Introduction to Statistical Learning

Jean-Philippe Vert
Jean-Philippe.Vert@ensmp.fr

Mines ParisTech and Institut Curie

Master Course, 2011.

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Motivations

- Predict the risk of second heart from demographic, diet and clinical measurements
- Predict the future price of a stock from company performance measures
- Recognize a ZIP code from an image
- Identify the risk factors for prostate cancer

and many more applications in many areas of science, finance and industry where a lot of data are collected.

Learning from data

- Supervised learning
  - An outcome measurement (target or response variable)
  - which can be quantitative (regression) or categorial (classification)
  - which we want to predicted based on a set of features or descriptors or predictors)
  - We have a training set with features and outcome
  - We build a prediction model, or learner to predict outcome from features for new unseen objects
- Unsupervised learning
  - No outcome
  - Describe how data are organized or clustered
- *Examples - Fig 1.1-1.3*

Learning from data

- Supervised learning
  - An outcome measurement (target or response variable)
  - which can be quantitative (regression) or categorial (classification)
  - which we want to predicted based on a set of features or descriptors or predictors)
  - We have a training set with features and outcome
  - We build a prediction model, or learner to predict outcome from features for new unseen objects
- Unsupervised learning
  - No outcome
  - Describe how data are organized or clustered
- *Examples - Fig 1.1-1.3*

Learning from data

- Supervised learning
  - An outcome measurement (target or response variable)
  - which can be quantitative (regression) or categorial (classification)
  - which we want to predicted based on a set of features or descriptors or predictors)
  - We have a training set with features and outcome
  - We build a prediction model, or learner to predict outcome from features for new unseen objects
- Unsupervised learning
  - No outcome
  - Describe how data are organized or clustered
- *Examples - Fig 1.1-1.3*

Machine learning / data mining vs statistics

They share many concepts and tools, but in ML:

- Prediction is more important than modelling (understanding, causality)
- There is no settled philosophy or theoretical framework
- We are ready to use ad hoc methods if they seem to work on real data
- We often have many features, and sometimes large training sets.
- We focus on efficient algorithms, with little or no human intervention.
- We often use complex nonlinear models dfs

Organization

- Focus on supervised learning (regression and classification)
- Reference: "The Elements of Statistical Learning" by Hastie, Tibshirani and Friedman (HTF)
- Available online at `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`
- Practical sessions using `R`

Notations

- $Y \in \mathcal{Y}$ the response (usually $\mathcal{Y} = \{-1, 1\}$ or $\mathbb{R}$)
- $X \in \mathcal{X}$ the input (usually $\mathcal{X} = \mathbb{R}^p$)
- $x_1, \ldots, x_N$ observed inputs, stored in the $N \times p$ matrix $\mathbf{X}$
- $y_1, \ldots, y_N$ observed inputs, stored in the vector $\mathbf{Y} \in \mathcal{Y}^N$

Simple method 1: Linear least squares

- Parametric model for $\beta \in \mathbb{R}^{p+1}$:

$$f_\beta(X) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i = X^\top \beta$$

- Estimate $\hat{\beta}$ from training data to minimize

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f_\beta(x_i))^2$$

- See Fig 2.1
- Good if model is correct...

Simple method 2: Nearest neighbor methods (k-NN)

- Prediction based on the k nearest neighbors:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

- Depends on k
- Less assumptions that linear regression, but more risk of overfitting
- *Fig 2.2-2.4*

Statistical decision theory

- Joint distribution $Pr(X, Y)$
- Loss function $L(Y, f(X))$, e.g. squared error loss

$$L(Y, f(X)) = (Y - f(X))^2$$

- Expected prediction error (EPE):

$$EPE(f) = E_{(X,Y)\sim Pr(X,Y)} L(Y, f(X))$$

- Minimizer is $f(X) = E(Y \mid X)$ (regression function)
- Bayes classifier for $0/1$ loss in classification (*Fig 2.5*)

Least squares and k-NN

- Least squares assumes $f(x)$ is linear, and pools over values of X to estimate the best parameters. Stable but biased
- k-NN assumes $f(x)$ is well approximated by a locally constant function, and pools over local sample data to approximate conditional expectation. Less stable but less biased.

