicit relation between the (input) data X and the response Y . We focus here on *Discriminative models*, where we just model explicitly the conditional distribution $P(Y|X)$ rather than the joint distribution $P(X, Y)$.

Model based approaches: Generative vs Discriminative methods

Generative methods

Deduction of $P(Y|X)$ from Bayes rule

- ▶ Linear or Quadratic Discriminant Analysis
- ▶ Naïve Bayes
- ▶ ...

Discriminative methods

Direct learning of $P(Y|X)$, e.g.

- ▶ Linear regression
- ▶ Logistic "regression" (← generalized linear model for [classification](#) tasks)
- ▶ ...

Linear model: Keep it simple!

Simple linear approach may seem overly simplistic

- true prediction functions are never linear
- + extremely useful, both conceptually and practically

Practically

Gorge Box, 60': "Essentially, all models are wrong, but some are very useful"

- 👉 *Simple is actually very good*: works very well in a lot of situations by capturing the main effects (which are generally the most interesting)

Conceptually

Many concepts developed for the linear problem are important for a lot of the supervised learning techniques

- 👉 Although it is never correct, a linear model serves as a good and interpretable approximation of the unknown true function $f(X)$

Outline

Reminder on Linear regression

Stochastic Gradient Descent

Regularization and shrinkage methods

- Ridge regression

- Lasso estimator

- Application: prostate data

Logistic regression

- Model

- Estimation

- Application: Heart diseases data

Conclusions

Linear Regression Problem

- ▶ $X_i = (X_{i,1}, \dots, X_{i,p})^T \in \mathbb{R}^p$,
 - ▶ $Y_i \in \mathbb{R}$,
- for $i = 1, \dots, n$ (sized n training set)

Linear Regression Model

$$Y_i = \beta_0 + \sum_{j=1}^p \beta_j X_{i,j} + \sigma \epsilon_i, \quad \text{for } i = 1, \dots, n,$$

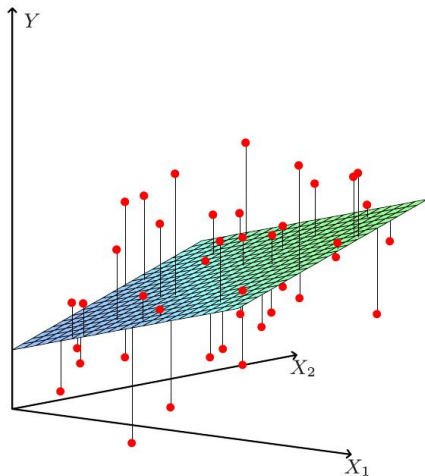
- ▶ ϵ_i is a centered with unit variance ($E[\epsilon_i] = 0$, $\text{var}(\epsilon_i) = 1$) white noise
- ▶ β_0 is the “intercept” (reduces to the ordinate at the origin when $p = 1$)
- ▶ $\beta \equiv (\beta_0, \dots, \beta_p) \in \mathbb{R}^{p+1}$ is the **coefficient vector**

Objective: estimation of β using the samples in the training set \leftarrow supervised learning problem

Remark: model linear w.r.t. $\beta \equiv (\beta_0, \dots, \beta_p) \in \mathbb{R}^{p+1}$, but not necessarily linear w.r.t.

- ▶ the inputs X_i : we can add non linear predictors $h(X_1, \dots, X_p)$ in the model, e.g. X_i^2 , $X_i X_j \dots$
- ▶ the outputs Y_i : we can introduce a non linear link function \leftarrow generalized linear model, e.g. logistic regression

Least Squares (LS) Estimator



Linear least squares fitting with $X \in \mathbb{R}^2$

$$y_i = \beta_0 + \sum_{j=1}^p \beta_j x_{i,j} + \sigma \epsilon_i, \quad \text{for } i = 1, \dots, n,$$

LS estimate defined by minimizing the **Residual Sum of Squares (RSS)**

$$\hat{\beta} = \arg \min_{\beta} \underbrace{\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2}_{\text{RSS}(\beta)}$$

► $\text{RSS}(\beta) \propto$ training error rate for quadratic loss

Least Squares Estimator (Cont'd)

$$\hat{\beta} = \arg \min_{\beta} \text{RSS}(\beta), \quad \text{where } \text{RSS}(\beta) = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{i,j} \right)^2$$

Matrix expression of RSS

$$\text{RSS}(\beta) = \|Y - X\beta\|_2^2,$$

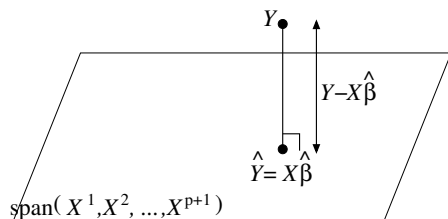
$$\text{where } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n, \quad X = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & \dots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times (p+1)}, \quad \beta = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_p \end{pmatrix} \in \mathbb{R}^{p+1}$$

LS Estimator derivation

$\hat{Y} = X\hat{\beta}$ is the prediction in the space spanned by the column vectors of X such that the euclidean error norm $\|Y - X\hat{\beta}\|_2$ is minimized

Orthogonality principle

Let X^j be the j th column of X



for $j = 1, \dots, p + 1$

$$\begin{aligned}\langle X^j, Y - X\hat{\beta} \rangle &= (X^j)^T (Y - X\hat{\beta}) = 0, \\ \Leftrightarrow X^T (Y - X\hat{\beta}) &= 0, \\ \Leftrightarrow (X^T X) \hat{\beta} &= X^T Y\end{aligned}$$

Rk: This condition can also be derived by setting the gradient of $\text{RSS}(\beta) = (Y - X\beta)^T (Y - X\beta)$ to 0.

