Ingénierie des systèmes décisionnels (NFE212)
Data Mining and Learning

Nicolas Thome
Conservatoire Nationnal des Arts et Métiers (Cnam)
Laboratoire CEDRIC

le cnam

Cédric

Schedule

- Courses:
  - Machine Learning for data mining
  - Deep learning
- Practical session:
  - Data Visualization
  - Deep learning and manifold untangling

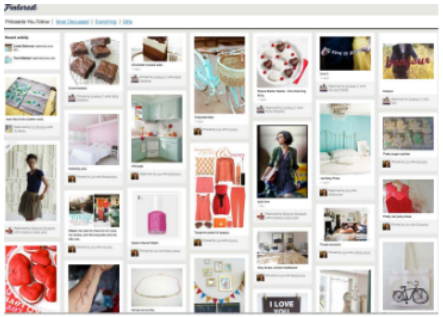
Outline

Context

Big Data

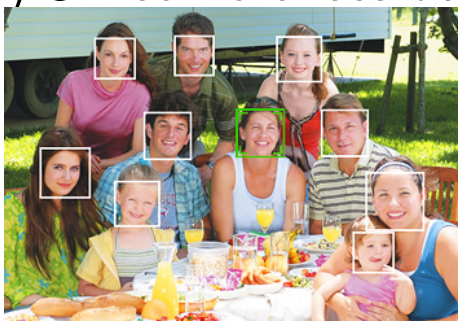
- Superabundance of data: images, videos, audio, text, use traces, *etc*

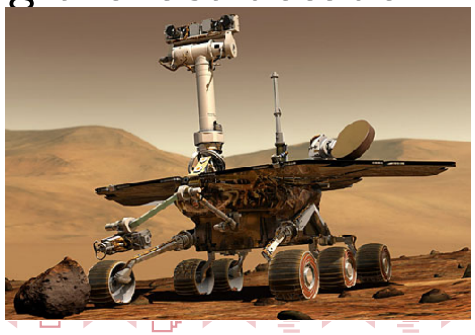
BBC: 2.4M videos


Facebook: 350B images
1B each day


100M monitoring cameras

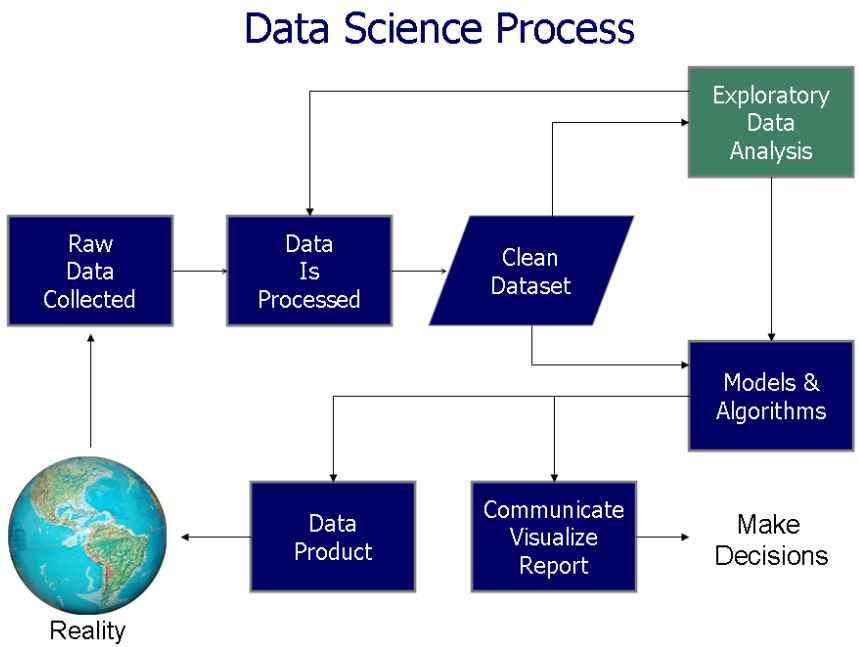
- Obvious need to access, search, or classify these data: **Recognition**
- Huge number of applications: mobile visual search, robotics, autonomous driving, augmented reality, medical imaging *etc*
- Leading track in major ML/CV conferences during the last decade





Data science

- Data collect, storage and access
- Data Cleaning, pre-processing
- Data Visualization
- Data Mining



Data Science

Terminology

- Data are heterogeneous
- Quantitative variables: continuous $\in \mathbb{R}$, discrete in $\in \mathbb{Z}$
  - Natural to compute distances or statics, *e.g.* mean, variance, higher-order statics, distributions, *etc*
- Qualitative variables:
  - Categorical: blood type of a person (A, B, AB or O), state for US resident, type of a music (rock, pop, folk), sex, religion *etc*
  - Ordinal: ordered categories but no metric, *e.g.* education level (license, master), rate for a product (bad, good, excellent)
- Database: collection of n objects or individuals.
- Each object: p variables
- $\Rightarrow$ represented by a matrix of size $n \times p$
  - n rows : objects, individuals, entities, *etc*
  - p columns : variables, attributes, features, *etc*

Data science

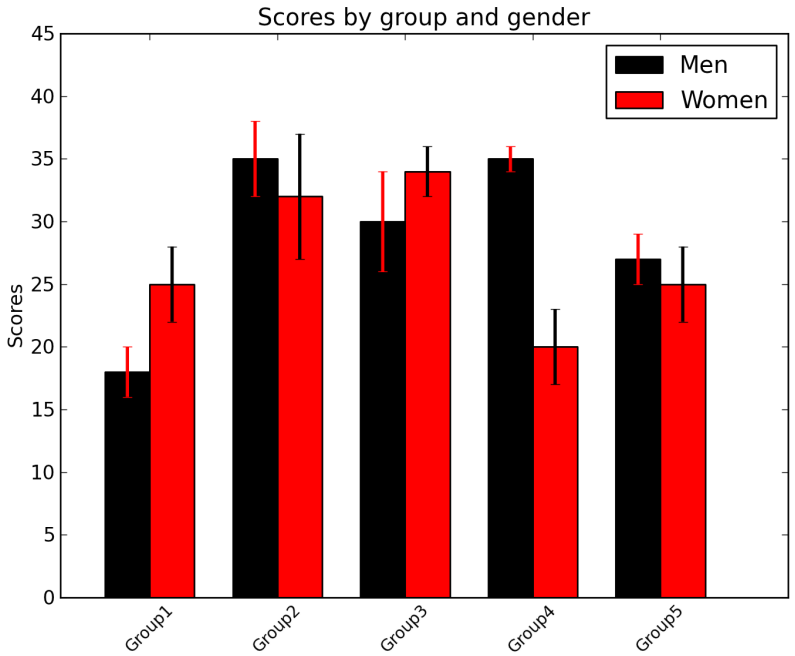
Data Cleaning

- Detecting corrupt / inaccurate parts of the data
  - Corruption ex: entry errors (format, missing or duplicate data), corruption in transmission or storage, *etc*
  - Correcting / removing them : ex address with an invalid zip code
- Quality requirements $\Rightarrow$ need to pass a set of quality criteria
  - Lore high level requirements
  - Might use machine learning tools