Local methods in high dimension

- If N is large enough, k-NN seems always optimal (universally consistent)
- But when p is large, curse of dimension:
  - No method can be "local' (*Fig 2.6*)
  - Training samples sparsely populate the input space, which can lead to large bias or variance (*eq. 2.25 and Fig 2.7-2.8*)
- If structure is known (eg, linear regression function), we can reduce both variance and bias (*Fig. 2.9*)

Bias-variance trade-off

Assume $Y = f(X) + \epsilon$, on a fixed design. $Y(x)$ is random because of ϵ, $\hat{f}(X)$ is random because of variations in the training set $\mathcal{T}$. Then

$$\begin{aligned} E_{\epsilon,\mathcal{T}}\left(Y - \hat{f}(X)\right)^2 &= EY^2 + E\hat{f}(X)^2 - 2EY\hat{f}(X) \\ &= Var(Y) + Var(\hat{f}(X)) + \left(EY - E\hat{f}(X)\right)^2 \\ &= noise + bias(\hat{f})^2 + variance(\hat{f}) \end{aligned}$$

Structured regression and model selection

Define a family of function classes $\mathcal{F}_\lambda$, where λ controls the "complexity", eg:

- Ball of radius λ in a metric function space
- Bandwidth of the kernel is a kernel estimator
- Number of basis functions

For each λ, define

$$\hat{f}_\lambda = \operatorname*{argmin}_{\mathcal{F}_\lambda} EPE(f)$$

Select $\hat{f} = \hat{f}_{\hat{\lambda}}$ to minimize the bias-variance tradeoff (*Fig. 2.11*).

Cross-validation

A simple and systematic procedure to estimate the risk (and to optimize the model's parameters)

1. Randomly divide the training set (of size N) into K (almost) equal portions, each of size K/N
2. For each portion, fit the model with different parameters on the $K-1$ other groups and test its performance on the left-out group
3. Average performance over the K groups, and take the parameter with the smallest average performance.

Taking $K = 5$ or 10 is recommended as a good default choice.

Summary

To learn complex functions in high dimension from limited training sets, we need to optimize a bias-variance trade-off. We will do that typically by:

1. Define a family of learners of various complexities (eg, dimension of a linear predictor)
2. Define an estimation procedure for each learner (eg, least-squares or empirical risk minimization)
3. Define a procedure to tune the complexity of the learner (eg, cross-validation)

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Linear least squares

- Parametric model for $\beta \in \mathbb{R}^{p+1}$:

$$f_\beta(X) = \beta_0 + \sum_{i=1}^{p} \beta_i X_i = X^\top \beta$$

- Estimate $\hat{\beta}$ from training data to minimize

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f_\beta(x_i))^2$$

- Solution if $\mathbf{X}^\top \mathbf{X}$ is non-singular:

$$\hat{\beta} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}$$

Fitted values

- Fitted values on the training set:

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{Y} = \mathbf{H}\mathbf{Y} \quad \text{with} \quad \mathbf{H} = \mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top$$

- Geometrically: **H** projects **Y** on the span of **X** *(Fig. 3.2)*
- If **X** is singular, $\hat{\beta}$ is not uniquely defined, but $\hat{\mathbf{Y}}$ is

Inference on coefficients

- Assume $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$
- Then $\hat{\beta} \sim \mathcal{N}\left(\beta, \sigma^2 \left(\mathbf{X}^\top \mathbf{X}\right) - 1\right)$
- Estimating variance: $\hat{\sigma} = \| \mathbf{Y} - \hat{\mathbf{Y}} \|^2 / (N - p - 1)$
- Statistics on coefficients:

$$\frac{\hat{\beta}_j - \beta_j}{\hat{\sigma}\sqrt{v_j}} \sim t_{N-p-1}$$

allows to test the hypothesis $H_0 : \beta_j = 0$, and gives confidence intervals

$$\hat{\beta}_j \pm t_{\alpha/2, N-p-1} \hat{\sigma} \sqrt{v_j}$$

Inference on the model

Compare a large model with p_1 features to a smaller model with p_0 features:

$$F = \frac{(RSS_0 - RSS_1) / (p_1 - p_0)}{RSS_1 / (N - p_1 - 1)}$$

follows the Fisher law $F_{p_1 - p_0, N - p_1 - 1}$ under the hypothesis that the small model is correct.