LS Estimator computation

Assumption: $\text{rank } X = p + 1$, hence $X^T X$ is invertible

Analytical expression

Because $(X^T X) \hat{\beta} = X^T Y$,

$$\hat{\beta} = (X^T X)^{-1} X^T Y,$$

Numerical computation in high dimension

Pb: When $p > 10^3$ or $p > 10^4$, too expensive to compute $(X^T X)^{-1} \dots$

- 👉 more efficient to use a numerical procedure to minimize the RSS, e.g. **steepest descent** (see next section)

LS Estimator properties

For a known X , $Y = X\beta + \sigma\varepsilon$ where $E[\varepsilon] = 0_n$ and $\text{cov } \varepsilon = I_n$.

- $\hat{\beta}$ is an unbiased estimator of β

$$E[\hat{\beta}] = E \left[(X^T X)^{-1} X^T Y \right] = (X^T X)^{-1} X^T X \beta + (X^T X)^{-1} X^T E[\varepsilon] = \beta$$

- Covariance

$$\text{cov } \hat{\beta} = (X^T X)^{-1} X^T \text{cov}(Y) X (X^T X)^{-1} = \sigma^2 (X^T X)^{-1}$$

- MSE (Power of the estimation error)

$$\begin{aligned} E \left[\|\hat{\beta} - \beta\|^2 \right] &= E \left[(\hat{\beta} - \beta)^T (\hat{\beta} - \beta) \right] = E \left[\text{trace} \left((\hat{\beta} - \beta) (\hat{\beta} - \beta)^T \right) \right], \\ &= \text{trace} \left(\text{cov } \hat{\beta} \right) = \sigma^2 \text{trace} \left((X^T X)^{-1} \right) = \sigma^2 \sum_{j=1}^{p+1} \frac{1}{\lambda_j} \end{aligned}$$

where $\lambda_i > 0$ are the eigenvalues of the symm. def. pos. matrix $X^T X$. What happens for $\lambda_i \approx 0$?

LS Estimator properties (Cont'd)

For a known X , $Y = X\beta + \sigma\varepsilon$ where $E[\varepsilon] = 0_n$ and $\text{cov } \varepsilon = I_n$.

Noise variance estimator

An **unbiased** estimate of the noise variance σ^2 can be deduced as

$$\hat{\sigma}^2 = \frac{1}{n-p-1} \text{RSS}(\hat{\beta}),$$

Gaussian noise $\varepsilon \sim \mathcal{N}(0, I_n)$

- ▶ $\hat{\beta}$ is $\mathcal{N}(\beta, \sigma^2 (X^T X)^{-1})$ distributed,
- ▶ LS estimator $\hat{\beta}$ is also the maximum likelihood estimator

Outline

Reminder on Linear regression

Stochastic Gradient Descent

Regularization and shrinkage methods

- Ridge regression

- Lasso estimator

- Application: prostate data

Logistic regression

- Model

- Estimation

- Application: Heart diseases data

Conclusions

Reminder on Steepest descent, aka gradient descent

We can define the criterion to be minimized as $J(\beta) \equiv \frac{1}{2} \text{RSS}(\beta)$

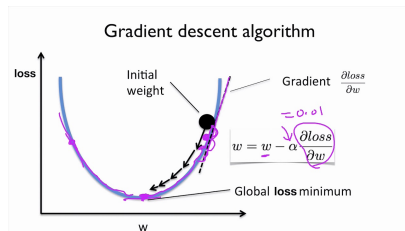
Steepest descent

Ubiquitous iterative procedure based on the observation that $J(\mathbf{w})$ decreases fastest if one goes from \mathbf{w} in the direction of the negative gradient of $J(\cdot)$ at \mathbf{w} , i.e. $-\nabla J(\mathbf{w})$. Here one iteration consists in

$$\beta_{k+1} = \beta_k - \alpha_k \nabla_{\beta} J(\beta_k),$$

where

- ▶ $\alpha_k \in \mathbb{R}$ is the **learning rate**
- ▶ $\nabla_{\beta} J(\beta) = X^T X \beta - X^T Y$ is the **batch gradient** (computed over the whole training set)



Stochastic Gradient Descent (SGD)

Remember: batch gradient $\nabla_{\beta} J(\beta) = X^T X \beta - X^T Y$

Pb: For large and high-dimensional datasets, still too expensive to compute the batch gradient (requires to store and compute $X^T X \dots$)

☞ stochastic approximation of the batch gradient to decrease the computational burden

Stochastic gradient

Descent direction is computed as $\nabla_{\beta} J(\beta) \approx X_i^T X_i \beta - X_i^T Y_i$, for a given sample $i \in \{1, \dots, n\}$ in the training set, where X_i is the *ith line vector* of X .