Data science

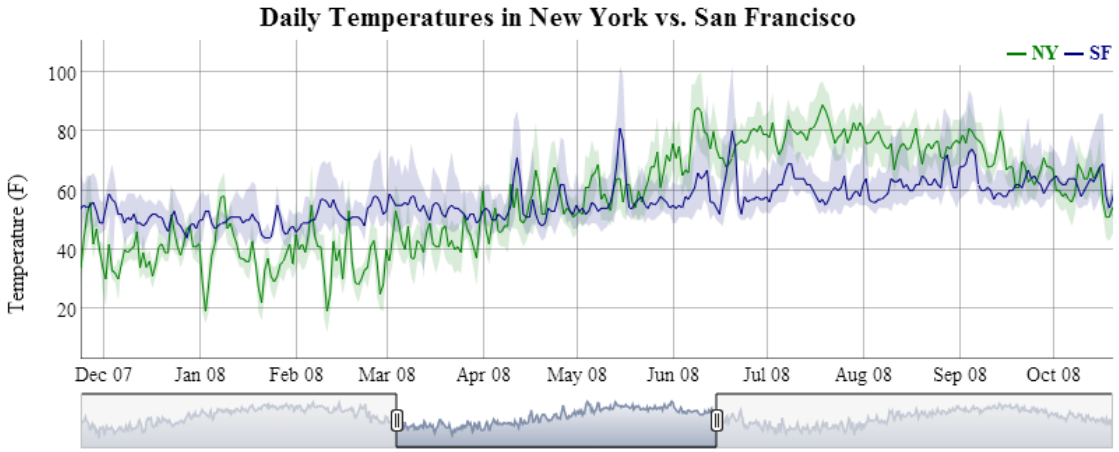
Data Visualization

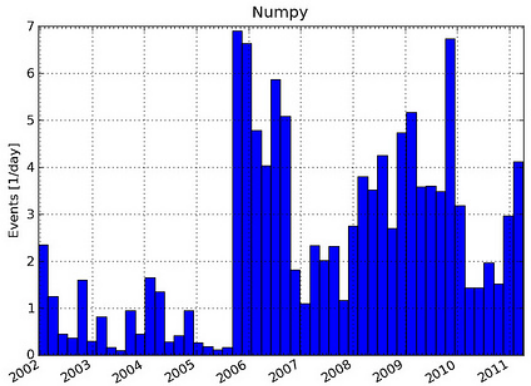
- Important to visualize a database
- Visualization of mono-dimensionnal data:
  - Use scalar statistics: mean, median, variance (std), quantile
  - Ex: visualize mean and std :



Data Visualization

- Visualization of mono-dimensional data:
  - Raw curve, *e.g.* temporal series
  - Represent statistical density (pdf) of the data: histogram, density estimation

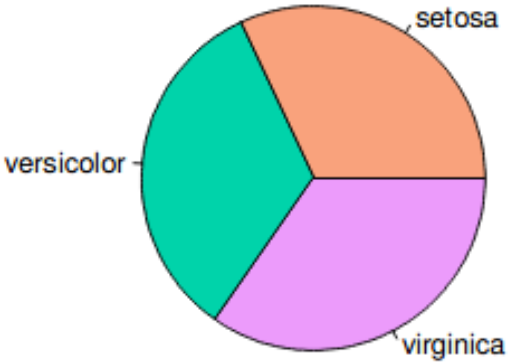


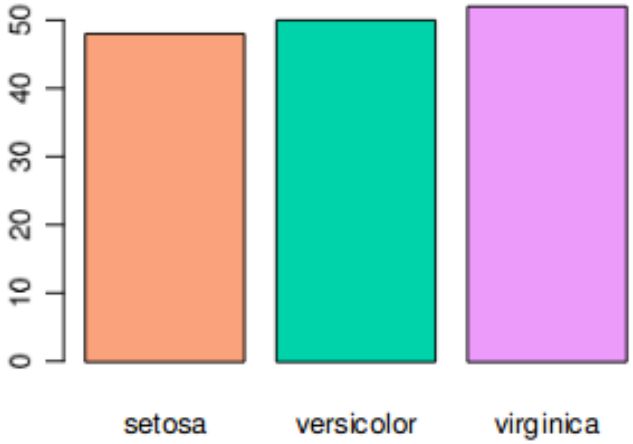


Visualization of mono-dimensional data

Some better than the others :

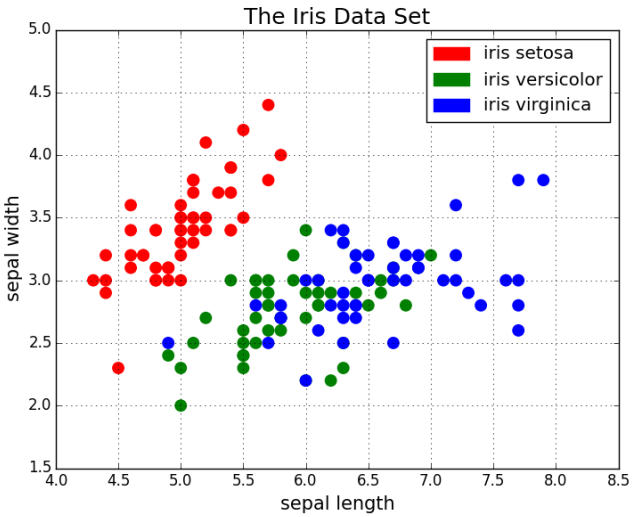
Difficult to estimate area ratios with pie charts



