Advanced Machine Learning

Nora Ouzir : nora.ouzir@centralesupelec.fr
Lucca Guardiola : lucca.guardiola@centralesupelec.fr

Oct. - Nov. 2020

CentraleSupélec

Course Structure and Evaluation

- 8 Lectures + Exam
- 1h30 lecture + 1h30 lab session in Python

Grading

- Written exam 50% (questions + research paper)
- Lab reports 50% (3 reports)

References

1. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer, 2011.
2. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition, Springer, 2017.
3. Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.

Links to course slides/lab sessions/data

- fredericpascal.blogspot.fr
- http://www-syscom.univ-mlv.fr/ chouzeno/

Content

1. Reminders on ML
2. Robust regression
3. Classification and supervised learning
4. Hierarchical clustering
5. Nonnegative matrix factorization
6. Mixture models fitting
7. Model order selection
8. Dimension reduction and data visualization

Today's Lecture

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

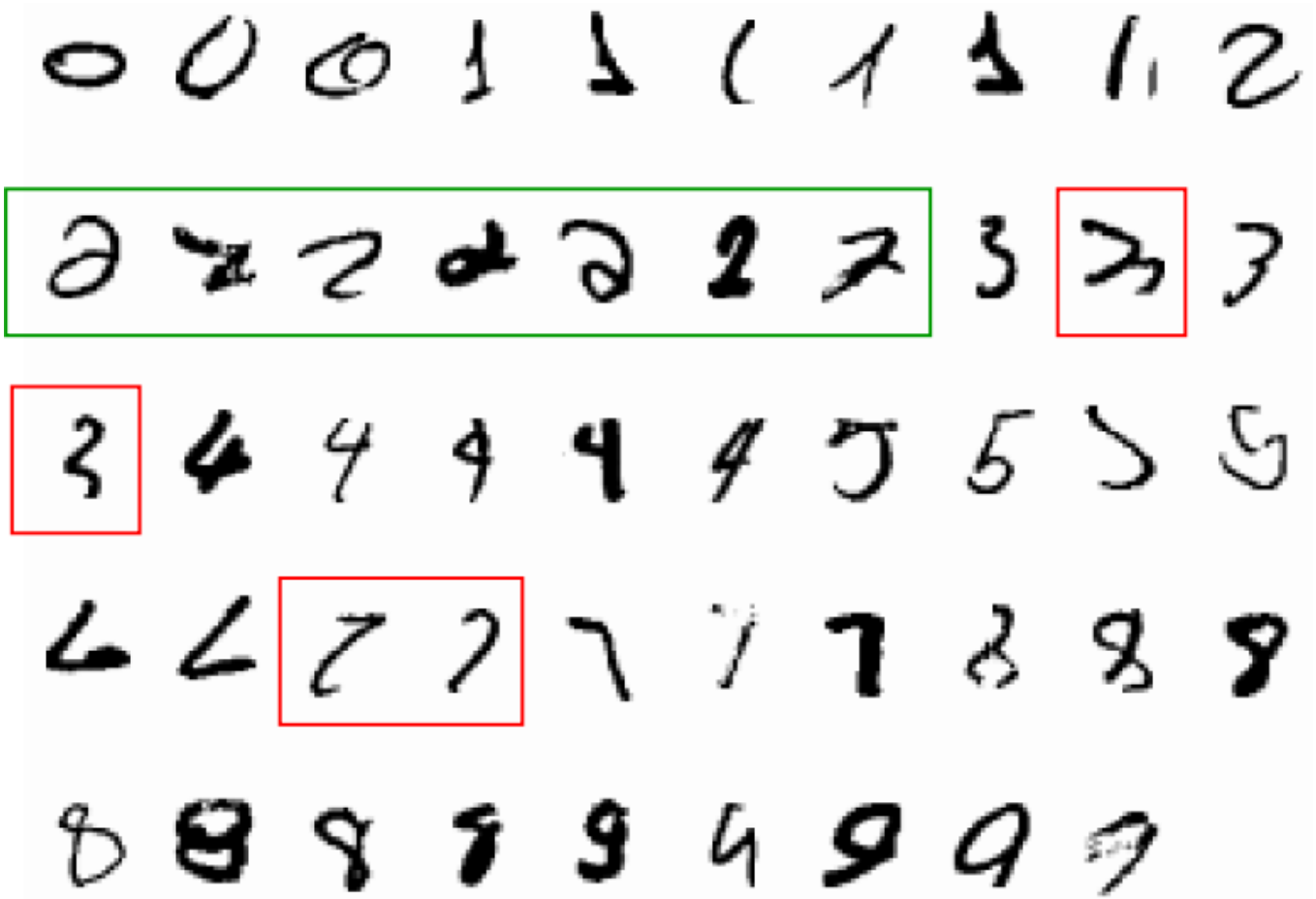
3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

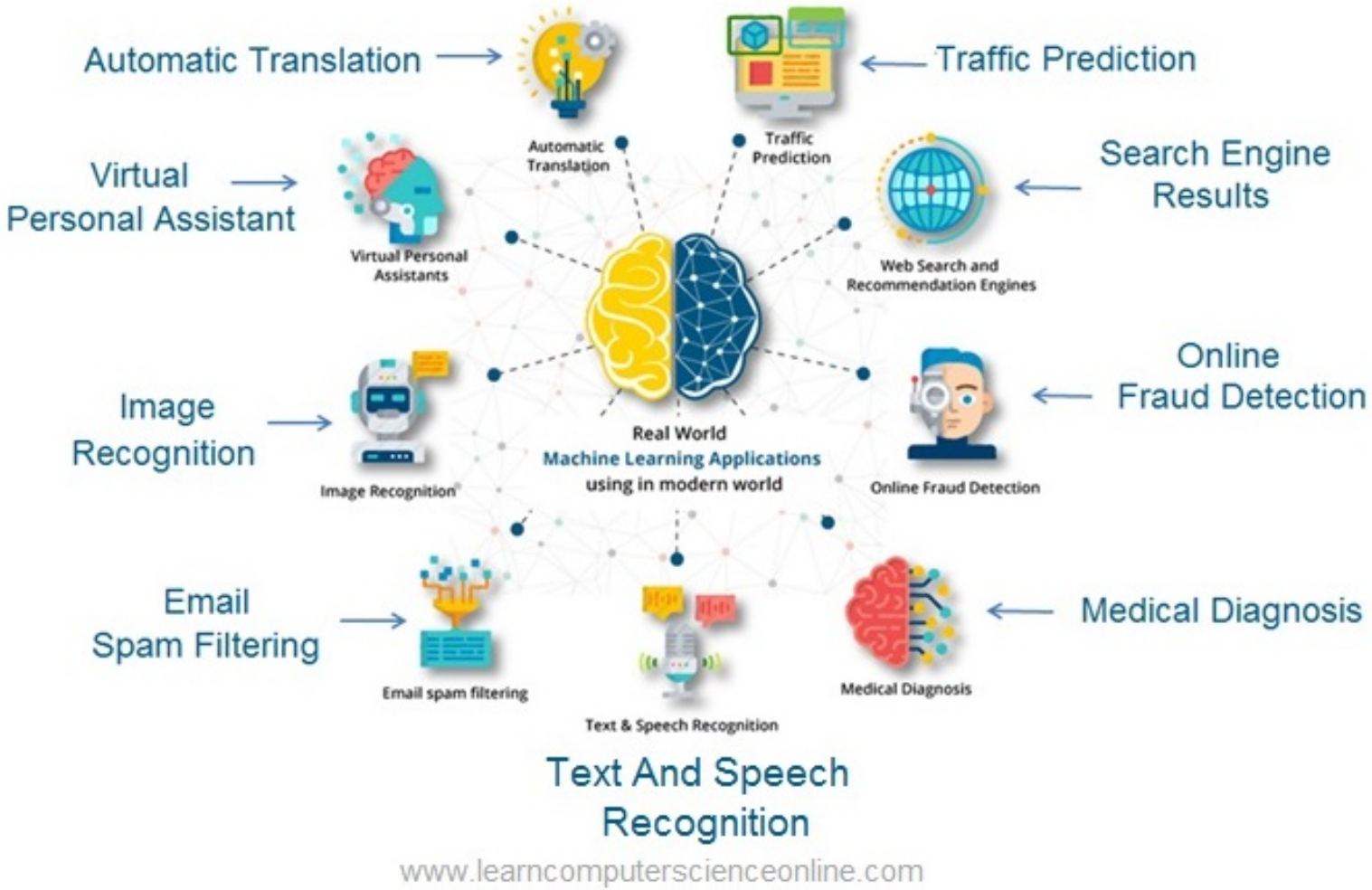
What is ML ?