- ▶ cheaper than batch one for a single iteration, can be much more efficient
- ▶ one loop over all the $i = 1, \dots, n$ training samples is called an **epoch**

Mini-batch SGD

Tradeoff between batch and stochastic gradients:

- ▶ gradient computed on a small subset (**mini-batch**) of the training set,
- ▶ one loop over all the mini-batches (thus over the whole training set) is an **epoch**
- ▶ one epoch is one iteration of the gradient procedure, which is repeated many times to achieve a good minimization

Properties

- ▶ smoother convergence than pure SGD
- ▶ more computationally efficient than batch gradient
- ▶ size b of the mini-batch drives the trade-off ($b = 1$ is pure SGD, $b = n$ is batch gradient). Basically $b = 32, 64$ or 128 .
- 👉 standard optimization procedure for many ML methods (e.g. deep neural nets)

Reminder on Least Squares Estimators (LSE)

Linear regression model

For a sized n training set with p variables (may include the intercept)

$$Y = X\beta + \varepsilon,$$

where

- ▶ $Y \in \mathbb{R}^n$ is the response/output vector,
- ▶ $X \in \mathbb{R}^{n \times p}$ is the data matrix (j th column X^j is the sample vector for j th input variable)
- ▶ $\varepsilon \in \mathbb{R}^n$ is the non-predictible part (noise)
- ▶ $\beta \in \mathbb{R}^p$ are the (unknown) coefficients/weights for the input variables

Least Squares (LS) prediction

For a test data $x \in \mathbb{R}^p$, we predict $\hat{y} = x^T \hat{\beta}$ where the LSE

$$\hat{\beta} = (X^T X)^{-1} X^T y,$$

is the LS fit on the training set

Limitations of Least Squares Estimators (LSE)

Problem

When $\text{rank } X < p$, or when X has singular values close to zero, then $X^T X$ is no more invertible, or ill conditioned (eigenvalues close to zero)...

Causes

- ▶ redundant or nearly-collinear predictors, e.g. $X^k \approx aX^l + b$, where X^j is the j th column of X
- ▶ **high dimensional** problem where $p \approx n$ (or $p > n$)

Effects

no single, or stable, solution for $\hat{\beta}$

- ▶ high variance of $\hat{\beta}$ as an eigenvalue λ_i of $X^T X$ is close to zero ($\|\hat{\beta}\| \rightarrow +\infty$ as $\lambda_i \rightarrow 0$),
- ▶ true error rate explodes since a small perturbation in the training set yields a substantially different estimate $\hat{\beta}$ and prediction rule $\hat{y} = x^T \hat{\beta}$

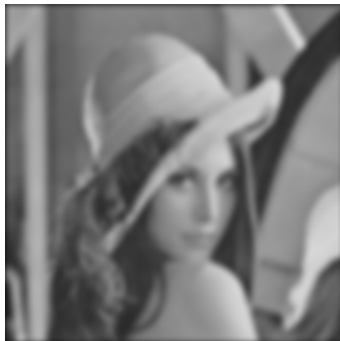
☞ **over-fitting problem**

Instability of LSE: Deconvolution illustration

- ▶ $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- ▶ $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...



$\beta \leftarrow$ original image



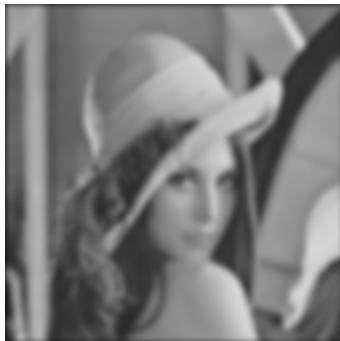
$y = X\beta \leftarrow$ blurred image

Instability of LSE: Deconvolution illustration

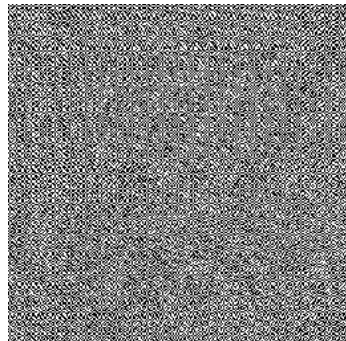
- ▶ $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- ▶ $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...



$\beta \leftarrow$ original image



$y = X\beta \leftarrow$ blurred image



$\hat{\beta}_{=(X^T X)^{-1} X^T y} \leftarrow$ LS estimate

Due to the bad conditioning of $X^T X$ (e.v. close to zero), the noise (here numerical round-off errors) is multiplied by an almost infinite gain, and the estimated coefficients $\hat{\beta}_j$ explode to $\pm\infty$!