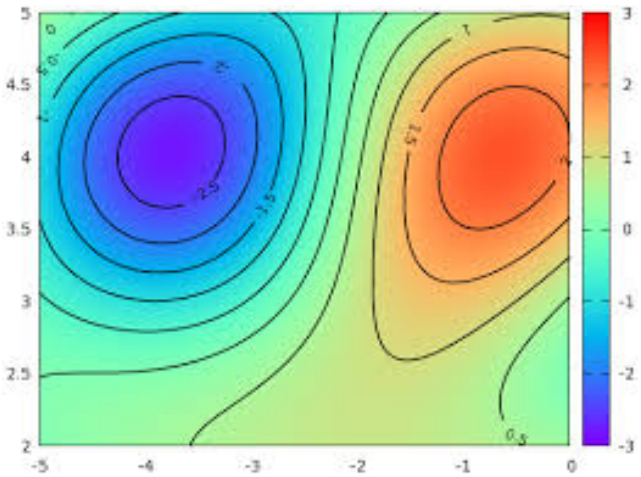


Data Visualization: Visualization of 2-dimensional data

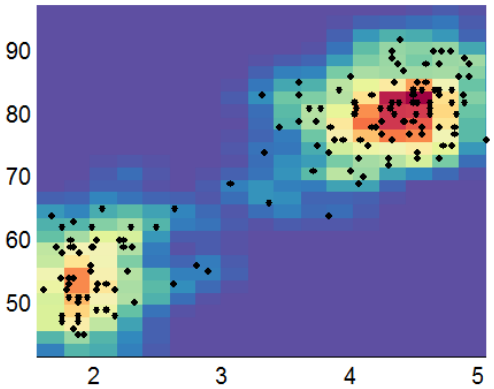
Scatter plot

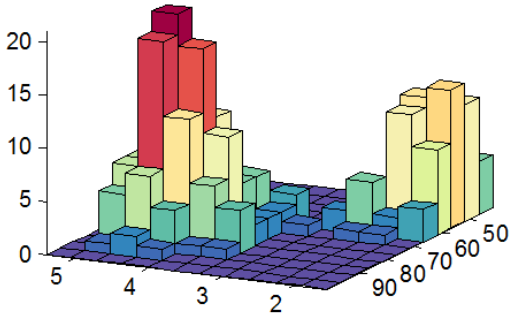


Contour plot



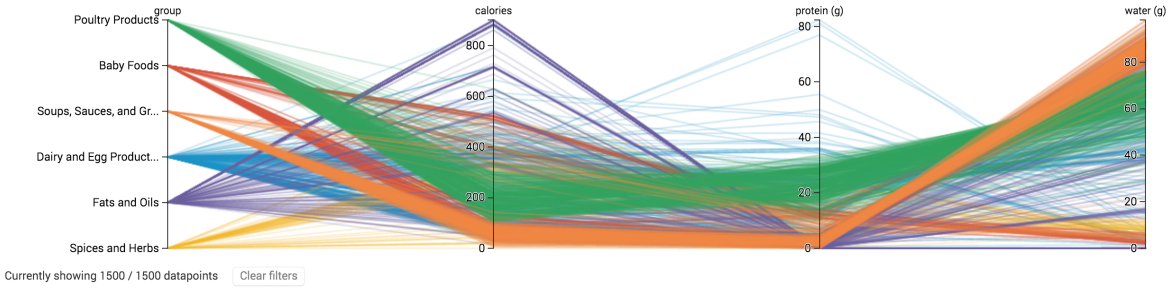
2D histogram



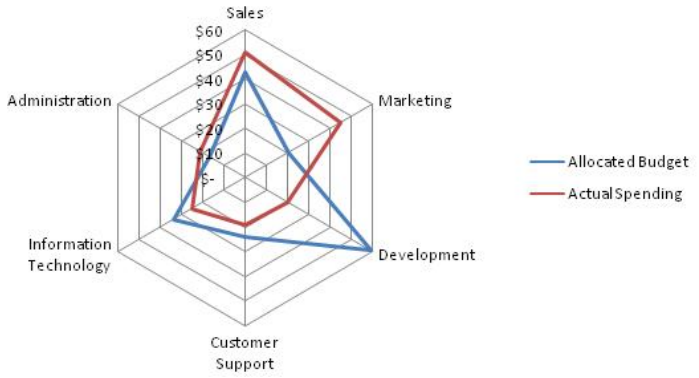


Visualization of multi-dimensionnal data

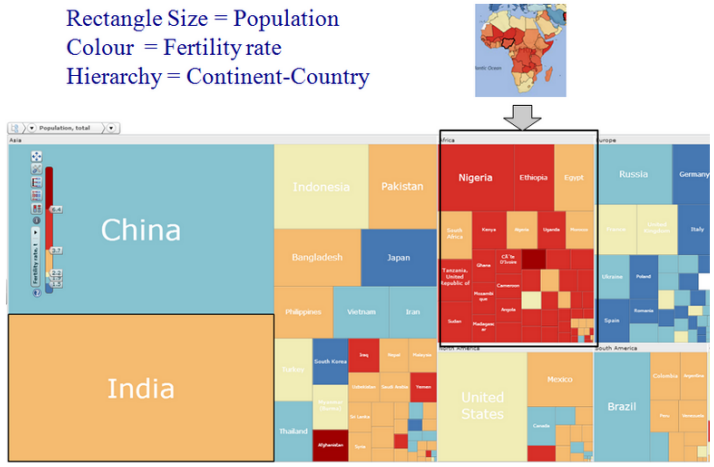
Parallel coordinates



Star Graph



Tree Map



Data Visualization

Visualization of multi-dimensionnal data

- And many others: adjacency diagrams, graph layout, *etc*
- Main limitation: only visualize a few variables at a time
- How to capture semantic information from multi-dimensional data for visualization ? $\Rightarrow$ data mining

Data science

Data Mining

- General goal of Data Mining: present certain aspects of the data in an usable and understandable manner
  - Finding relations, dependence between individuals and variables
  - Perform variable selection
  - Anomaly detection: diagnostic, fraud, *etc*
- Input x, expected output y, dependent of the task:
  - Density estimation, clustering
  - Classification
  - Regression

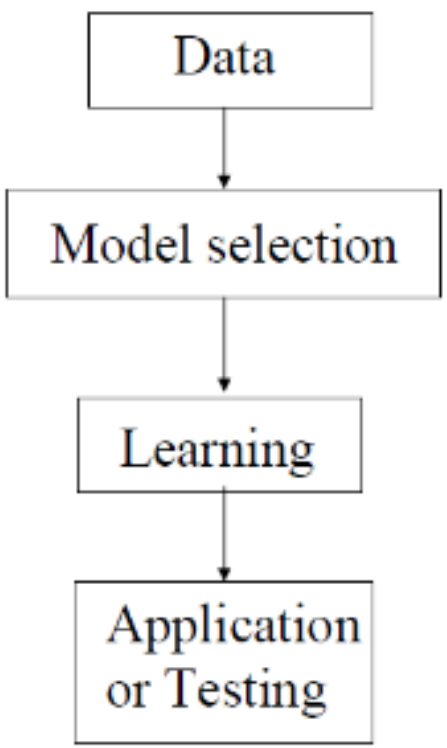
This course: Data Mining & Machine Learning

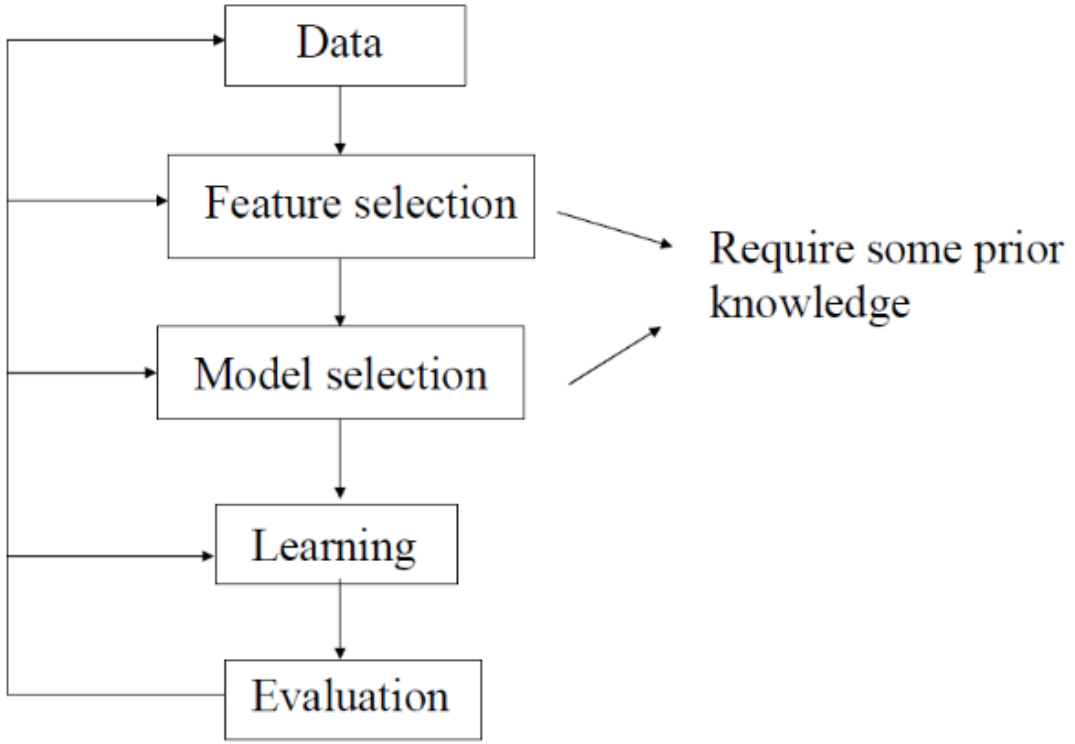
Models

- **This course: machine learning models**
  - Data oriented, less assumption on data, expertise and prior knowledge ($\neq$ physical models, *e.g.* weather forecas)
  - Weather forecast ex: capturing statical regularities between x to y, not the causal process from x to y !
- **Outline:**
  - Week 1: Machine learning intro, probabilistic models
  - Week 2: Manifold learning, visualisation
  - Week 3: Introduction to deep learning
  - Week 4: practical session, visualization & deep learning

Outline

Learning Contexts for Data Mining

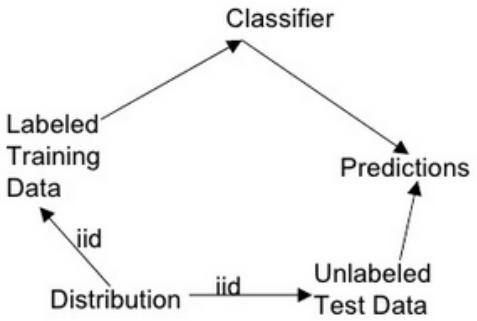




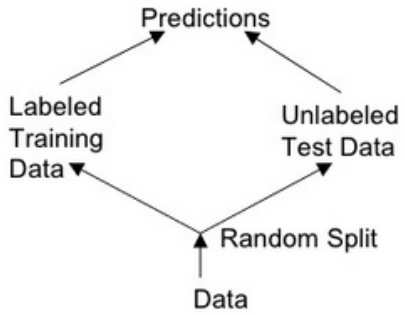
Learning Contexts for Data Mining

Transductive and Inductive Learning [CSZ06]