Gauss-Markov theorem

Assume $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, where $E\epsilon = 0$ and $E\epsilon\epsilon^\top = \sigma^2 I$.
Then the least squares estimator $\hat{\beta}$ is BLUE (best linear unbiased estimator), i.e., for any other estimator $\tilde{\beta} = C\mathbf{Y}$ with $E\tilde{\beta} = \beta$,

$$Var(\hat{\beta}) \leq Var\tilde{\beta}$$

Nevertheless, we may have smaller total risk by increasing bias to decrease variance, in particular in the high-dimensional setting.

Gauss-Markov theorem

Assume $\mathbf{Y} = \mathbf{X}\beta + \epsilon$, where $E\epsilon = 0$ and $E\epsilon\epsilon^\top = \sigma^2 I$.
Then the least squares estimator $\hat{\beta}$ is BLUE (best linear unbiased estimator), i.e., for any other estimator $\tilde{\beta} = C\mathbf{Y}$ with $E\tilde{\beta} = \beta$,

$$Var(\hat{\beta}) \leq Var\tilde{\beta}$$

Nevertheless, we may have smaller total risk by increasing bias to decrease variance, in particular in the high-dimensional setting.

Decreasing the complexity of linear models

1. Feature subset selection
2. Penalized criterion
3. Feature construction

Feature subset selection

- Best subset selection
  - Usually NP-hard, "leaps and bound" procedure works for up to $p = 40$
  - Best k selected by cross-validation of various criteria *(Fig 3.5)*
- Greedy selection: forward, backward, hybrid

Ridge regression

- Minimize
$$RSS(\beta) + \lambda \sum_{i=1}^{p} \beta_i^2$$
- Solution:
$$\hat{\beta}^\lambda = \left(\mathbf{X}^\top \mathbf{X} + \lambda I\right)^{-1} \mathbf{X}^\top \mathbf{Y}$$
- If $\mathbf{X}^\top \mathbf{X} = I$ (orthogonal design), then $\hat{\beta}^\lambda = \hat{\beta}/(1+\lambda)$, otherwise nonlinear solution path *Fig 3.8*
- Equivalent to shrinking on the small principal components *(Fig 3.9)*

Lasso

- Minimize

$$RSS(\beta) + \lambda \sum_{i=1}^{p} |\beta_i|$$

- No explicit solution, but convex quadratic program and efficient algorithm for the solution path (LARS, *Fig. 3.10*)
- Performs feature selection because the ℓ_1 ball has singularities *(Fig 3.11*

Discussion

- In orthogonal design, best subset selection, ridge regression and Lasso correspond to 3 different ways to shrink the $\hat{\beta}$ coefficients *(Fig 3.10)*
- They minimize $RSS(\beta)$ over respectively the ℓ_0, ℓ_2 and ℓ_1 balls
- Generalization: penalize by $\| \beta \|_q$, but:
  - convex problem only for $q \geq 1$
  - feature selection only for $q \leq 1$
- Generalization: group lasso, fused lasso, elastic net...

Using derived input space

- PCR
  - OLS on the top M principal components
  - Similar to ridge regression, but truncates instead of shrinking
- PLS
  - Similar to PCR but uses $\mathbf{Y}$ to construct the directions: maximize

$$\max_{\alpha} Corr^2(\mathbf{Y}, \mathbf{X}\alpha) Var(\mathbf{X}\alpha)$$

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Supervised classification

- $\mathcal{Y} = \{-1, 1\}$ (can be generalized to K classes)
- Goal: estimate $P(Y = k \,|\, X = x)$, or (easier) $\hat{Y}(x) = \arg\max_k P(Y = k \,|\, X = x)$
- Approach: estimate a function $f : \mathcal{X} \to \mathbb{R}$ and predict according to

$$\hat{Y}(x) = \begin{cases} 1 & \text{if } f(x) \geq 0\,, \\ -1 & \text{if } f(x) < 0\,. \end{cases}$$