Outline

Reminder on Linear regression

Stochastic Gradient Descent

Regularization and shrinkage methods

- Ridge regression

- Lasso estimator

- Application: prostate data

Logistic regression

- Model

- Estimation

- Application: Heart diseases data

Conclusions

Regularization methods

Supplementary materials



Prof. A. Ihler short (8mn) and educational video

<https://www.youtube.com/watch?v=s04ZirJh9ds>



Wikipedia page

https://en.wikipedia.org/wiki/Regularized_least_squares#Specific_examples



Scikit-learn nice documentation with examples (can stop just before section 1.1.4)

https://scikit-learn.org/stable/modules/linear_model.html

Regularization: shrinkage

Idea: introducing a little bias in the estimation of β may lead to a substantial decrease in variance and, hence, in the true error rate

Penalized regression

Regularize the estimation problem by introducing a penalization term for β

$$\tilde{\beta} = \arg \min_{\beta} [\text{RSS}(\beta) + \lambda \text{Pen}(\beta)]$$

- ▶ $\text{RSS}(\beta)$ is the *fidelity term* to the training set (replace with the opposite log-likelihood $-\ell(\beta)$ for generalized linear model, e.g. logistic regression)
- ▶ $\text{Pen}(\beta)$ is the *a priori* to regularize the solution,
- ▶ $\lambda > 0$ is the penalization coefficient

Choosing λ : tradeoff between overfitting (small λ) and underfitting (large λ)

- 👉 standard practice is to use cross-validation to estimate an optimal λ for the test error rate

Ridge regression

Penalization in the (squared) ℓ_2 sense:

$$\text{Pen}(\beta) \equiv \beta^T \beta = \|\beta\|_2^2, \quad \leftarrow \text{Tychonov regularization}$$

$\tilde{\beta}$ is thus obtained by minimizing

$$\begin{aligned} \text{RSS}(\beta) + \lambda \text{Pen}(\beta) &= (Y - X\beta)^T (Y - X\beta) + \lambda \beta^T \beta, \\ &= (\beta - (X^T X + \lambda I)^{-1} X^T Y)^T (X^T X + \lambda I) (\beta - (X^T X + \lambda I)^{-1} X^T Y) + \text{Cst}, \end{aligned}$$

Ridge estimator: $\tilde{\beta} = (X^T X + \lambda I)^{-1} X^T Y$

Remark

similar to LSE, with an additional 'ridge' on the diagonal of $X^T X$

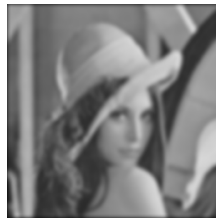
- $X^T X + \lambda I$ has all its eigenvalues greater than $\lambda > 0$, \leftarrow ensures that $\tilde{\beta}$ is always defined, and stable for large enough λ
- 👉 when $\lambda \rightarrow 0$, then $\tilde{\beta} \rightarrow \hat{\beta}$ (over-fitting risk),
- 👉 when $\lambda \rightarrow +\infty$, then $\tilde{\beta} \rightarrow 0$ (under-fitting)

Ridge Regression: deconvolution illustration

- ▶ $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- ▶ $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...



$\beta \leftarrow$ original image



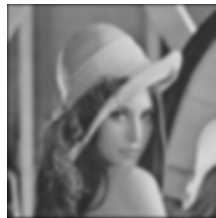
$y = X\beta \leftarrow$ blurred image

Ridge Regression: deconvolution illustration

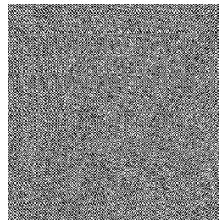
- ▶ $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- ▶ $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...



$\beta \leftarrow$ original image



$y = X\beta \leftarrow$ blurred image



$\hat{\beta}_{(X^T X)^{-1} X^T y} \leftarrow$ LS estimate

Ridge Regression: deconvolution illustration

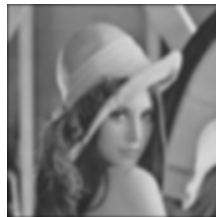
- ▶ $y \in \mathbb{R}^n$ with $n = 256^2$, $\beta \in \mathbb{R}^p$ with $p = 256^2$,
- ▶ $X \in \mathbb{R}^{n \times p} \leftarrow$ sized $(256^2) \times (256^2)$ matrix...



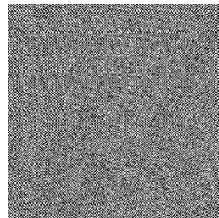
$\beta \leftarrow$ original image



$\tilde{\beta}_{=(X^T X + \lambda I)^{-1} X^T y} \leftarrow$ ridge estimate



$y = X\beta \leftarrow$ blurred image



$\hat{\beta}_{=(X^T X)^{-1} X^T y} \leftarrow$ LS estimate

Regularization by promoting sparsity

Sparse representations/approximations

A representation, or an approximation, is said to be sparse when most of the coefficients are zero

'Bet on Sparsity' principle

Sparsity is a good option in high dimension!