- Transductive learning: reasoning from observed, specific (training) cases to specific (test) cases
  - A closed database, no need to predict outside of it
- Inductive Learning: reasoning from observed training cases to general rules
  - Generalization capacity outside training dataset crucial



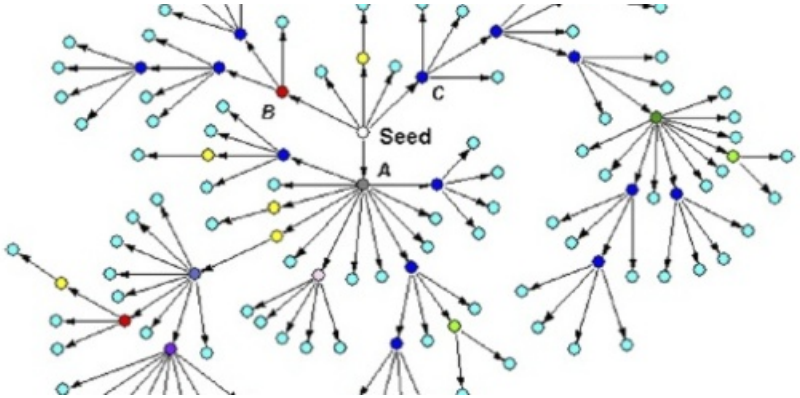
Inductive Learning



Transductive Learning

Transductive and Inductive Learning: Applications

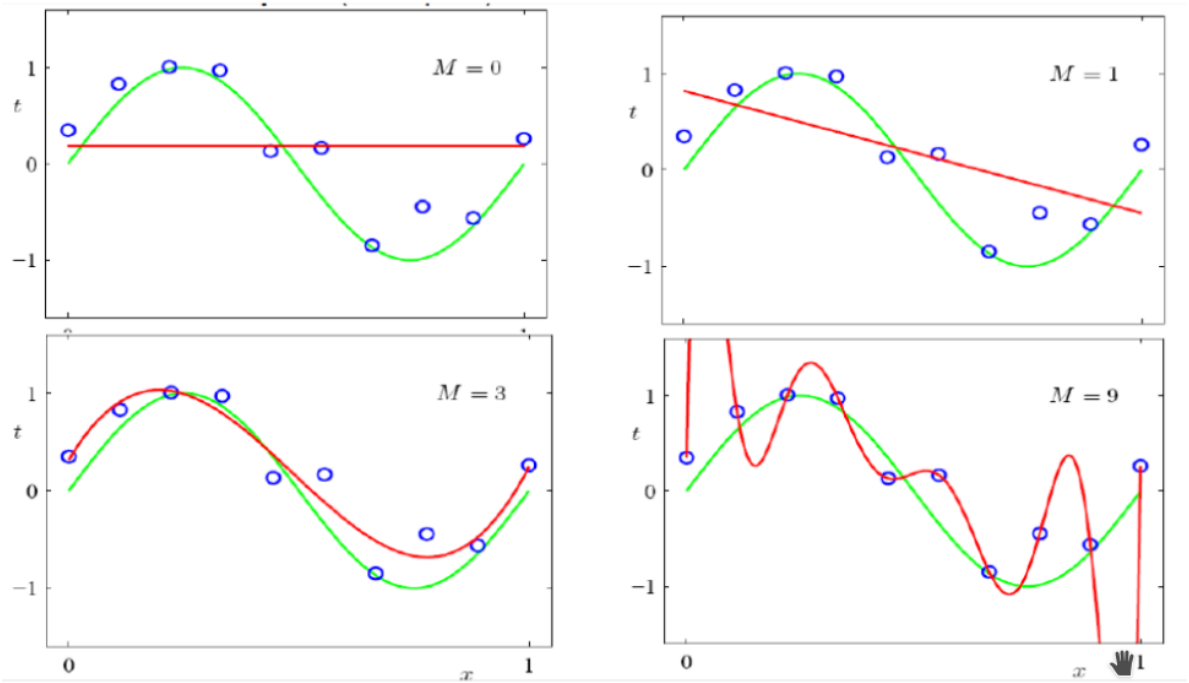
- Transductive Learning:
  - Interactive Key-word search in the web
  - Information diffusion in social networks
  - Data visualization in a closed database
- Inductive learning:
  - Semantic data prediction: text / image / sound / speech recognition
  - Spam filtering, anomaly detection
  - Recommendation, *etc*



Learning Contexts for Data Mining

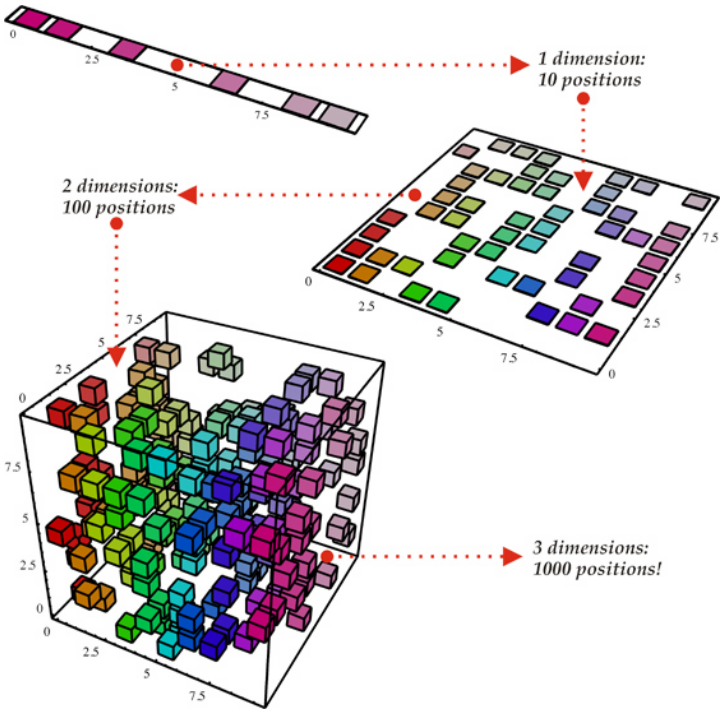
Generalization and Model selection [HTF01]

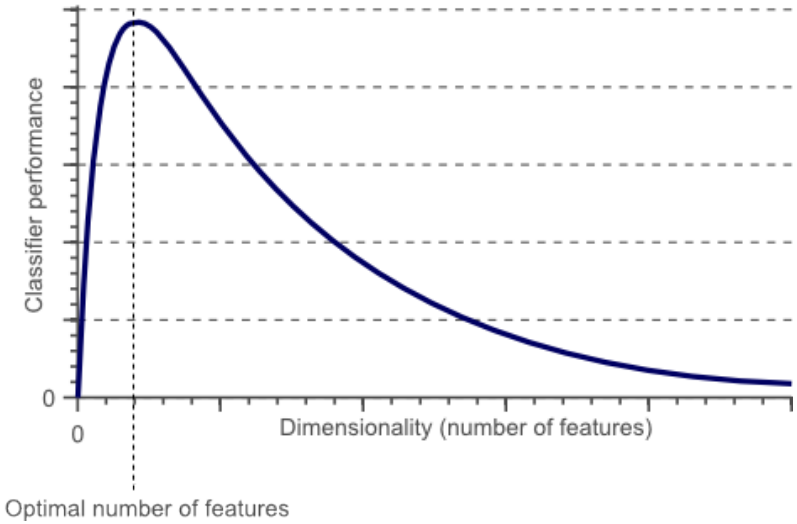
- Inductive Learning: must generalize to unseen examples
- **Generalization:** specific constraints on the choice of the model
  - Under-fitting *vs* over-fitting