- Perform tasks without explicit programming : learning
- Improve performance based on experience
- Learn from observed data by constructing stochastic models that can be used for making predictions and decisions

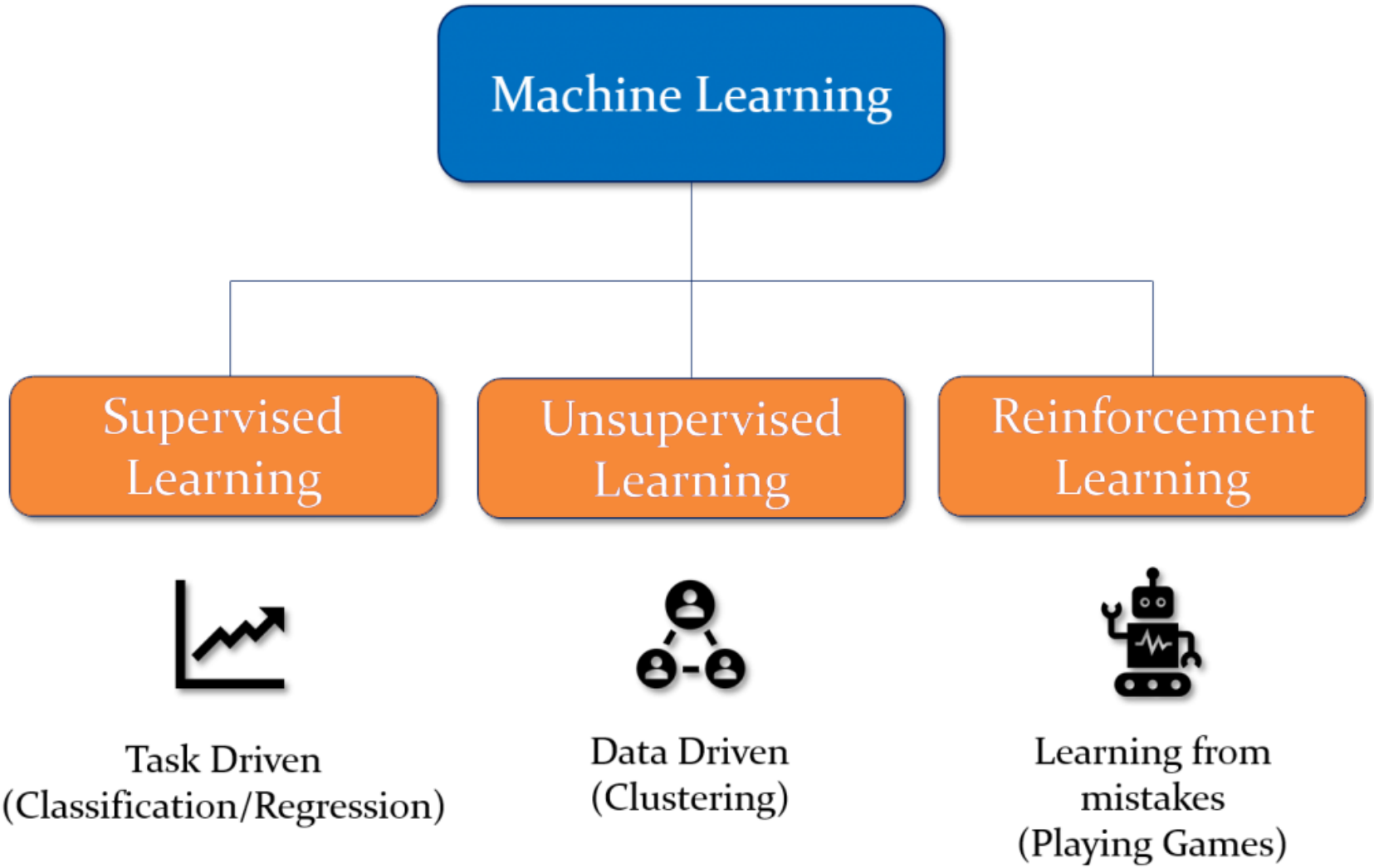
What makes a two ?



