Intégration de données

Introduction to Supervised Learning

Ana Karina Fermin

Université Paris Nanterre

M2 Miage APP

http://fermin.perso.math.cnrs.fr/

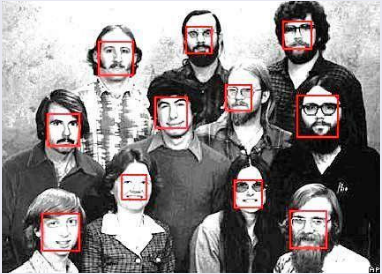
Credit Default, Credit Score, Bank Risk, Market Risk Management

- Data: Client profile, Client credit history...
- Input: Client profile
- Output: Credit risk

Spam detection (Text classification)

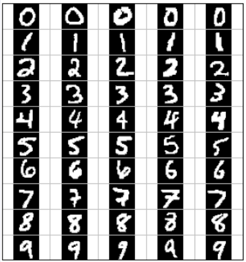
- Data: email collection
- Input: email
- Output : Spam or No Spam

Face Detection

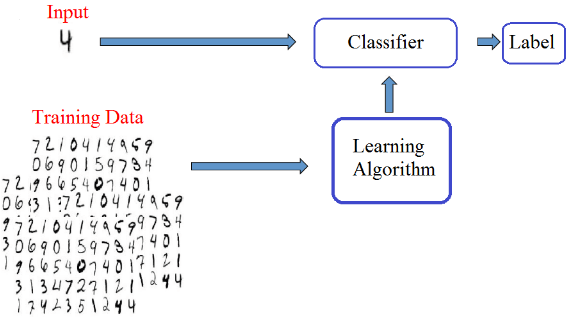


- Data: Annotated database of images
- Input : Sub window in the image
- Output : Presence or no of a face...

Number Recognition



- Data: Annotated database of images (each image is represented by a vector of $28 \times 28 = 784$ pixel intensities)
- Input: Image
- Output: Corresponding number



A definition by Tom Mitchell (`http://www.cs.cmu.edu/~tom/`)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Supervised Learning Framework

- Input measurement $\mathbf{X} = (X^{(1)}, X^{(2)}, \dots, X^{(d)}) \in \mathcal{X}$
- Output measurement $Y \in \mathcal{Y}$.
- $(\mathbf{X}, Y) \sim \mathbf{P}$ with $\mathbf{P}$ unknown.
- Training data : $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- Often
  - $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \{-1, 1\}$ (classification)
  - or $\mathbf{X} \in \mathbb{R}^d$ and $Y \in \mathbb{R}$ (regression).

- A classifier is a function in $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$

Goal

- Construct a good classifier $\widehat{f}$ from the training data.

- Need to specify the meaning of good.
- Formally, classification and regression are the same problem!

Loss function

- Loss function : $\ell(f(x), y)$ measure how well $f(x)$ "predicts" y.
- Examples:
  - Prediction loss: $\ell(Y, f(\mathbf{X})) = \mathbf{1}_{Y \neq f(\mathbf{X})}$
  - Quadratic loss: $\ell(Y, \mathbf{X}) = |Y - f(\mathbf{X})|^2$

Risk of a generic classifier

- Risk measured as the average loss for a new couple:

$$\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right]$$

- Examples:
  - Prediction loss: $\mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{P}\left\{Y \neq f(\mathbf{X})\right\}$
  - Quadratic loss: $\mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right] = \mathbb{E}\left[|Y - f(\mathbf{X})|^2\right]$

- **Beware:** As $\widehat{f}$ depends on $\mathcal{D}_n$

Experience, Task and Performance measure

- Training data : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- Predictor: $f : \mathcal{X} \to \mathcal{Y}$
- Cost/Loss function : $\ell(f(\mathbf{X}), Y)$
- Risk: $\mathcal{R}(f) = \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right]$

- Often $\ell(f(\mathbf{X}), Y) = |f(\mathbf{X}) - Y|^2$ or $\ell(f(\mathbf{X}), Y) = \mathbf{1}_{Y \neq f(\mathbf{X})}$

Goal

- Learn a rule to construct a classifier $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. the risk $\mathcal{R}(\widehat{f})$ is small on average or with high probability with respect to $\mathcal{D}_n$.