- ▶ if the sparsity assumption does not hold, no method will be able to recover the underlying model in high dimension where $p \approx n$ or $p > n$
- ▶ but if the sparsity assumption holds true, then the parameters can be efficiently estimated by a method that promotes sparsity
- 👉 Occam's razor or KISS (keep it simple, stupid) principles: same idea that simpler models are preferable than more complex ones

Application to the regression problem

choosing a penalization function $\text{Pen}(\beta)$ that promotes the sparsity of β (i.e. with many components $\beta_j = 0$ for $j = 1, \dots, p+1$) \leftarrow Lasso estimator

Lasso ('least absolute shrinkage and selection operator') estimator

Definition

$$\tilde{\beta}_{\text{lasso}} = \arg \min_{\beta} [\text{RSS}(\beta) + \lambda \|\beta\|_1],$$

where $\|\beta\|_1 = \sum_{j=1}^{p+1} |\beta_j|$ is the ℓ_1 norm

- ▶ no analytical expression of $\tilde{\beta}_{\text{lasso}}$
- ▶ but convex optimization problem where very efficient numerical procedures are available to compute $\tilde{\beta}_{\text{lasso}}$

Lasso advantages

Converges to a generally **sparse** solution, i.e. such that $\beta_k = 0$ for a subset of index k

- ☞ the less significant variables are explicitly discarded
- ☞ similar stability than ridge estimator + **variable selection**

Penalization with ℓ_1 and ℓ_2 norms: geometrical interpretation

- ▶ Least Squares estimator: $\hat{\beta} = \arg \min \text{RSS}(\beta)$,
- ▶ Penalized/Regularized estimator: $\tilde{\beta} = \arg \min (\text{RSS}(\beta) + \lambda \text{Pen}(\beta))$
 $\Leftrightarrow \tilde{\beta} = \arg \min \text{RSS}(\beta)$ under the constraint $\text{Pen}(\tilde{\beta}) \leq s(\lambda)$.

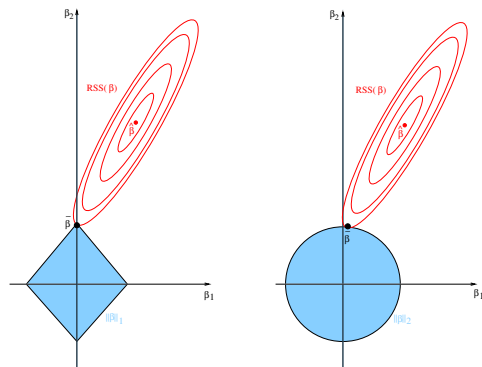


Illustration in dimension $p = 2$: $\beta = (\beta_1, \beta_2)^T$

- ▶ red ellipses are the contour plots of RSS
- ▶ blue "balls" are the constraint sets for
lasso: $\text{Pen}(\beta) = \|\beta\|_1 = |\beta_1| + |\beta_2|$ (**left**),
ridge: $\text{Pen}(\beta) = \|\beta\|_2^2 = \beta_1^2 + \beta_2^2$ (**right**).
- ▶ LSE $\hat{\beta}$ is the center of the red ellipses
- ▶ Penalized LSE $\tilde{\beta}$ is the intersection between red ellipses and blue "ball"
- ☞ Here the RSS mainly varies along β_2 , and we get
 $\tilde{\beta}_1 = 0$ for lasso
 (while $\tilde{\beta}_1 \approx 0$ but not zero for ridge)

ℓ_1 norm promotes the sparsity of the estimator: the less significant predictors are explicitly discarded (coeffs β_k are zero) \leftarrow **model selection**

Scale your data!

- ▶ Linear models (w/o regularization) are invariant under the scaling of the variables: the prediction function is unchanged.
- ▶ Regularized linear models are not due to the penalty term: **scaling of the variables matters!**
- ☞ the variables that have the greatest magnitudes are favoured (same problem for distance based ML methods s.t. K-NN, SVM, ...)

Practical advices

- ▶ If the variables are in different units, scaling each is **strongly recommended**.
- ▶ If they are in the same units, you might or might not scale the variables (depend on your problem)

Usual scaling methods

- ▶ **normalization** in $[0, 1]$: $\tilde{x}_i = \frac{x_i - \min_i}{\max_i - \min_i}$
- ▶ **standardization** to get zero mean and unit variance: $\tilde{x}_i = \frac{x_i - \mu_i}{\sigma_i}$

Application: prostate data

Stamey et al. (1989) study to examine the association between prostate specific antigen (PSA) and several clinical measures that are potentially associated with PSA in men. Objective is to predict the Log PSA (supervised regression problem) from eight variables

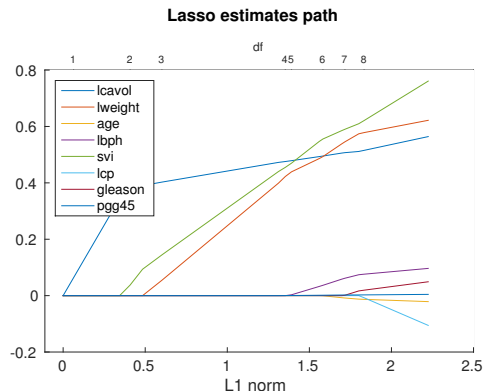
- ▶ lcavol: Log cancer volume
- ▶ lweight: Log prostate weight
- ▶ age: The man's age
- ▶ lbph: Log of the amount of benign hyperplasia
- ▶ svi: Seminal vesicle invasion; 1=Yes, 0=No
- ▶ lcp: Log of capsular penetration
- ▶ gleason: Gleason score
- ▶ pgg45: Percent of Gleason scores 4 or 5