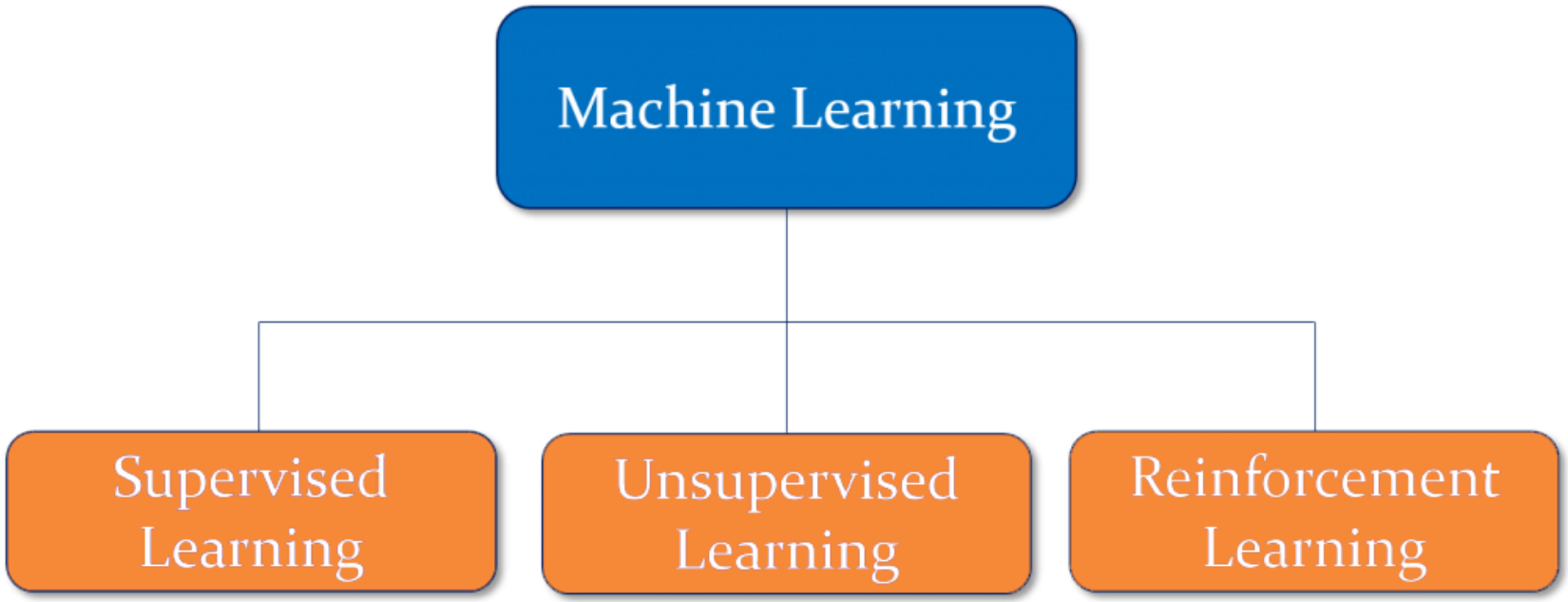
Real World Applications



Types of ML

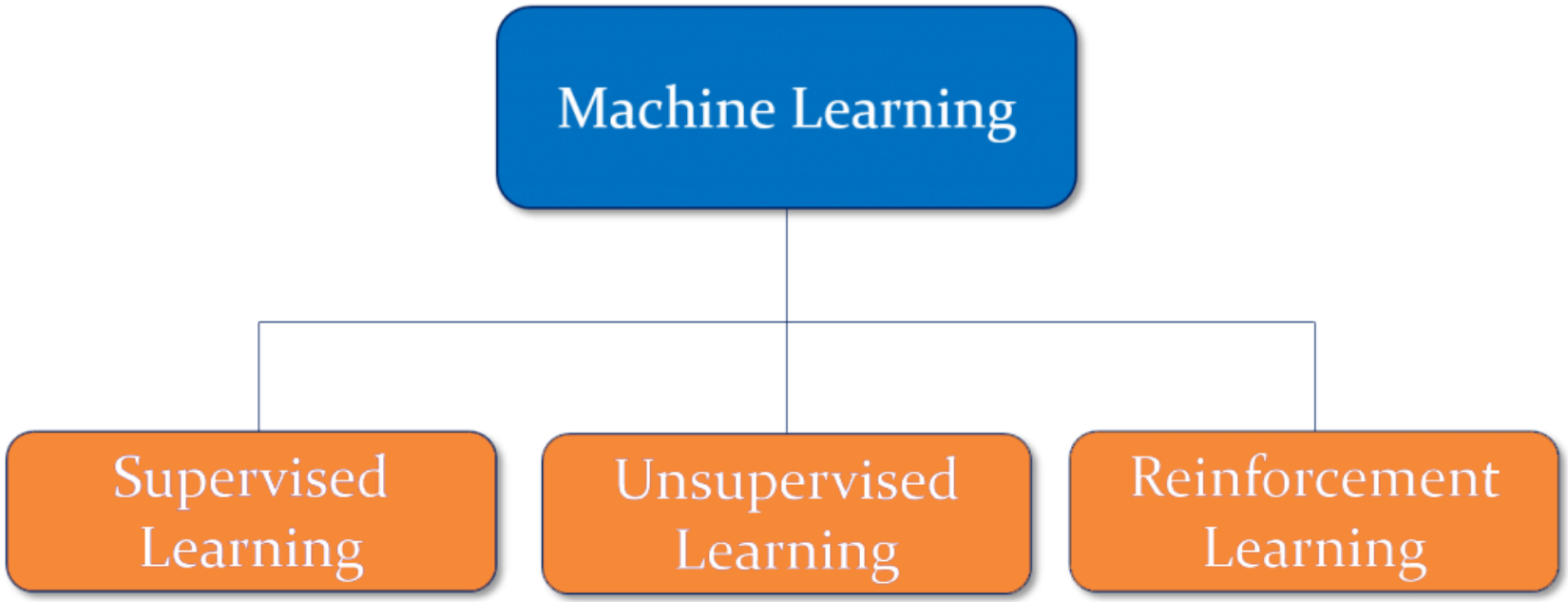

Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Task Driven
(Classification/Regression)
Data Driven
(Clustering)
Learning from mistakes
(Playing Games)

Types of ML



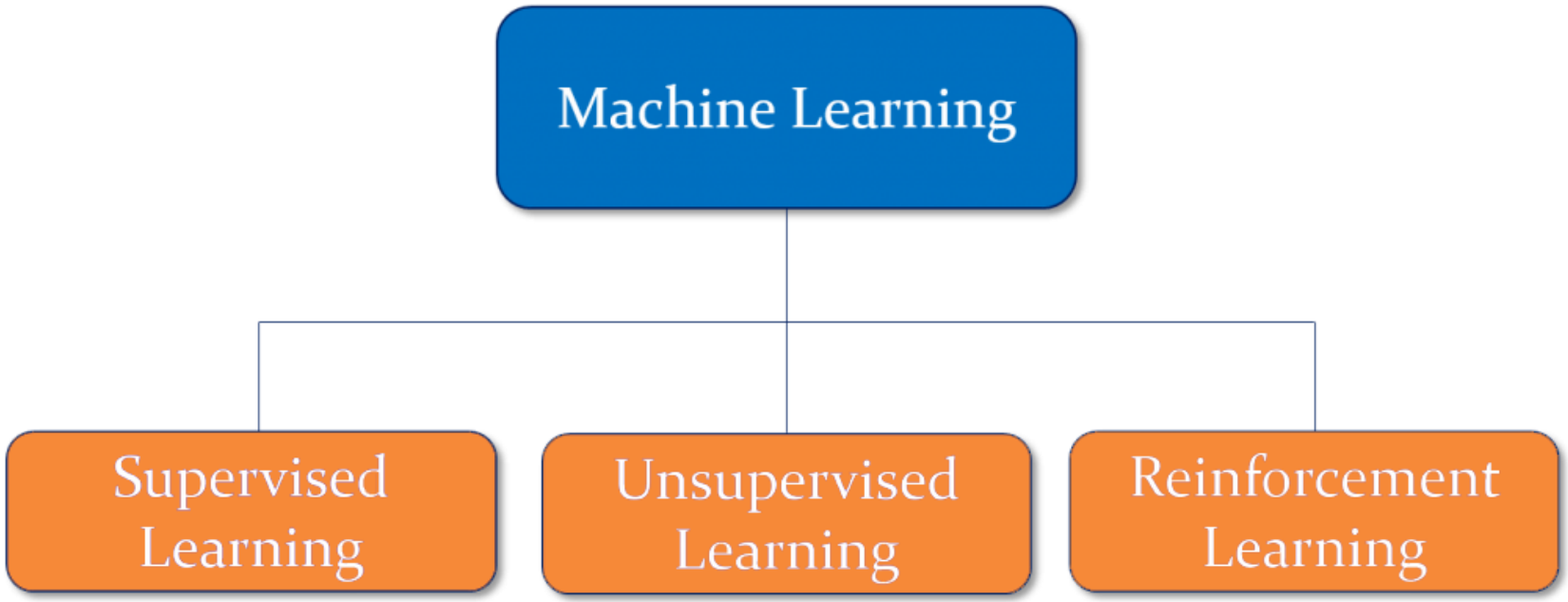
- Linear regression
- Ridge/LASSO regression
- K-NN
- Naive Bayes
- SVM
- Logistic regression
- Decision trees

Types of ML



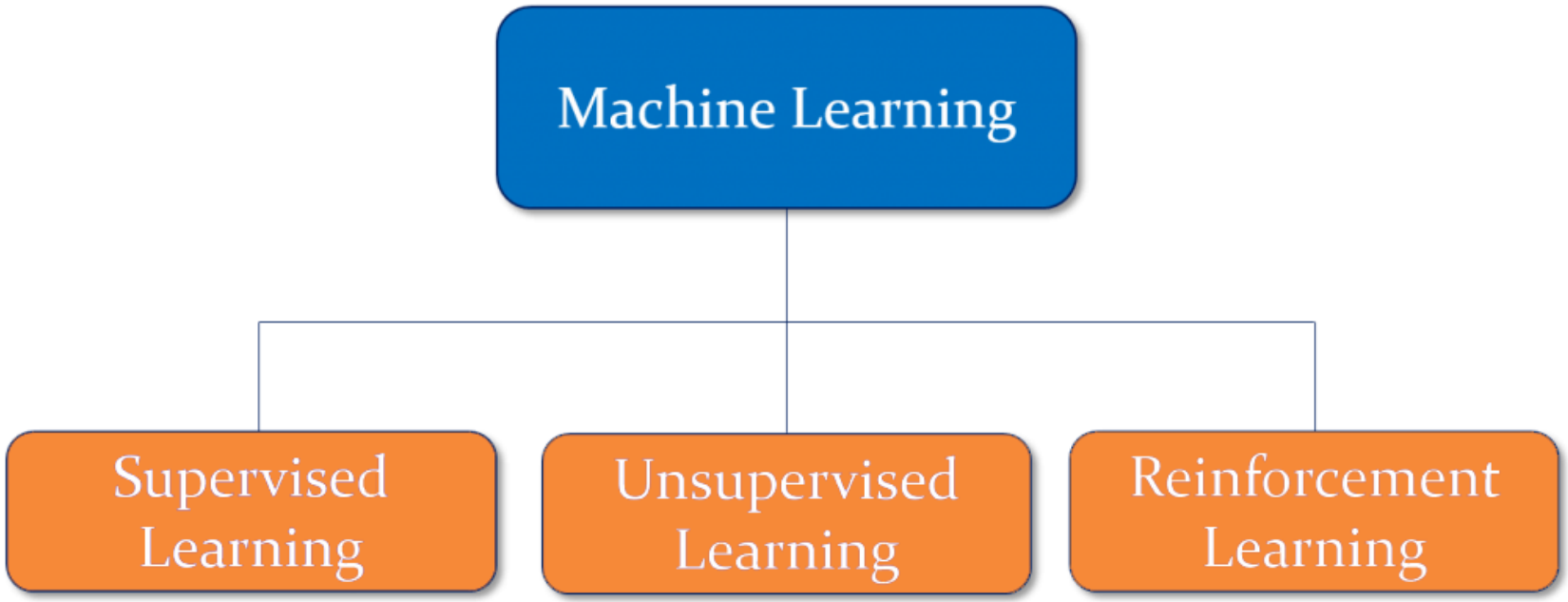
- K-means
- DBSCAN
- Hierarchical
- GMM
- PCA, SVD, NMF
- LDA
- t-SNE