Generalization and Model selection

- Related to the **curse of dimensionality**
  - The volume of the space exponentially grows as dimension (# features) increases
  - Needs exponentially more examples to generalize well

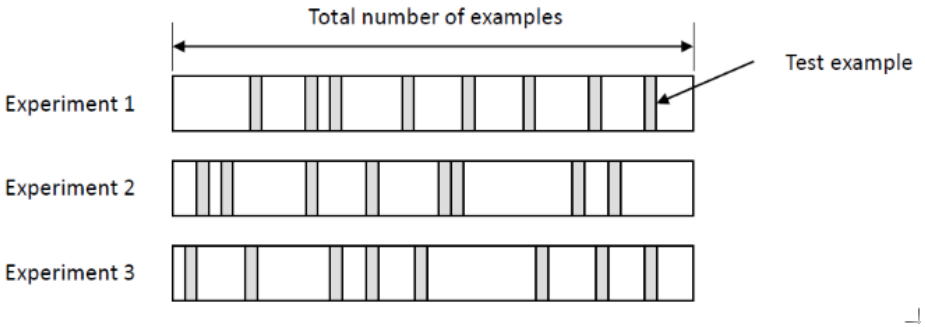




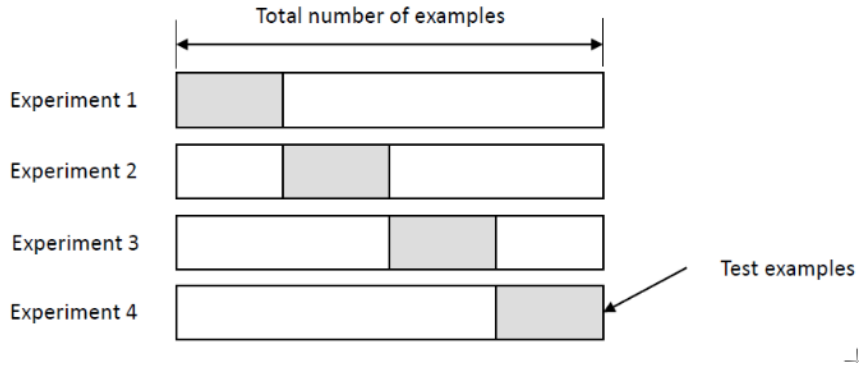
Generalization and Model selection

How to perform model selection and measure over-fitting ?

- Use a separate train / validation and test set
  - Model parameters learned on the train set
  - Perf eval and hyper-parameters tuned on the validation set
  - Generalization performance evaluated on the test set
- To asses the impact of the train / val / set splitting, use several folds
  - Random sampling, cross-validation, leave-one-out



Random sampling

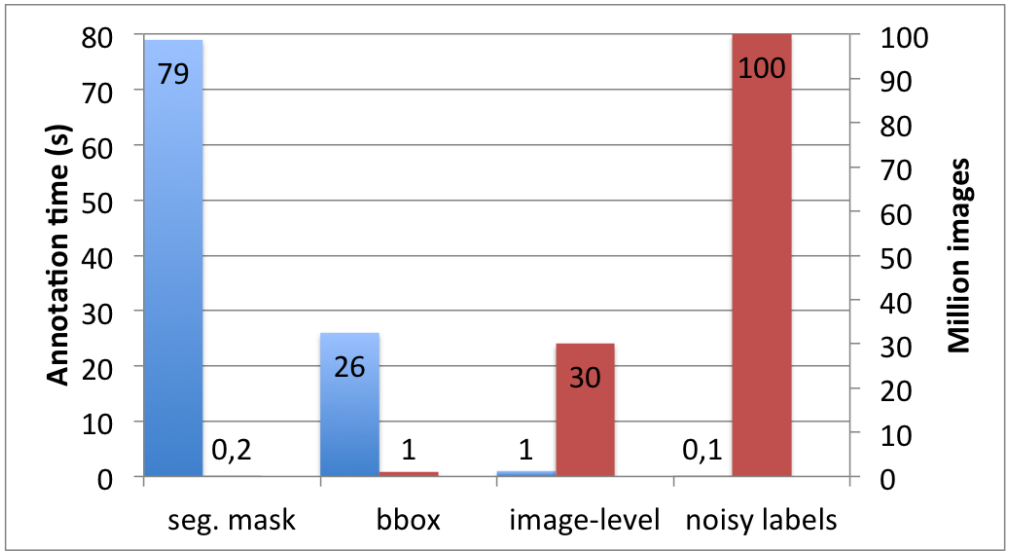


Cross-validation

Learning Contexts for Data Mining

Supervision and Learning

- Big data: tons of examples, but (accurate) annoations expensive
- Unsupervised learning: no annotation $\Rightarrow$ Discover data structure
- Supervised learning: full annotations
- Semi-supervised learning: full and no annotations
- Weakly supervised learning: coarse annotations for accurate prediction

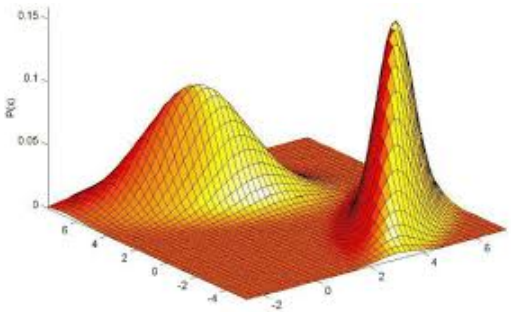


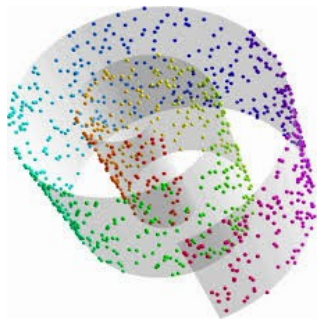
Outline

Unsupervised Learning

Unsupervised Learning

- Learning the structure of the data, without supervision
  - Structure or distribution
- Learning internal representation of the data
- Internal representations low-dimensional (2d / 3d)
  $\Rightarrow$ dimension reduction
  - Can be used for data visualization
- Method classification (non exhaustive) :
  1. Probabilistic Generative Models
  2. Manifold Learning





Probabilistic Generative Models [Mur12, Bis06]

Problem Formulation

- x input, θ model parameters
- Bayesian setting : $p(\mathrm{x};\theta) = p(\mathrm{x}/\theta)p(\theta) = p(\theta/\mathrm{x})p(\mathrm{x})$

$$\Rightarrow p(\theta/\mathrm{x}) = \frac{p(\mathrm{x}/\theta)p(\theta)}{p(\mathrm{x})} \propto p(\mathrm{x}/\theta)p(\theta)$$

- What we model:
  - $p(\mathrm{x}/\theta)$: likelihood
  - $p(\theta)$: prior
  - $p(\theta/\mathrm{x}) \propto p(\mathrm{x}/\theta)p(\theta)$ posterior
- Maximum Likelihood (ML): ignore (or assume uniform) prior
  $\Rightarrow$ find $\hat{\theta}$ s.t $p(\mathrm{x}/\theta)$ max
- Maximum A Posteriori (MAP): use prior $p(\theta)$
  $\Rightarrow$ find $\hat{\theta}$ s.t $p(\theta/\mathrm{x})$ max