Application : prostate data

Lasso estimate (ℓ_1 -penalization): $\tilde{\beta}(\lambda) = \arg \min_{\beta} \text{RSS}(\beta) + \lambda \|\beta\|_1$,

Lasso path: We can plot the estimated variable coeffs $\tilde{\beta}(\lambda)_j$ vs λ , or equivalently vs $\|\tilde{\beta}(\lambda)\|_1$

- For large λ all the coefficients are zeros ($\|\tilde{\beta}(\lambda)\|_1 = 0$)
- When $\lambda \searrow$ then $\|\tilde{\beta}(\lambda)\|_1 \nearrow$: most significant variables sequentially enter the model (non-zero coeffs)

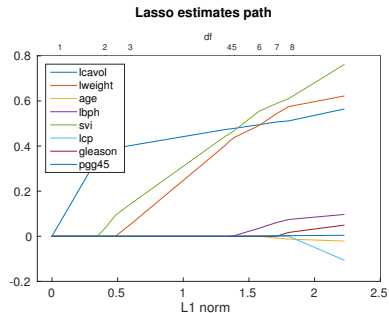
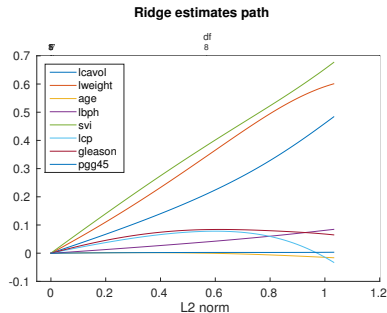


Choosing λ

- large $\|\tilde{\beta}(\lambda)\|_1$ (small λ) \rightarrow overfitting
 - small $\|\tilde{\beta}(\lambda)\|_1$ (large λ) \rightarrow underfitting
 - cross-validation estimation of λ yields $\|\tilde{\beta}(\lambda)\|_1 = 1.06$ ($\lambda = 0.21$)
- \Rightarrow only 3 predictors enter the model to predict PSA: **lcavol**, **svi**, **lweight**

Application : prostate data

Comparison of ridge and lasso estimators



Path of the penalized coefficients as a function of $\|\tilde{\beta}(\lambda)\|$

- ▶ Ridge estimates are **smooth** functions of λ , with coefficients that are never stuck at zero.
- ▶ Lasso estimates are **piecewise linear** functions, with a kink each time a new variable enter the model
- ▶ **Shrinkage effect**: the larger λ , the more the coefficients are shrunk toward 0 for both penalties
- ▶ For small λ , thus large $\|\tilde{\beta}(\lambda)\|$, both estimator becomes equivalent (convergence toward LSE)

Outline

Reminder on Linear regression

Stochastic Gradient Descent

Regularization and shrinkage methods

Ridge regression

Lasso estimator

Application: prostate data

Logistic regression

Model

Estimation

Application: Heart diseases data

Conclusions

Discriminative model for classification: $Y \in \mathcal{Y} \leftarrow$ discrete set

Discriminative model

For a given $X = x$, we want to model directly

$$\Pr(Y = k | X = x)$$

for each value of the class label $k \in \mathcal{Y}$

- ▶ do not require to specify the marginal distribution of the inputs X

Model-based classification rule

We predict the class with the **highest probability**

$$\hat{Y} = \arg \max_k \Pr(Y = k | X = x)$$

- ▶ this is the optimal rule for misclassification rate referred to as *Bayes Classifier*... if the model is true (of course this is not the case, but it may be useful)!

How can we use linear regression to model a probability $\Pr(Y = k | X = x)$?

Linear model for classification: Logistic regression (LR)

Classification problem $Y \in \mathcal{Y} \leftarrow$ discrete set

Binary classification problem: $\mathcal{Y} = \{1, 2\}$

Consider the following model

$$\Pr(Y_i = 1 | X_i = x_i) = \phi(x_i^T \beta) = \frac{\exp(x_i^T \beta)}{1 + \exp(x_i^T \beta)},$$

where

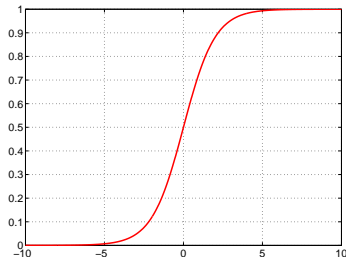
- ▶ $x_i = (\mathbf{1}, x_{i,1}, \dots, x_{i,p})^T \in \mathbb{R}^{p+1} \leftarrow$ **intercept** term included by default,
- ▶ ϕ is the **logistic** function: maps a real value to a probability