Types of ML



- Q-learning
- Genetic Algorithm
- SARSA
- A3C
- DQN

Types of ML


Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Linear regression
Ridge/LASSO regression
K-NN
Naive Bayes
SVM
Logistic regression
Decision trees
K-means
DBSCAN
Hierarchical
GMM
PCA, SVD, NMF
LDA
t-SNE
Q-learning
Genetic Algorithm
SARSA
A3C
DQN

Content

1. Reminders on ML
2. Robust regression
3. Classification and supervised learning
4. Hierarchical clustering
5. Nonnegative matrix factorization
6. Mixture models fitting
7. Model order selection
8. Dimension reduction and data visualization

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Basics

Let us denote $T_n(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ or $\hat{\theta}_n$ an estimator of θ (or the true value θ_0 if needed).

Consistancy
An estimator $\hat{\theta}_n$ of $g(\theta)$ is strongly (resp. weakly) consistant if it P_{θ_0}-almost surely (resp. in proba.) converges towards $g(\theta_0)$, with $g : \Theta \to \mathbb{R}^p$.

Asymptotically unbiased
An estimator $\hat{\theta}_n$ of $g(\theta)$ is asymptotically unbiased if its limiting distribution is zero-mean, i.e.,

$$\exists c_n \to \infty \text{ s.t. } c_n \left(\hat{\theta}_n - g(\theta_0)\right) \xrightarrow[n\to\infty]{dist.} \mathbf{z} \text{ with } E_{\theta_0}[\mathbf{z}] = 0.$$

Remark: Different from *"unbiased at the limit"*: $E_{\theta_0}\left[\hat{\theta}_n\right] \xrightarrow[n\to\infty]{} g(\theta_0)$.

Basics

Asymptotically normal

$\hat{\theta}_n$ is asymptotically normal if

$$\sqrt{n}\left(\hat{\theta}_n - g(\theta_0)\right) \xrightarrow[n\to\infty]{dist.} \mathcal{N}\left(\mathbf{0}, \Sigma(\theta_0)\right)$$

where $\Sigma(\theta_0)$ (PDS) is the asymptotic CM of $\hat{\theta}_n$.
Remark: This implies that $\hat{\theta}_n$ is asymptotically unbiased.

Asymptotically efficient

An estimator is asymptotically efficient if it is asymptotically normal and if:

$$\Sigma(\theta_0) = \frac{\partial g}{\partial \theta^t}(\theta_0)\, I(\theta_0)^{-1} \frac{\partial g^t}{\partial \theta}(\theta_0)$$

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Method of Moments

Let a n-sample $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ i.i.d. with $\mathbf{x}_1 \sim P_\theta$ where $\theta \in \Theta \subset \mathbb{R}^d$ s.t. $E[|\mathbf{x}_1|]^d) < \infty$. Let us assume that:

$$m = \begin{pmatrix} m_1 \\ \vdots \\ m_d \end{pmatrix} = \begin{pmatrix} \phi_1(\theta_1, \cdots, \theta_d) \\ \vdots \\ \phi_d(\theta_1, \cdots, \theta_d) \end{pmatrix} = \phi \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_d \end{pmatrix}$$

where $m_k = E_\theta[\mathbf{x}^k]$. If function ϕ is invertible (with inverse ψ), one has:

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_d \end{pmatrix} = \begin{pmatrix} \psi_1(m_1, \cdots, m_d) \\ \vdots \\ \psi_d(m_1, \cdots, m_d) \end{pmatrix} = \psi \begin{pmatrix} m_1 \\ \vdots \\ m_d \end{pmatrix}$$

- $U_p \xrightarrow[n\to\infty]{a.s} m_p$ where $\forall p,\ U_p = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i^p$
- $\sqrt{n}\,(\mathbf{U} - \mathbf{m}) \xrightarrow[n\to\infty]{dist.} \mathcal{N}(\mathbf{0}, \mathbf{Z})$ where $\mathbf{U} = (U_1, \cdots, U_p)^t$, $\mathbf{m} = (m_1, \cdots, m_p)^t$.

Method of Moments

The estimator of the Method of Moments (MME) is defined as

$$\hat{\theta}_n = \begin{pmatrix} \hat{\theta}_{n1} \\ \vdots \\ \hat{\theta}_{nd} \end{pmatrix} = \begin{pmatrix} \psi_1(U_1, \cdots, U_d) \\ \vdots \\ \psi_d(U_1, \cdots, U_d) \end{pmatrix} = \psi \begin{pmatrix} U_1 \\ \vdots \\ U_d \end{pmatrix}$$

where $\forall p,\ U_p = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i^p$ with $\mathbf{x}_i$ are i.i.d.

Asymptotics of the MM estimator

If function ψ is differentiable, then

- $\hat{\theta}_n \xrightarrow[n\to\infty]{a.s} \theta$
- $\sqrt{n}\left(\hat{\theta}_n - \theta\right) \xrightarrow[n\to\infty]{dist.} \mathcal{N}\left(\mathbf{0}, \mathbf{A}(\theta)\right)$ where $\mathbf{A}(\theta) = \frac{\partial \psi}{\partial \theta^t}(m)\Sigma(\theta)\frac{\partial \psi^t}{\partial \theta}(m)$ with $m = \phi(\theta)$.

MME strongly consistant, asymptotically normal BUT generally NOT asymptotically efficient!

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Method of Maximum Likelihood

Assume a regular model + (A_5) +

(A_6) $\forall x \in \Delta$, for θ close to θ_0, $\log(f(x;\theta))$ is $3\times$ differentiable w.r.t. θ and

$$\left| \frac{\partial^3}{\partial\theta_j \partial\theta_k \partial\theta_l} \log\left(f(x;\theta)\right) \right| \leq M(x)$$

with $E_{\theta_0}\left[M(x)\right] < +\infty$.

Assume the model is identifiable, then $\forall \theta \neq \theta_0$, one has

$$P_{\theta_0}\left(L(\mathbf{x}_1,\ldots,\mathbf{x}_n;\theta_0) > L(\mathbf{x}_1,\ldots,\mathbf{x}_n;\theta)\right) \xrightarrow[n\to=\infty]{} 1$$

where $L\left(\mathbf{x}_1,\ldots,\mathbf{x}_n;\theta\right)$ is the LF.

The LF is maximum at the point θ_0...

Maximum Likelihood Estimation

Maximum Likelihood Estimator (MLE)
The MLE is defined by

$$T : (\mathbf{x}_1, \ldots, \mathbf{x}_n) \to \widehat{\theta}_n \in \arg\max_{\theta \in \Theta} L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta)\,.$$

The MLE has to verified the following likelihood equations!

$$\begin{cases} \dfrac{\partial}{\partial \theta} l(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta) & = & 0 \\ \dfrac{\partial^2}{\partial \theta \partial \theta^t} l(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta) & \leq & 0\,, \end{cases}$$

where $l(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta) = \log\left(L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta)\right)$

Let $g : \Theta \mapsto \mathbb{R}^p$. If $\hat{\theta}_n$ is a MLE of θ, then $g(\hat{\theta}_n)$ is also a MLE of $g(\theta)$.

The MLE is not necessary unique...