- 3 strategies
  1. Model $P(X, Y)$ (LDA)
  2. Model $P(Y \,|\, X)$ (logistic regression)
  3. Separate positives from negative examples (SVM)

Linear discriminant analysis (LDA)

- Model $P(Y = k) = \pi_k$ and $P(X \mid Y = k) \sim \mathcal{N}(\mu_k, \Sigma)$
- Estimation:

$$\hat{\pi}_k = \frac{N_k}{N}$$

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{i:y_i=k} x_i$$

$$\hat{\Sigma} = \frac{1}{N-1} \sum_{k \in \{-1,1\}} \sum_{i:y_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top$$

- Prediction:

$$\ln \frac{P(Y = 1 \mid X = x)}{P(Y = -1 \mid X = x)} =$$

$$x^\top \hat{\Sigma}^{-1} (\mu_1 - \mu_{-1}) - \frac{1}{2} \hat{\mu}_2^\top \hat{\Sigma}^{-1} \hat{\mu}_2 + \frac{1}{2} \hat{\mu}_1^\top \hat{\Sigma}^{-1} \hat{\mu}_1 + \ln \frac{N_1}{N_2}$$

Remarks on LDA

- If a $\hat{\Sigma}$ is estimated on each class, we obtain a quadratic function : quadratic discriminant analysis (QDA)
- LDA performs linear discrimination $f(X) = \beta^\top X + b$. β can also be found by OLS, taking $Y_i = N_i/N$
- Good baseline method, even if the data are not Gaussian

Quadratic discriminant analysis (QDA)

- Model $P(Y = k) = \pi_k$ and $P(X \mid Y = k) \sim \mathcal{N}(\mu_k, \Sigma)$
- Estimation: same as LDA except

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{i:y_i=k} (x_i - \mu_k)(x_i - \mu_k)^\top$$

- Prediction:

$$\ln \frac{P(Y = k \mid X = x)}{P(Y = l \mid X = x)} = \delta_k(x) - \delta_l(x)$$

with

$$\delta_k(x) = -\frac{1}{2} \ln |\Sigma_k| - \frac{1}{2} (x - \mu_k)^\top \Sigma_k^{-1} (x - \mu_k) + \ln \pi_k$$

Logistic regression

- Model:

$$\begin{cases} P(Y=1 \,|\, X=x) = \frac{e^{\beta^\top x}}{1+e^{\beta^\top x}} \\ P(Y=-1 \,|\, X=x) = \frac{1}{1+e^{\beta^\top x}} \end{cases}$$

- Equivalently

$$P(Y=y \,|\, X=x) = \frac{1}{1+e^{-y\beta^\top x}}$$

- Equivalently,

$$\ln \frac{P(Y=1 \,|\, X=x)}{P(Y=-1 \,|\, X=x)} = \beta^\top x$$

Logistic regression: parameter estimation

- Likelihood:

$$\ell(\beta) = -\sum_{i=1}^{N} \ln\left(1 + e^{-y_i \beta^\top x_i}\right)$$

$$\frac{\partial \ell}{\partial \beta}(\beta) = \sum_{i=1}^{N} \frac{y_i x_i}{1 + e^{y_i \beta^\top x_i}} = \sum_{i=1}^{N} y_i p(-y_i \mid x_i) x_i$$

$$\frac{\partial^2 \ell}{\partial \beta \partial \beta^\top}(\beta) = -\sum_{i=1}^{N} \frac{x_i x_i^\top e^{\beta^\top x_i}}{\left(1 + e^{\beta^\top x_i}\right)^2} = \sum_{i=1}^{N} p(1 \mid x_i)\left(1 - p(1 \mid x_i)\right) x_i x_i^\top$$

- Optimization by Newton-Raphson is iteratively reweighted least squares (IRLS)
- Problem if data linearly separable $\Longrightarrow$ regularization

Regularized logistic regression

- Problem if data linearly separable : infinite likelihood possible
- Classical ℓ_2 regularization

$$\min_{\beta} \sum_{i=1}^{N} \ln\left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \sum_{i=1}^{p} \beta_i^2$$