Probabilistic Generative Models

Learning Formulation

- A set of training N examples $\mathbb{X} = \{x_1, x_2, ...x_N\}$
- Assumed to be i.i.d (independent and identically distributed)

$$\Rightarrow p(x/\theta) = \prod_{i=1}^{N} p(x_i/\theta)$$

- $p(\theta/x) \propto p(\theta) \prod_{i=1}^{N} p(x_i/\theta)$
- Maximum Likelihood (ML): $\hat{\theta} = \arg\max_{\theta} p(x/\theta) = \arg\min_{\theta} \mathcal{L}_{ML}(\theta)$

$$\mathcal{L}_{ML}(\theta) = -\sum_{i=1}^{N} log\,[p(x_i/\theta)]$$

- Maximum A Posteriori (MAP): $\hat{\theta} = \arg\max_{\theta} p(\theta/x) = \arg\min_{\theta} \mathcal{L}_{MAP}(\theta)$

$$\mathcal{L}_{MAP}(\theta) = -log\,[p(\theta)] - \sum_{i=1}^{N} log\,[p(x_i/\theta)]$$

Probabilistic Generative Models

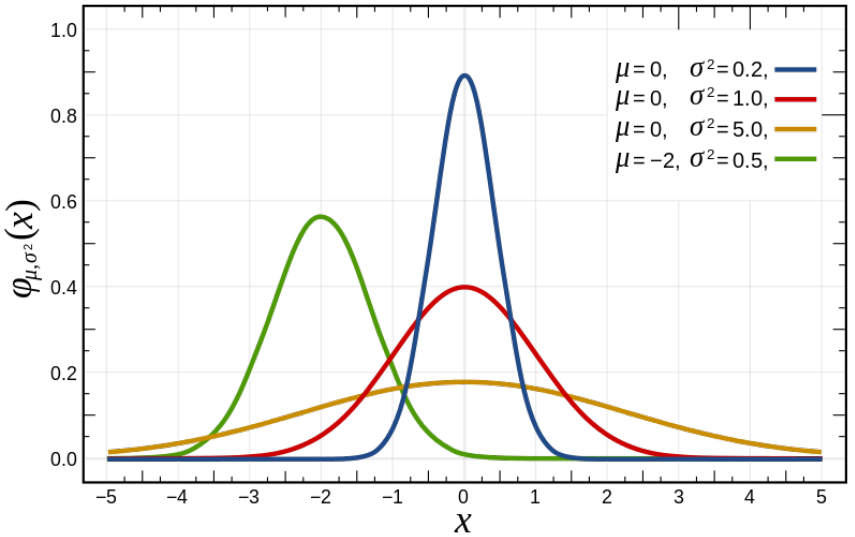
Gaussian Model

- Multivariate Gaussian distribution, $x_i \in \mathbb{R}^d$: likelihood

$$p(x_i/\theta) = \mathcal{N}(x_i|, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-(x_i-\mu)^t \Sigma^{-1}(x_i-\mu)}$$

- μ mean vector $\in \mathbb{R}^d$, $\Sigma \in S_d^+(\mathbb{R})$ PSD covariance matrix
- In the in 1d case:

$$p(x_i/\theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$



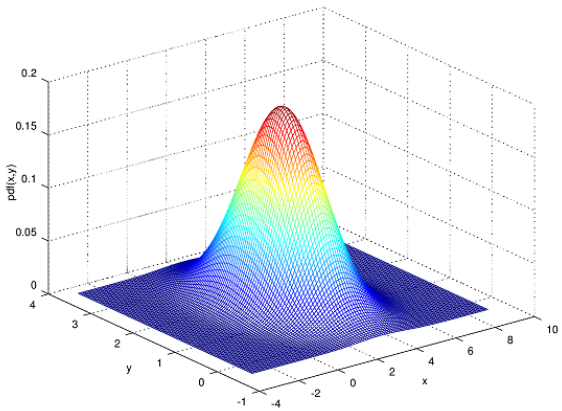
Probabilistic Generative Models

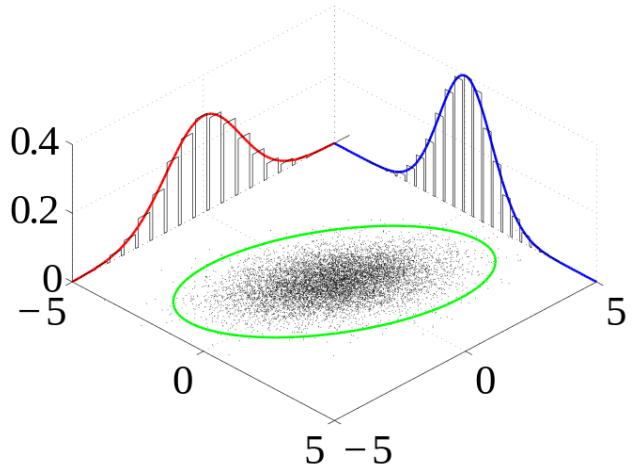
Gaussian Model

- Multivariate Gaussian distribution, $x_i \in \mathbb{R}^d$: likelihood

$$p(x_i/\theta) = \mathcal{N}(x_i|, \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-(x_i-\mu)^t \Sigma^{-1}(x_i-\mu)}$$

- μ mean vector $\in \mathbb{R}^d$, $\Sigma \in S_d^+(\mathbb{R})$ PSD covariance matrix
- Example in 2D: product of 1d gaussians





Probabilistic Generative Models

Gaussian Model and Maximum Likelihood

- Maximum Likelihood (ML) estimation for Gaussians: closed-form solution for minimizing $\mathcal{L}_{ML}(\theta)$

$$\mathcal{L}_{ML}(\theta) = \frac{N}{2} log(|\Sigma|) + \frac{1}{2}\sum_{i=1}^{N}(x_i - \mu)^t \Sigma^{-1}(x_i - \mu)$$

- $\frac{\partial \mathcal{L}_{ML}}{\partial \mu} = 0 \Rightarrow \hat{\mu}_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i$
- $\frac{\partial \mathcal{L}_{ML}}{\partial \Sigma} = 0 \Rightarrow \hat{\Sigma}_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})(x_i - \hat{\mu})^t$
- Intuitive interpretation : μ data mean, Σ data covariance matrix

Limits of ML Estimation

- Learning model to predict coin toss with MLE
  - Bernoulli variable X with param p:
    $$P(\mathrm{x}|p) = \prod_{i=1}^{N} P(x_i|p) = \prod_{i=1}^{N} p^{x_i}(1-p)^{1-x_i}$$
  - MLE: $\ln P(\mathrm{x}|p) = \sum_{i=1}^{N} [x_i \ln p + (1-x_i)\ln(1-p)] \Rightarrow \boxed{p_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i}$
  - MLE: predict $P(X|p_{MLE}) = 1$ for all futures tosses!

- Using prior knowledge on p, *e.g.* $P(p) = 0.5$ or $P(p) = 0.3 \Rightarrow$ MAP

Probabilistic Generative Models

Gaussian Model and Maximum A Posteriori (MAP)

- Maximum A Posteriori (MAP): assume we have a gaussian prior on parameters
- Example: 1d Gaussian example, we have Gaussian prior on μ $p(\mu) \sim \mathcal{N}(\mu_0, \sigma_0)$
- What is the form of the posterior $p(\mu/\mathrm{x}) \propto p(\mathrm{x}/\mu)p(\mu)$?
- $p(\mu/\mathrm{x}) \propto e^{\sum_{i=1}^{N} \frac{(x_i-\mu)^2}{2\sigma} + \frac{(\mu-\mu_0)^2}{2\sigma_0}}$ ....

Probabilistic Generative Models

Gaussian Model and Maximum A Posteriori (MAP)

- We can show that $p(\mu/\mathrm{x})$ takes the form:

$$p(\mu/\mathrm{x}) = \frac{1}{\sqrt{2\pi}\sigma_n} e^{-\frac{(\mu-\mu_n)^2}{2\sigma_n^2}}$$

- This is a normal distribution ! Mean μ_n, std σ_n
- So what?
  $\Rightarrow$ there is a closed formed for the Maximum A Posteriori (MAP) computation in the Gaussian case !