MLE asymptotics

Assume: identifiable model, (A_1), (A_2), $\theta_0 \in \Theta \neq \varnothing$, compact, and

- $x_1 \mapsto L(x_1, \theta)$ is bounded $\forall \theta \in \Theta$;
- $\theta \mapsto L(x_1, \theta)$ is continuous $\forall x_1 \in \Delta$;

$$\text{Thus, } \hat{\theta}_n^{ML} \xrightarrow[n\to\infty]{a.s} \theta_0 \text{ (Existence from a given } n_0)$$

Classical asymptotics

Assume: identifiable model, Θ open set of $\mathbb{R}^d$ and $(A_1) - (A_6)$.
Thus, $\exists\, \hat{\theta}_n^{ML}$ (from a given n_0) solution to the likelihood equations s.t.

$$\begin{cases} \hat{\theta}_n^{ML} \xrightarrow[n\to\infty]{a.s} \theta_0 \\ \sqrt{n}\left(\hat{\theta}_n^{ML} - \theta_0\right) \xrightarrow[n\to\infty]{dist.} \mathcal{N}\left(\mathbf{0}, I_1(\theta_0)^{-1}\right) \end{cases}$$

MLE asymptotics

Classical asymptotics

Assume: identifiable model, Θ open set of $\mathbb{R}^d$ and $(A_1) - (A_6)$
AND $g : \mathbb{R}^d \to \mathbb{R}^p$ differentiable
Thus, $\exists\, \hat{\theta}_n^{ML}$ (from a given n_0) solution to the likelihood equations s.t.

$$\begin{cases} g\left(\hat{\theta}_n^{ML}\right) \xrightarrow[n\to\infty]{a.s} g\left(\theta_0\right) \\ \sqrt{n}\left(g\left(\hat{\theta}_n^{ML}\right) - g\left(\theta_0\right)\right) \xrightarrow[n\to\infty]{dist.} \mathcal{N}\left(\mathbf{0}, \frac{\partial g}{\partial \theta^t}(\theta_0)\, I_1(\theta_0)^{-1} \frac{\partial g^t}{\partial \theta}(\theta_0)\right) \end{cases}$$

Conclusions

The MLE is strongly consistant, asymptotically normal and asymptotically efficient.

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Bayesian estimation

Principles: Philosophy is different from previous MM/ML estimation approaches (frequentist methods). The purpose is the same: estimating an unknown parameter $\theta \in \mathbb{R}$ or $\mathbb{R}^p$ thanks to the sample $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ likelihood (parameterized by θ) and an a prior distribution $p(\theta)$.
So, θ is assumed to random...

- Key Idea: Minimize a cost function $c(\theta, \hat{\theta})$ that represents the error between θ and its estimator $\hat{\theta}$.
- Reminders: A posteriori distribution / posterior distribution

$$p(\theta|\mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta)\, p(\theta)}{f(\mathbf{x}_1, \ldots, \mathbf{x}_n)} = \frac{L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta)\, p(\theta)}{\displaystyle\int_{\mathbb{R}^p} L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta)\, p(\theta)\, d\theta}$$

$$\propto L(\mathbf{x}_1, \ldots, \mathbf{x}_n; \theta)\, p(\theta)$$

MMSE estimator

MMSE estimator (mean of the posterior PDF) is the estimator that minimizes the MSE as the cost function: $\mathbf{c}(\theta, \hat{\theta}) = E\left[\left(\theta - \hat{\theta}\right)^2\right]$.

When $\theta \in \mathbb{R}$

$$E\left[\left(\theta - \hat{\theta}_{MMSE}(\mathbf{x})\right)^2\right] = \min_{\pi} E\left[\left(\theta - \pi(\mathbf{x})\right)^2\right]$$

with $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, hence the MMSE estimator is $\hat{\theta}_{MMSE}(\mathbf{x}) = E[\theta|\mathbf{x}]$

When $\theta \in \mathbb{R}^p$

The MMSE estimator $\hat{\theta}_{MMSE}(\mathbf{x}) = E[\theta|\mathbf{x}]$ minimizes the quadratic cost

$$E\left[\left(\theta - \pi(\mathbf{x})\right)^t \mathbf{Q}\left(\theta - \pi(\mathbf{x})\right)\right]$$

for any symmetric definite positive matrix $\mathbf{Q}$ (and in particular for $\mathbf{Q} = \mathbf{I}_p$, the identity matrix).

MAP estimator

When $\theta \in \mathbb{R}$

The MAP estimator $\hat{\theta}_{MAP}(\mathbf{x})$ minimizes the average of a "uniform" cost function

$$c((\theta - \pi(\mathbf{x}))) = \begin{cases} 0 \text{ if } |\theta - \pi(\mathbf{x})| \leq \Lambda/2 \\ 1 \text{ if } |\theta - \pi(\mathbf{x})| > \Lambda/2 \end{cases}$$

and is defined by

$$c(\left(\theta - \hat{\theta}_{MAP}(\mathbf{x})\right)) = \min_{\pi} c((\theta - \pi(\mathbf{x})))$$

If Λ is arbitrary small, $\hat{\theta}_{MAP}(\mathbf{x})$ is the value of $\pi(\mathbf{x})$ which maximizes the posterior $p(\theta|\mathbf{x})$ hence its name MAP estimator. $\hat{\theta}_{MAP}(\mathbf{x})$ is computed by setting to zero the derivative of $p(\theta|\mathbf{x})$ (or of its log) with respect to θ.

$\theta \in \mathbb{R}^p$

Determine the values of θ_i which make the partial derivatives of $p(\theta|\mathbf{x})$ (or of its logarithm) with respect to θ_i equal to zero.

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Supervised Learning Principle

- Let $\boldsymbol{X} = (x_1, ..., x_N)$ and $\boldsymbol{Y} = (y_1, ..., y_N)$ be a set of N input/ouput training samples.
- Estimate/learn a prediction function $y = f(x)$
- $\boldsymbol{Y}$ known : supervised

Applications

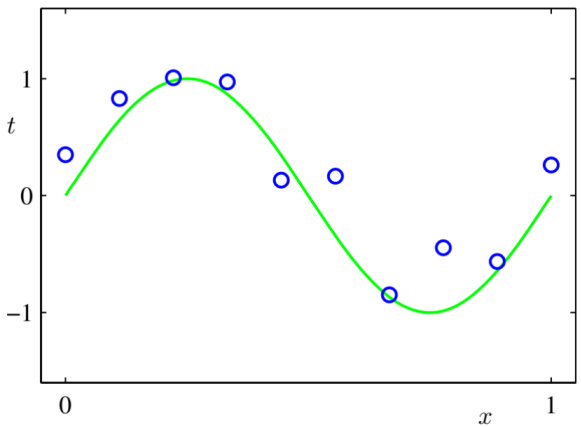
Stock price prediction, image classification, ...

The labels are known!

Regression vs Classification

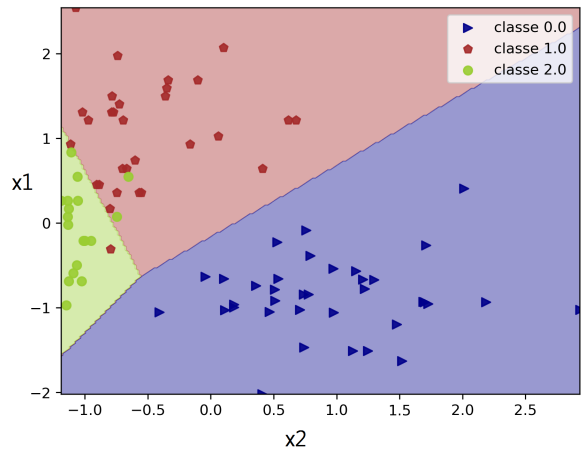
Regression

- $y \in \mathbb{R}$ is a continuous variable
- Predict a numerical value



Classification

- labels are discrete variables
- Binary Classification
  $y \in \{0, 1\}, y \in \{-1, 1\}, \ldots$
- Multiclass $y \in \{1, \ldots, K\}$



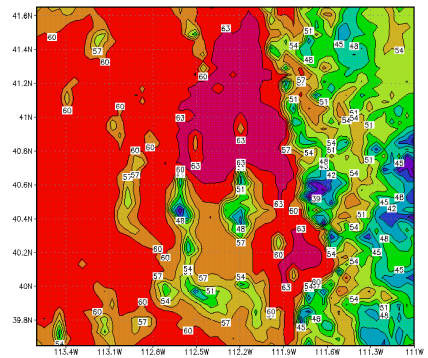
Regression Applications

Financial data

- ***x***: economical, social, political variables
- ***y***: stock price

Weather prediction

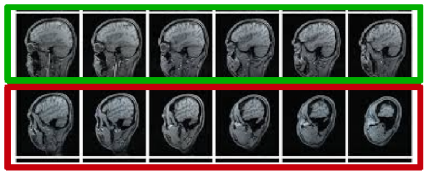
- ***x***: location, ...
- ***y***: temperature value



Classification Applications in Computer Vision

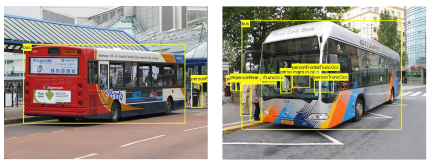
Image classification

- x: pixels, voxels
- y: binary or multiclass labels



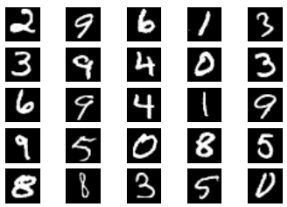
Object detection

- Cars, animals, faces...
- Diagnosis: presence of tumors
- y: objects, contours, boxes, ...