- ℓ_1 regularization (feature selection)

$$\min_{\beta} \sum_{i=1}^{N} \ln\left(1 + e^{-y_i \beta^\top x_i}\right) + \lambda \sum_{i=1}^{p} |\beta_i|$$

LDA vs Logistic regression

- Both methods are linear
- Estimation is different: model $P(X, Y)$ (likelihood) or $P(Y \mid X)$ (conditional likelihood)
- LDA works better if data are Gaussian, but more sensitive to outliers

Hard-margin SVM

- If data are linearly separable, separate them with largest margin
- Equivalently, $\min_\beta \| \beta \|^2$ such that $y_i \beta^\top x_i \geq 1$
- Dual problem:

$$\max_{\alpha \geq 0} \sum_{i=1}^N \alpha_i - \frac{1}{1} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

and

$$\hat{\beta}_i = \sum_{i=1}^N y_i \alpha_i x_i$$

Soft-margin SVM

- If data are not linearly separable, add slack variable: $\min_\beta \| \beta \|^2/2 + C \sum_{i=1}^N \zeta_i$ such that $y_i \beta^\top x_i \geq 1 - \zeta_i$
- Dual problem: same as hard-margin with the additional constraint $0 \leq \alpha \leq C$
- Equivalently,

$$\min_\beta \sum_{i=1}^N \max(0, 1 - y_i \beta^\top x_i) + \lambda \| \beta \|^2$$

Large-margin classifiers

- The margin is $yf(x)$
- LDA, logistic and SVM all try to ensure large margin:

$$\min_{\beta} \sum_{i=1}^{N} \phi(y_i f(x_i)) + \lambda \Omega(\beta)$$

where

$$\phi(u) = \begin{cases} (1-u)^2 & \text{for LDA} \\ \ln(1+e^{-}u) & \text{for logistic regression} \\ \max(0, 1-u) & \text{for SVM} \end{cases}$$

Outline

1. Introduction
2. Linear methods for regression
3. Linear methods for classification
4. Nonlinear methods with positive definite kernels

Feature expansion

- We have seen many linear methods for regression and classification, of the form

$$\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{N} L\left(y_i, \beta^\top x_i\right) + \lambda \| \beta \|_2^2$$

- To be nonlinear in x, we can apply them after some transformation $x \mapsto \Phi(x) \in \mathbb{R}^q$, where q may be larger than p
- Example: nonlinear functions of x, polynomials, ...
- Notation: we define the kernel corresponding to Φ by

$$K(x, x') = \Phi(x)^\top \Phi(x')$$

Representer theorem

For any solution of

$$\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{N} L\left(y_i, \beta^\top x_i\right) + \lambda \| \beta \|_2^2$$

there exists $\hat{\alpha} \in \mathbb{R}^n$ such that

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i \Phi(x_i) \,.$$

Consequences:

$$\hat{f}(x) = \sum_{i=1}^{N} \hat{\alpha}_i K(x_i, x)$$

Solving in α

- $\left(f(x_i)\right)_{i=1,\dots,N}^{\top} = K\alpha$ and $\| \beta \|_2^2 = \alpha^{\top} K \alpha$, so we can plug α in the optimization problem instead of β, and only K is needed
- Example: kernel ridge regression:

$$\hat{\alpha} = (K + \lambda I)^{-1} Y$$

- Example: kernel SVM

$$\max_{0 \leq \alpha \leq C} \sum_{i=1}^{N} \alpha_i y_i - \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j K(x_i, x_j)$$

Positive definite kernel

Theorem (Aronszajn)

There exists $\Phi : \mathbb{R}^p \to \mathbb{R}^q$ for some q (possibly infinite if and only if K is positive definite, i.e., $K(x, x') = K(x', x)$ for any x, x', and

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_i K(x_i, x_j) \geq 0$$

for any n, a and x.

Examples:

- Linear: $K(x, x') = x^\top x'$
- Polynomial: $K(x, x') = (x^\top x')^d$
- Gaussian: $K(x, x') = \exp - \| x - x' \|^2 / 2\sigma^2$