Machine Learning

- Learn a rule to construct a classifier $\widehat{f} \in \mathcal{F}$ from the training data $\mathcal{D}_n$ s.t. the risk $\mathcal{R}(\widehat{f})$ is small on average or with high probability with respect to $\mathcal{D}_n$.

Canonical example: Empirical Risk Minimizer

- One restricts f to a subset of functions $\mathcal{S} = \{f_\theta, \theta \in \Theta\}$
- One replaces the minimization of the average loss by the minimization of the empirical loss

$$\widehat{f} = f_{\widehat{\theta}} = \operatorname*{argmin}_{f_\theta, \theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, f_\theta(\mathbf{X}_i))$$

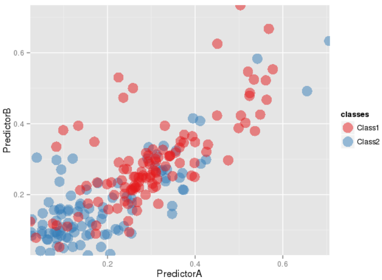
- Examples:
  - Linear discrimination with

$$\mathcal{S} = \{\mathbf{x} \mapsto \mathtt{sign}\{\beta^T \mathbf{x} + \beta_0\} \,/ \beta \in \mathbb{R}^d, \beta_0 \in \mathbb{R}\}$$

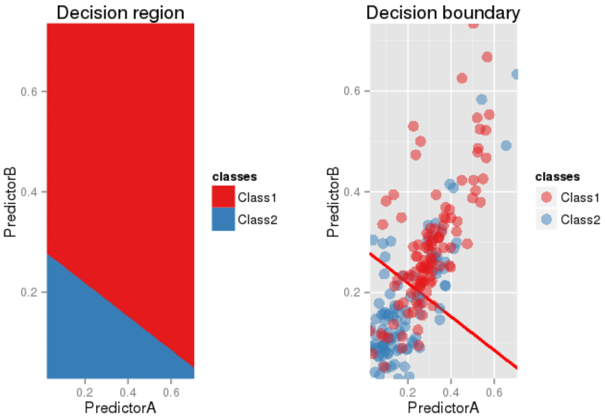
Synthetic Dataset

- Two features/covariates.
- Two classes.

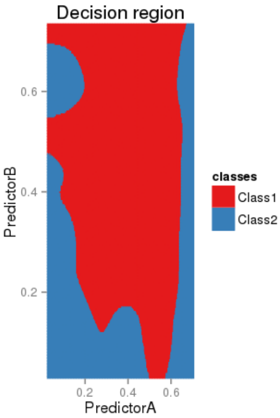
- Dataset from *Applied Predictive Modeling*, M. Kuhn and K. Johnson, Springer
- Numerical experiments with **R** and the **caret** package.

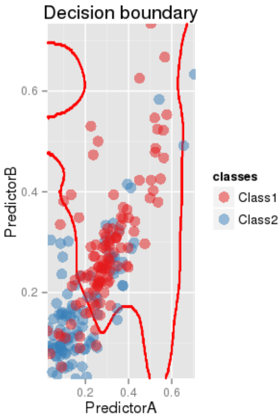


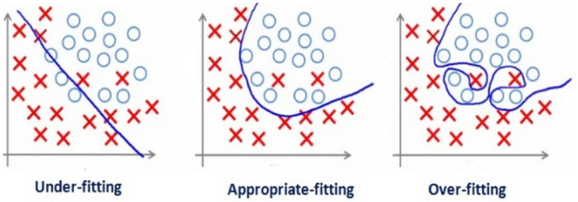
Example: Linear Discrimination



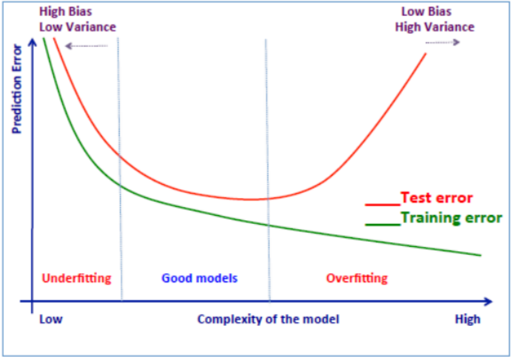
Example: More Complex Model







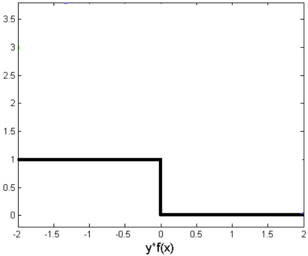
- Different behavior for different model complexity
- Under-fit : Low complexity models are easily learned but too simple to explain the truth.
- Over-fit : High complexity models are memorizing the data they have seen and are unable to generalize to unseen examples.