Probabilistic Generative Models

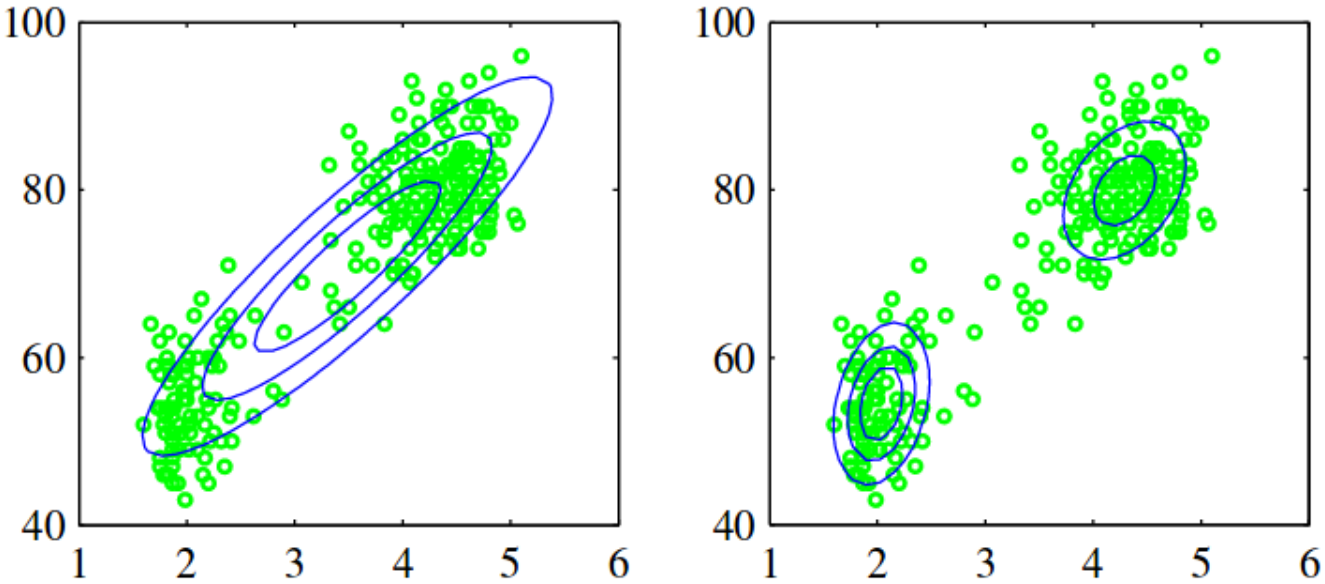
Conjugate Prior

- Prior function conjugate to likelihood $\Rightarrow$ posterior as the same form as the prior
- In the Gaussian case: Gaussian prior, Gaussian likelihood $\Rightarrow$ Gaussian posterior
  - Gaussian family self-conjugate
- Consequence: if we have closed form solution for the prior $\Rightarrow$ closed form solution for the posterior
- Other example : Beta distribution conjugate prior to the binomial:
  - Resulting posterior also a Beta distribution.
- All members of the exponential family have conjugate priors

Probabilistic Generative Models

Gaussian Mixture Model

- Limitation of single a Gaussian: only able to model unimodal distributions
  $\Rightarrow$ Need to use different Gaussians for different modes
- Using a mixture of Gaussian $\Rightarrow$ general distribution modeling



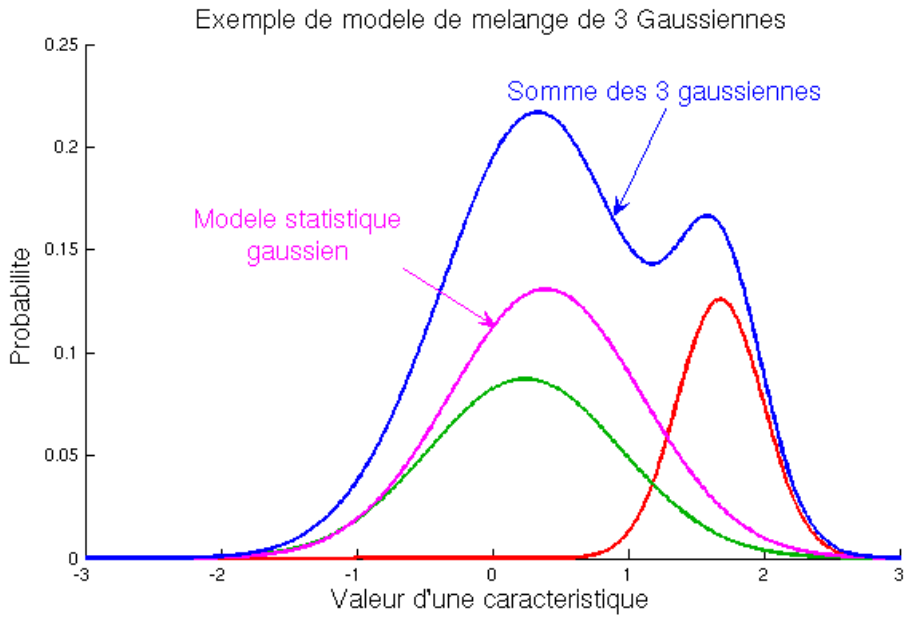
Probabilistic Generative Models

Gaussian Mixture Model (GMM)

- GMM: weighted sum of K Gaussian

$$p(x) = \sum_{m=1}^{k} \pi_m \, \mathcal{N}(x|\mu_m, \Sigma_m)$$

- Using a sufficient large $K \Rightarrow$ general distribution modeling



Gaussian Mixture Model (GMM)

Maximum likelihood (ML) for GMM

- Log likelihood :

$$\mathcal{L}_{ML}(X;\Theta) = -\sum_{i=1}^{n} log \sum_{m=1}^{k} \pi_m \, \mathcal{N}(x_i|\mu_m, \Sigma_m) \quad (1)$$

- Eq. (1): no closed-form solution $\neq$ Gaussian (sum inside log)
- Expectation-Maximization (EM) algorithm [DLR77]
- From Jensen inequality:

$$\mathcal{L}_{ML}(X;\Theta) = \sum_{i=1}^{n} log \sum_{m=1}^{k} q(m,i) \geq \sum_{i=1}^{n} \sum_{m=1}^{k} p(m|i) log(q(m,i)) - \sum_{i=1}^{n} \sum_{m=1}^{k} p(m|i) log(p(m|i))$$

Gaussian Mixture Model (GMM)

EM pour GMM

$$\mathcal{L}_{ML}(X;\Theta) = \sum_{i=1}^{n} log \sum_{m=1}^{k} q(m,i) \geq \sum_{i=1}^{n}\sum_{m=1}^{k} p(m|i) log(q(m,i)) - \sum_{i=1}^{n}\sum_{m=1}^{k} p(m|i) log(p(m|i))$$

- Let us define : $q(m,i) = \pi_m \, \mathcal{N}(x_i|\mu_m, \Sigma_m)$ et $p(m|i) = \frac{q(m,i)}{\sum_{l=1}^{k} q(l,i)}$
- If $p^t(m|i)$ fixed, maximize 1^{st} term: $\sum_{i=1}^{n}\sum_{m=1}^{k} p^t(m|i) log(q(m,i))$
- Iterative EM algorithm:
  - Compute $p^t(m|i)$ with parameters (π_m, μ_m, Σ_m) fixed : Expectation-step
  - Update parameters (π_m, μ_m, Σ_m) with $p^t(m|i)$ fixed : Maximization-step. $\sim$ single Gaussian optimization $\Leftrightarrow$ closed form solution.