Multiclass problem: $\mathcal{Y} = \{1, 2, \dots, K\}$

logistic model can be easily extended to the multiclass problem: **multinomial** logistic regression

$$\phi : \mathbb{R} \rightarrow (0, 1)$$

$$u \mapsto \frac{\exp u}{1 + \exp u} = \frac{1}{1 + \exp(-u)}.$$



LR is a generalized linear model

Consider

- ▶ $p_i \equiv \Pr(Y_i = 1 | X_i = x_i) = \phi(x_i^T \beta)$
- ▶ $\phi^{-1} : p \in (0, 1) \mapsto \log \frac{p}{1-p} \in \mathbb{R}$ is the **logit** function

Generalized linear model

- ▶ **Linear** equation w.r.t. β :

$$\text{logit}(p_i) = x_i^T \beta,$$

- ▶ + additional nonlinear constraint (proba sum to 1):

$$\Pr(Y_i = 2 | X_i = x_i) = 1 - p_i = \frac{1}{1 + \exp(x_i^T \beta)}$$

Logistic regression for classification in K classes

When $\mathcal{Y} = \{1, \dots, K\} \leftarrow K$ classes, the models becomes

$$\begin{aligned} \log \frac{\Pr(Y_i=1|X_i=x_i)}{\Pr(Y_i=K|X_i=x_i)} &= x_i^T \beta_1, \\ \log \frac{\Pr(Y_i=2|X_i=x_i)}{\Pr(Y_i=K|X_i=x_i)} &= x_i^T \beta_2, \\ &\vdots \\ \log \frac{\Pr(Y_i=K-1|X_i=x_i)}{\Pr(Y_i=K|X_i=x_i)} &= x_i^T \beta_{K-1}, \end{aligned}$$

👉 $K - 1$ equations + sum-to-one constraint on the probabilities

Multiclass case: logistic equations

$$\begin{aligned} \Pr(Y_i = k|X_i = x_i) &= \frac{\exp(x_i^T \beta_k)}{1 + \sum_{l=1}^{K-1} \exp(x_i^T \beta_l)}, \quad \text{for } k = 1, \dots, K-1, \\ \Pr(Y_i = K|X_i = x_i) &= \frac{1}{1 + \sum_{l=1}^{K-1} \exp(x_i^T \beta_l)} \end{aligned}$$

Parameter estimation

Since the distribution of $Y|X$ is known, the log-likelihood expresses as

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^n \log p_{X,Y}(x_i, y_i | \beta), \\ &= \sum_{i=1}^n \log p_{Y|X}(y_i | x_i, \beta) + \sum_{i=1}^n \log p(x_i), \quad \leftarrow \text{Bayes rule}\end{aligned}$$

where the second term is a constant that does not depend on β

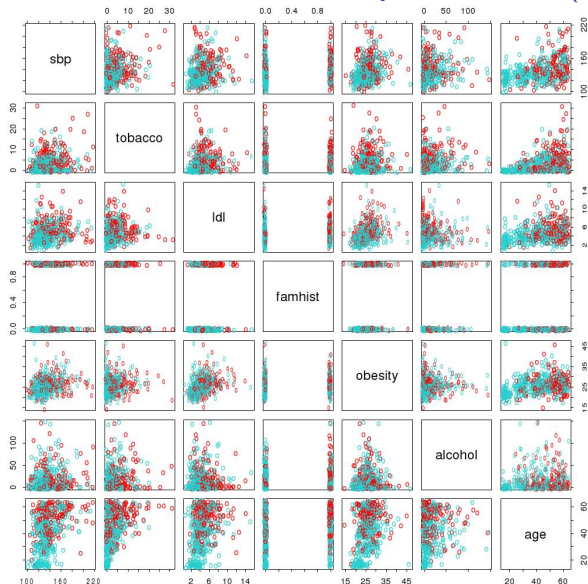
Maximum likelihood estimator

Maximizing the log-likelihood w.r.t $\beta \Leftrightarrow$ Maximizing the conditional log-likelihood

$$\beta \mapsto \sum_{i=1}^n \log p_{Y|X}(y_i | x_i, \beta)$$

- ▶ no analytical expression of the ML estimator,
- ▶ numerical computation usually performed by a Newton-Raphson procedure but more appropriate/efficient to use [Stochastic Gradient Descent](#) especially for regularized estimation.
- ▶ Rk: [regularized estimator](#) defined as $\tilde{\beta} = \arg \min_{\beta} -\ell(\beta) + \lambda \text{Pen}(\beta)$

Application: South African coronary heart disease (CHD)



A retrospective sample of males in a coronary heart-disease (CHD) high-risk region of the Western Cape, South Africa.