Digit recognition

- Multiclass classification $x \in \{0, ..., 9\}$



Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Discriminant Analysis: a MAP approach

Maximize the *posterior probability*: the probability that the observation is in the k-th class given the feature $\boldsymbol{X}$ takes on a specific value x

$$\hat{C}(x) = \arg\max_k P(C = k|\boldsymbol{X} = x)$$

- The prior probability of class k is π_k
- $f_k(x)$ conditional distribution of $\boldsymbol{X}$ in $C = k$
- Bayes theorem

$$P(C = k|\boldsymbol{X} = x) = \frac{f_k(x)\pi_k}{\sum_{i=1}^{K} f_i(x)\pi_i}$$

Discriminant Analysis

Bayes Classifier gives

$$\hat{C}(x) = \arg\max_k f_k(x)\pi_k$$

What do we need in order to compute $\hat{C}(x)$?

- The prior probability π_k: empirical frequencies of the training set $\hat{\pi_k} = \frac{N_k}{N}$
- Choose the distribution $f_k(x)$

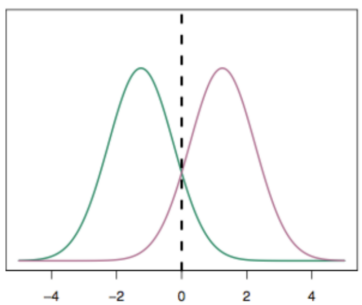
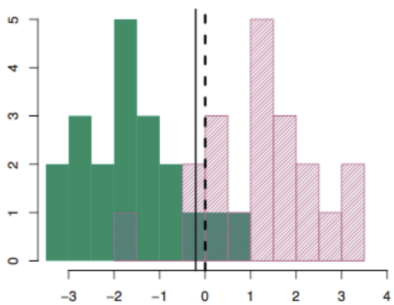
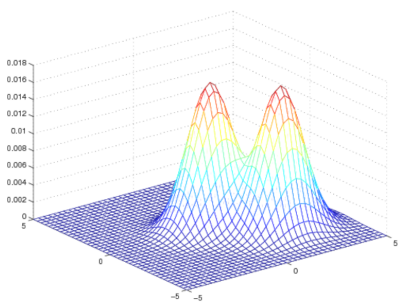
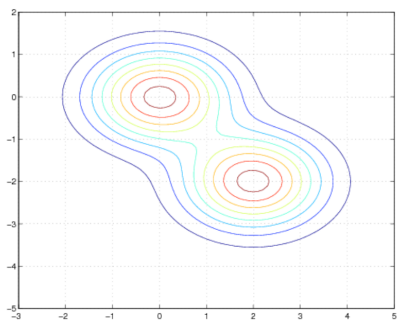
  Gaussian, Mixtures of Gaussians, non-parametric, product of marginal densities (naive Bayes), ...

Linear Discriminant Analysis (LDA)

- Gaussian densities

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp -\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)$$

- Equal covariances for all classes: $\Sigma_k = \Sigma,\ \forall k$

LDA: let's compute $\hat{C}(x)$

$$
\begin{aligned}
\hat{C}(x) &= \arg\max_k P(C = k | \boldsymbol{X} = x) \\
&= \arg\max_k f_k(x)\pi_k = \arg\max_k \log[f_k(x)\pi_k] \\
&= \arg\max_k -\left[\log[(2\pi)^{p/2}|\Sigma|^{1/2}]\right. \\
&- \frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k) \\
&= \arg\max_k -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log(\pi_k) \\
&= \arg\max_k x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k - \frac{1}{2}x^T\Sigma^{-1}x + \log(\pi_k)
\end{aligned}
$$

LDA: let's compute $\hat{C}(x)$

$$
\begin{aligned}
\hat{C}(x) &= \arg\max_k P(C = k | \boldsymbol{X} = x) \\
&= \arg\max_k f_k(x)\pi_k = \arg\max_k \log[f_k(x)\pi_k] \\
&= \arg\max_k -\left[\log[(2\pi)^{p/2}|\Sigma|^{1/2}\right] \\
&- \frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k) \\
&= \arg\max_k -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log(\pi_k) \\
&= \arg\max_k x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k - \frac{1}{2}x^T\Sigma^{-1}x + \log(\pi_k)
\end{aligned}
$$

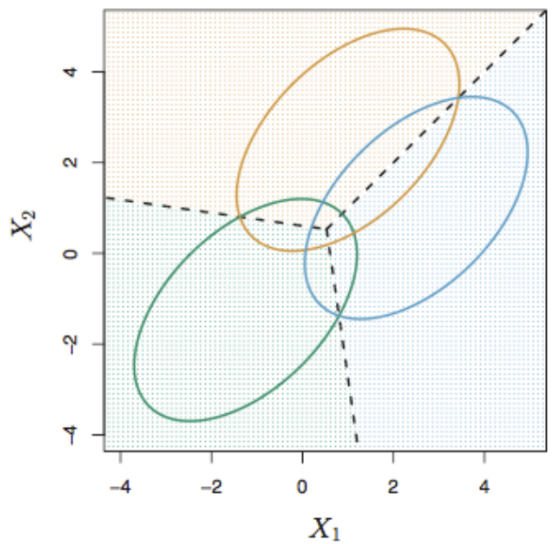
LDA: Linear decision boundary

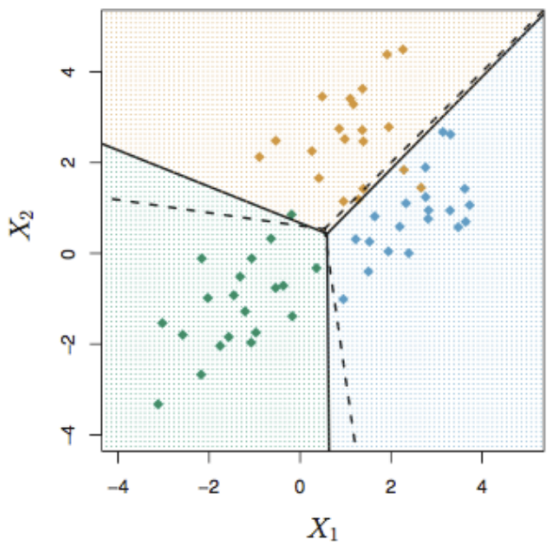
Linear discriminant function

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log(\pi_k)$$

The decision boundary: binary case

- $\{x : \delta_1(x) = \delta_2(x)\}$
- $\log\left(\frac{\pi_1}{\pi_2}\right) - \frac{1}{2}(\mu_1 - \mu_2)^T\Sigma^{-1}(\mu_1 - \mu_2) + x^T\Sigma^{-1}(\mu_1 - \mu_2) = 0$