- We can determine whether a predictive model is underfitting or overfitting the training data by looking at the prediction error on the training data and the test data.
- How to estimate the test error ?

Empirical Risk Minimizer

$$\widehat{f} = \underset{f \in \mathcal{S}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell^{0/1}(Y_i, f(\mathbf{X}_i))$$



- Classification loss: $\ell^{0/1}(y, f(x)) = \mathbf{1}_{y \neq f(x)}$
- Not convex and not smooth!

- The best solution f^* (which is independent of $\mathcal{D}_n$) is

$$f^* = \arg\min_{f \in \mathcal{F}} R(f) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\left[\ell(Y, f(\mathbf{X}))\right]$$

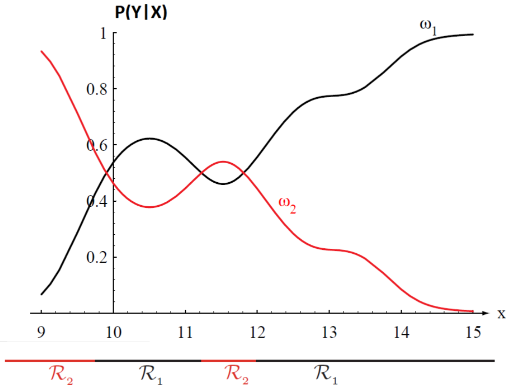
Bayes Predictor (Ideal solution)

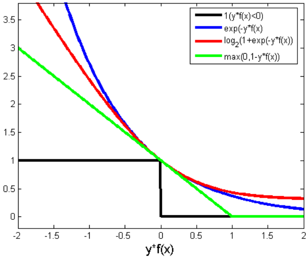
In binary classification with $0-1$ loss:

$$f^*(\mathbf{X}) = \begin{cases} +1 & \text{if } \ \mathbb{P}\left\{Y = +1|\mathbf{X}\right\} \geq \mathbb{P}\left\{Y = -1|\mathbf{X}\right\} \\ -1 & \text{otherwise} \end{cases}$$

Issue: Explicit solution requires to know $\mathbb{E}\left[Y|\mathbf{X}\right]$ for all values of $\mathbf{X}$!

Conditional prob. and Bayes Predictor



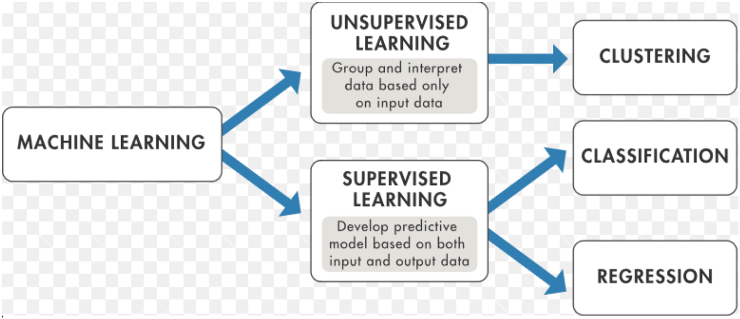


- Classification loss: $\ell^{0/1}(y, f(x)) = \mathbf{1}_{y \neq f(x)}$
- Not convex and not smooth!

Classical convexification

- Logistic loss: $\ell(y, f(x)) = \log(1 + e^{-yf(x)})$ (Logistic / NN)
- Hinge loss: $\ell(y, f(x)) = (1 - yf(x))_+$ (SVM)
- Exponential loss: $\ell(y, f(x)) = e^{-yf(x)}$ (Boosting...)

Machine Learning



Methods (Today):

1. k Nearest-Neighbors

T. Hastie, R. Tibshirani, and J. Friedman (2009)
The Elements of Statistical Learning
Springer Series in Statistics.

G. James, D. Witten, T. Hastie and R. Tibshirani (2013)
An Introduction to Statistical Learning with Applications in R
Springer Series in Statistics.

B. Schölkopf, A. Smola (2002)
Learning with kernels.
The MIT Press