Matrix of the predictor scatterplots

- ▶ each plot \equiv pair of risk factors
- ▶ Here 7 predictors:
 - ▶ *sbp*: systolic blood pressure,
 - ▶ *tobacco*: cumulative tobacco consumption (kg),
 - ▶ *ldl*: \sim cholesterol,
 - ▶ *famhist*: family history of heart disease (Present, Absent)
 - ▶ *obesity*: quantitative indicator,
 - ▶ *alcohol*: current alcohol consumption
 - ▶ *age*: age at onset
- ▶ response: CHD event (**case**) or not (**control**)
- ▶ 160 **cases** / 302 **controls**

Application: South African CHD (Cont'd)

Logistic regression fit of CHD events

	Coefficient	Std. Error	Z score
(Intercept)	-4.130	0.964	-4.285
sbp	0.006	0.006	1.023
tobacco	0.080	0.026	3.034
ldl	0.185	0.057	3.219
famhist	0.939	0.225	4.178
obesity	-0.035	0.029	-1.187
alcohol	0.001	0.004	0.136
age	0.043	0.010	4.184

- ▶ A Z score ($\equiv \text{Coeff} / \text{Std. Error}$) > 2 in absolute value is significant at the 5% level.

Must be interpreted with caution!

- ▶ systolic blood pressure (sbp) is not significant!
- ▶ nor is obesity (conversely, < 0 coefficient)!
- result of the **strong correlations** between the predictors: **over-fitting** issue !

Application: South African CHD (Cont'd) with greedy selection procedure

Model selection: greedy backward procedure

To prevent from over-fitting, find the variables that are sufficient for explaining the CHD outputs

- ▶ drop the least significant predictor, and refit the model
- ▶ repeat until no further terms can be dropped ← **backward selection**

Logistic regression fit with backward model selection procedure

	Coefficient	Std. Error	Z score
(Intercept)	-4.204	0.498	-8.45
tobacco	0.081	0.026	3.16
ldl	0.168	0.054	3.09
famhist	0.924	0.223	4.14
age	0.044	0.010	4.52

Interpretations

- ▶ Tobacco is measured in total lifetime usage in kilograms, with a median of 1kg for the controls and 4.1kg for the cases
- ▶ An increase of 1kg \Rightarrow increase of the CHD proba of $\exp(0.081) = 1.084$ or 8.4% (confidence interval at 95% [1.03, 1.14])

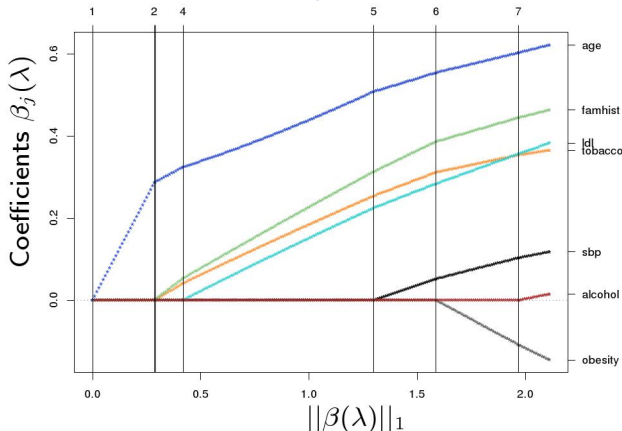
Application: South African CHD (Cont'd) with lasso selection procedure

Model selection: ℓ_1 penalization (Lasso type method)

$$\tilde{\beta}(\lambda) = \arg \min_{\beta} -\ell(\beta) + \lambda \|\beta\|_1,$$

→ function of λ where less significant variables are explicitly discarded

Path of the des coefficients ℓ_1 -penalized coefficients as a function of $\|\hat{\beta}(\lambda)\|_1$



Choosing λ

- ▶ large $\|\tilde{\beta}(\lambda)\|_1$ (small λ) → over-fitting
- ▶ small $\|\tilde{\beta}(\lambda)\|_1$ (large λ) → under-fitting
- ▶ $0.43 \leq \|\tilde{\beta}(\lambda)\|_1 \leq 1.3 \rightarrow$
4 same predictors than backward selection procedure

Outline

Reminder on Linear regression

Stochastic Gradient Descent

Regularization and shrinkage methods

- Ridge regression

- Lasso estimator

- Application: prostate data

Logistic regression

- Model

- Estimation

- Application: Heart diseases data

Conclusions

Conclusions

Generalized Linear Models

Learning of the prediction rule based on a model of Y given X

- ☞ Linear regression, Logistic regression

Properties

- ▶ Simplicity: useful to capture the main effects
- ▶ Interpretability
- ▶ Efficient numerical procedures for large or high-dimensional data

Conclusions on Regularization for linear models

Regularization procedures are essential tools for data analysis, especially for big datasets involving many predictors, to

- ▶ prevent for over-fitting,
- ▶ better interpret the relations between the variables,
- ▶ improve the prediction performance

Shrinkage procedures

- ▶ ℓ_2 (ridge) regularization promotes the **simplicity**: shrink all the coefficients toward 0
- ▶ ℓ_1 (lasso) regularization promotes the **simplicity+sparsity**: shrink all the coefficients toward 0 + coefficients of non-significant enough variables exactly equal to 0
- ▶ useful to capture the main effects and to interpret the relations between the variables
- 👉 concepts that extend to non-linear methods, e.g. neural nets