LDA: a supervised approach

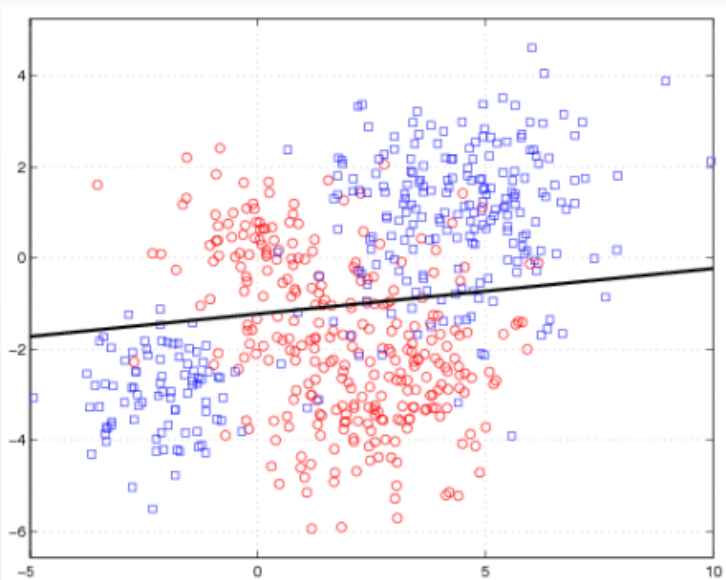
The training set allows us to:

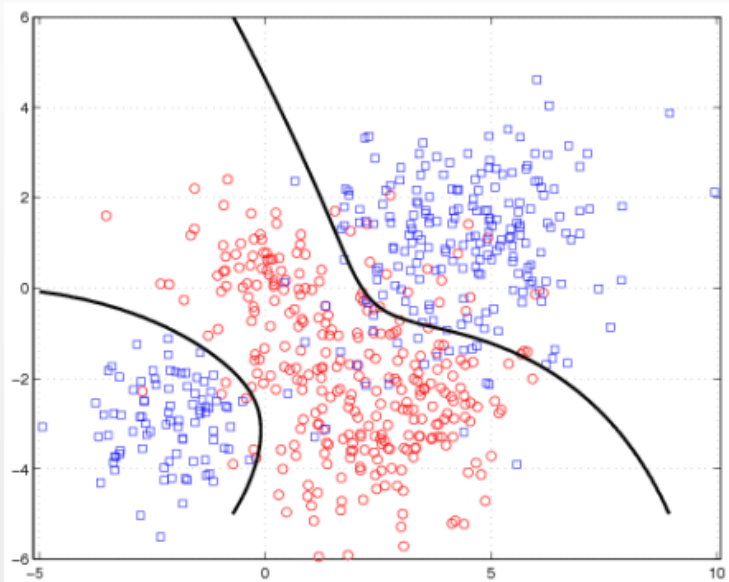
- Estimate the prior probabilities $\hat{\pi}_k = N_k/N$
- Estimate the Gaussian distributions, *i.e.*,
  - Mean vector: $\hat{\mu}_k = \frac{1}{N_k} \sum_{x_i \in K} x^i$
  - Covariance matrix: $\hat{\Sigma} = \frac{1}{N-K} \sum_{k=1}^{K} \sum_{x^i \in K} (x^i - \hat{\mu}_k)(x^i - \hat{\mu}_k)^T$

In conclusion, LDA...

- Is a MAP approach
- Uses Gaussian conditional densities
- Assumes equal covariance matrices
- Poor results when : heterogeneous Σ_k, non-linear boundaries

Example: LDA vs Mixture of Gaussians





Left: Decision boundaries by LDA. Right: Decision boundaries obtained by modeling each class by a mixture of two Gaussians.

http://www.stat.psu.edu/jiali

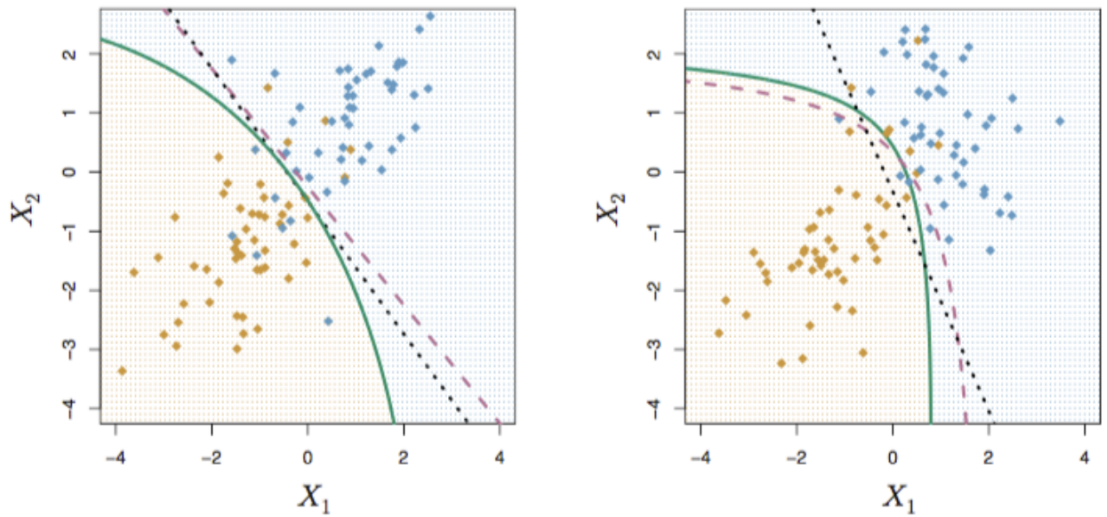
Quadratic Discriminant Analysis (QDA)

- Gaussian densities but separate covariance matrices Σ_k
- The discriminant function is quadratic:

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

QDA vs LDA

Fits better but more parameters to estimate!