Gaussian Mixture Model (GMM)

GMM and K-Means

- K-Means [Mac67]: certainly the most popular clustering method
- EM is a generalization GMM of K-Means
- K-Means is equivalent to GMM in the case of:
  - Uniform Gaussian weights $\pi_m = \frac{1}{k}$
  - Diagonal covariance Matrices with identical variance terms: $\Sigma_m = \sigma I_p$
  - $\sigma \to 0$

K-Means (recap)

- Let N vectors $x_i \in \mathbb{R}^p$: $\{x_i\}$, $i \in \{1; N\}$
- A set of K clusters, $\{c_m\}$, $c_m \in \mathbb{R}^p$ et $m \in \{1; k\}$

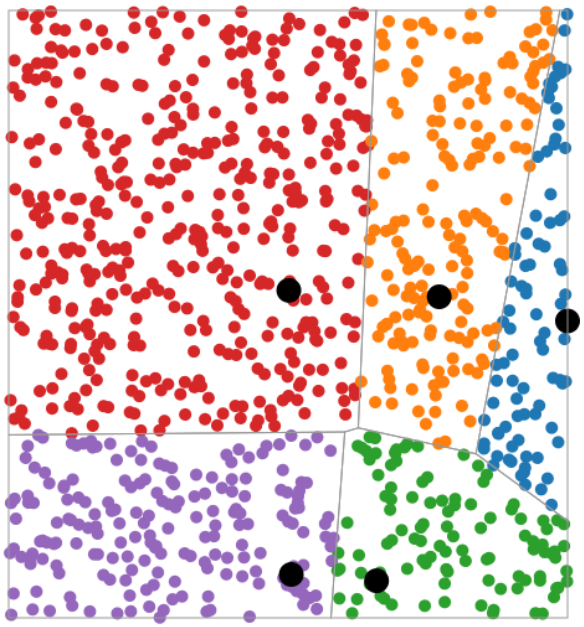
K-Means Formulation: find the k clusters $\widehat{c_m}$ minimizing data quantization error$f(c_m)$:

$$\{\widehat{c_m}\} = \underset{\{c_m\} \in \mathbb{R}^d}{\arg\min}\, f(c_m) = \underset{\{c_m\} \in \mathbb{R}^d}{\arg\min} \sum_{m=1}^{M} \sum_{x_i \in c_m} ||x_i - c_m||_2^2 \qquad (2)$$

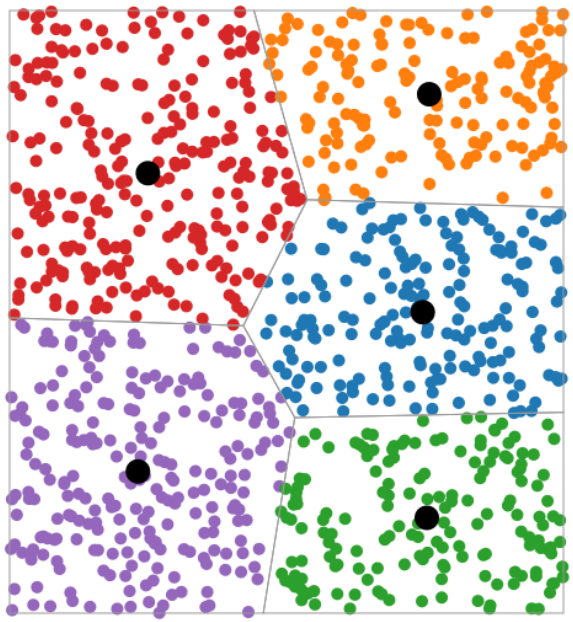
- Assigning points x_i to the closest center leads to non-linearity ($x_i \in c_m$)

K-Means

- Solving Eq. (2) (even with fixed M): prohibitive complexity, $\mathcal{O}(N^{Md+1}log(N))$
- K-Means: heuristic to solve the problem
  - Random Initialization of centers in data points
  - Center update
  - (re)-assigning points to closest center



Initialization



Convergence

K-Means

Algorithm

Algorithm 1 K-Means Algorithm

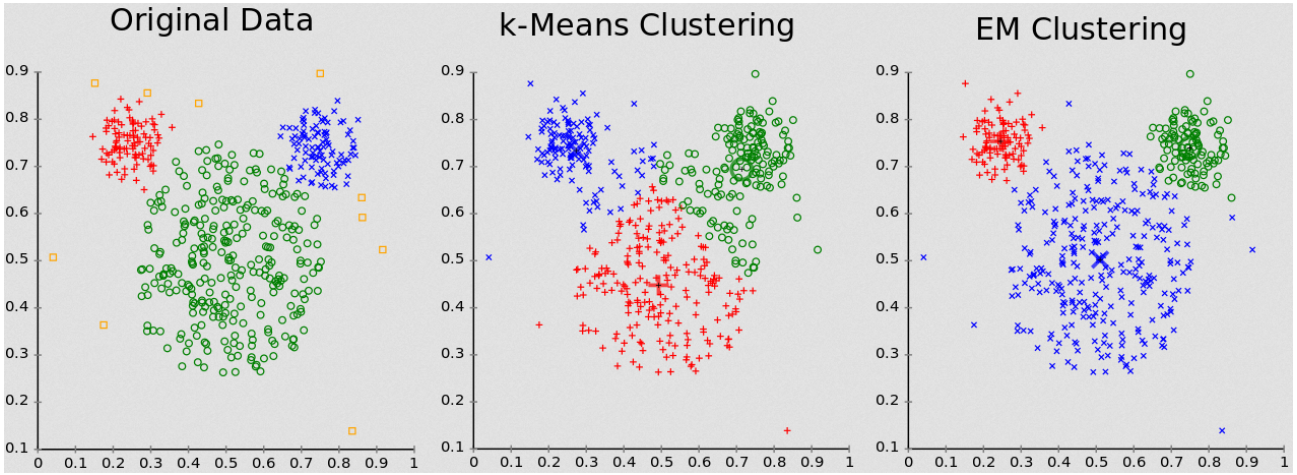
Input: $\{x_i\} \in \mathbb{R}^p$, $i \in \{1; n\}$ (n input points) and k (requested number of centers).
Output: $\{\widehat{c_m}\} \in \mathbb{R}^p$: k optimal centers.

1: $\{c_m^0\} \leftarrow \{x_i\}$, $(m \times i) \in \{1; k\} \times \{1; n\}$
// Random Initialization of the k centers in the n points.
2: **repeat**
3: Assignment: $c_m^t = \{x_i \ : \ ||x_i - c_m||_2 < |x_i - c_{m'}||_2, m' \neq m\}$
// Finding x_i points closer to c_m center than other $c_{m'}$ centers.
4: Update: $c_m^t = \frac{1}{|c_m|} \sum_{x_i \in c_m} x_i$
// Cluster center c_m is re-estimated as the barycentre of points $x_i \in c_m$.
5: **until** $\sum_{m=1}^{M} ||c_m^t - c_m^{t-1}||_2 > 0$ // Centers move
6: **return** $\{c_m\} = \{\widehat{c_m}\}$

Gaussian Mixture Model (GMM)

GMM and K-Means

- EM is a generalization GMM of K-Means
  - E GMM step: $\leftrightarrow$ K-Means assignment
  - M GMM step: update Σ_m and π_m in addition to μ_m for K-Means
- K-Means is equivalent to GMM in the case of:
  - Uniform Gaussian weights $\pi_m = \frac{1}{k}$
  - Diagonal covariance Matrices with identical variance terms: $\Sigma_m = \sigma I_p$
  - $\sigma \to 0$



References I


Christopher M. Bishop, *Pattern recognition and machine learning (information science and statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.


Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien (eds.), *Semi-supervised learning*, The MIT Press, 2006.


A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society: Series B **39** (1977), 1–38.


Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The elements of statistical learning*, Springer Series in Statistics, Springer New York Inc., New York, NY, USA, 2001.


J. B. MacQueen, *Some methods for classification and analysis of multivariate observations*, Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability (L. M. Le Cam and J. Neyman, eds.), vol. 1, University of California Press, 1967, pp. 281–297.


Kevin P. Murphy, *Machine learning: A probabilistic perspective*, The MIT Press, 2012.