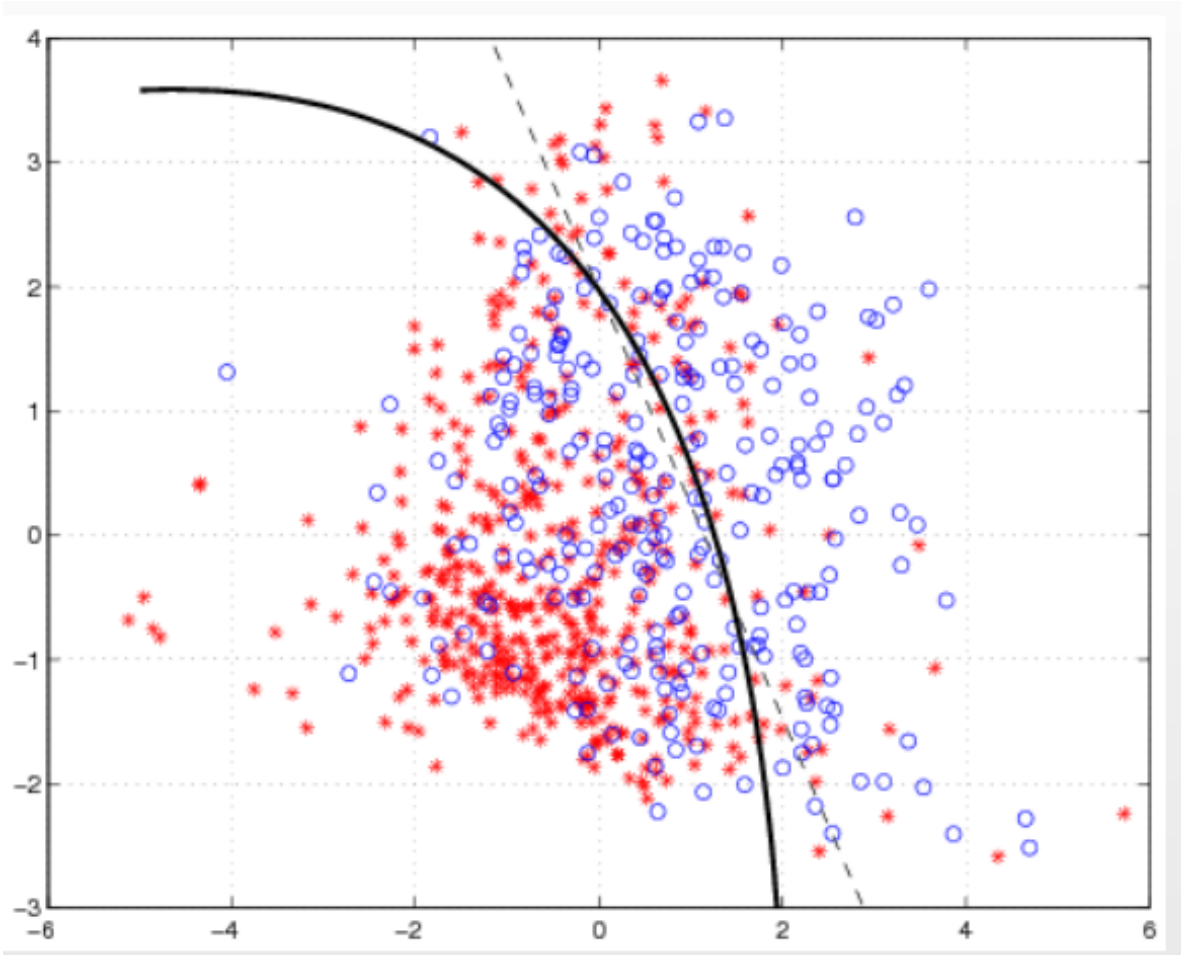


Diabetes Dataset

- 2 input variables obtained by PCA (8 original)
- Prior probabilities: $\hat{\pi}_1 = 0.651$ and $\hat{\pi}_2 = 0.349$
- $\hat{\mu}_1 = (0.4035, 0.1935)^T$ and $\hat{\mu}_1 = (0.7528, 0.3611)^T$
- $$\hat{\Sigma} = \begin{pmatrix} 1.7925 & 0.1461 \\ 0.1461 & 1.663 \end{pmatrix}$$
- $$\hat{\Sigma}_{1/2} = \begin{pmatrix} 1.6769/2.0087 & 0.04611/0.3330 \\ 0.0461/0.3330 & 1.5964/1.7887 \end{pmatrix}$$
- The classification rule is

$$\hat{C}(x) = \begin{cases} 1 & 1.1443 - x1 - 0.5802x2 \geq 0 \\ 2 & \text{otherwise} \end{cases}$$

Diabetes Dataset:



http://www.stat.psu.edu/jiali

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

Unsupervised Learning Principle

- Let $\boldsymbol{X} = (x_1, ..., x_N)$ be a set of $\boldsymbol{N}$ training samples.
- Discover/learn how the data is organized
- Clustering: Extract homogeneous categories
- $\boldsymbol{Y}$ unknown : unsupervised

Applications

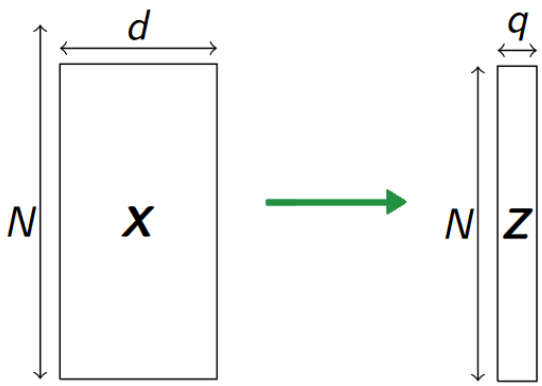
Image segmentation, targeted marketing, text mining, data visualization ...

The labels are unknown!

Dimension reduction vs Clustering

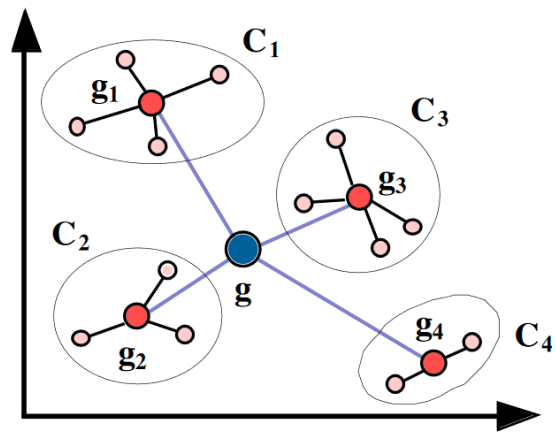
Dimension reduction

- Project $\boldsymbol{X} \in \mathbb{R}^{N,d}$ onto $\boldsymbol{Z} \in \mathbb{R}^{N,q}$ with $q < d$
- Visualize, denoise, reduce computational cost, ...



Clustering

- Groupe similar samples $\boldsymbol{X}^i$ into clusters C_k
- Based on a dissimilarity metric $\mathcal{D}(C_1, C_2)$

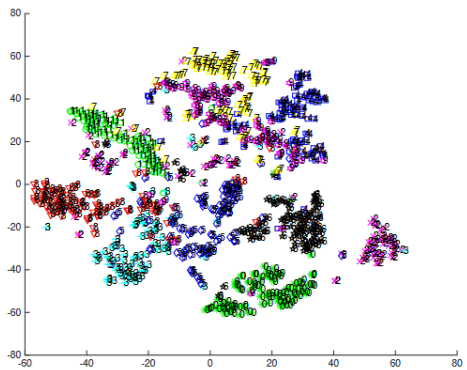


Dimension Reduction Applications

Data visualization

- X: high-dimensional data
- $Z \in \mathbb{R}^{N,q}$ s.t $q = 2$ or $q = 3$

$d = 784$



$q = 2$

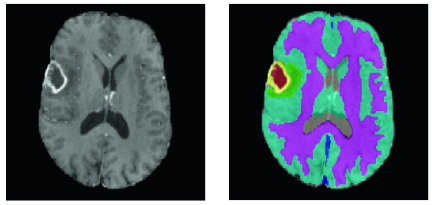
Clustering Applications

Market segmentation

- x: purchase history
- C_k: market segments

Medical image segmentation

- x: image pixels, voxels
- C_k: blood, muscle, tumor, ...



Text mining

- x: text, e-mails, ...
- C_k: folders, themes, ...

Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

K-means

- Partition the data into K clusters
- Find K clusters and their center μ_k that minimize the cluster within distance J_w
- J_w can be defined as

$$J_w = \Sigma_{k=1}^{K} \Sigma_{x^i \in C_k}^{K} \|\boldsymbol{x}^i - \mu_k\|^2$$

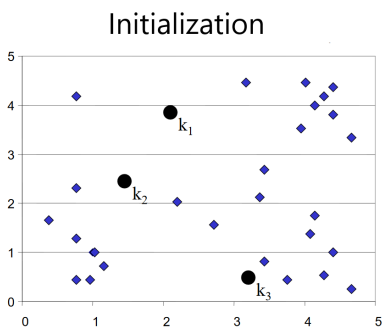
How can we solve this problem?

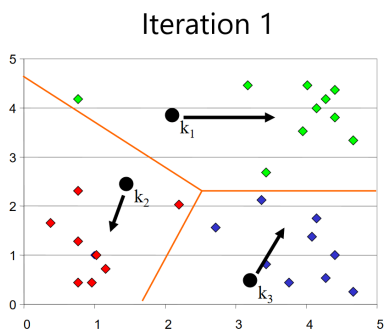
- NP-hard problem!
- Local solution is obtained with k-means ($O(tKN)$)

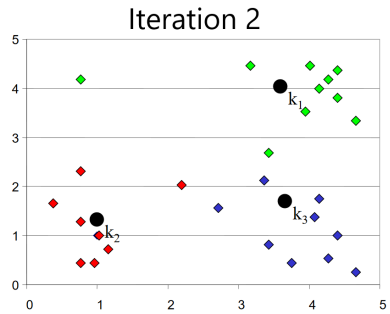
K-means algorithm

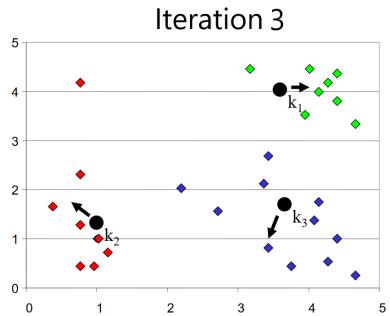
Iterative minimization

1. Initialize K cluster centers μ_k
2. Assign each $\boldsymbol{x}^i$ to the nearest cluster (nearest center)
   e.g., $s_i \leftarrow \arg\min_k \|\boldsymbol{x}^i - \mu_k\|^2$
3. Re-estimate K cluster centers
   e.g., $\mu_k = \frac{1}{K} \sum_{x^i \in c_k} \boldsymbol{x}^i$
4. Repeat until stopping criterion is reached





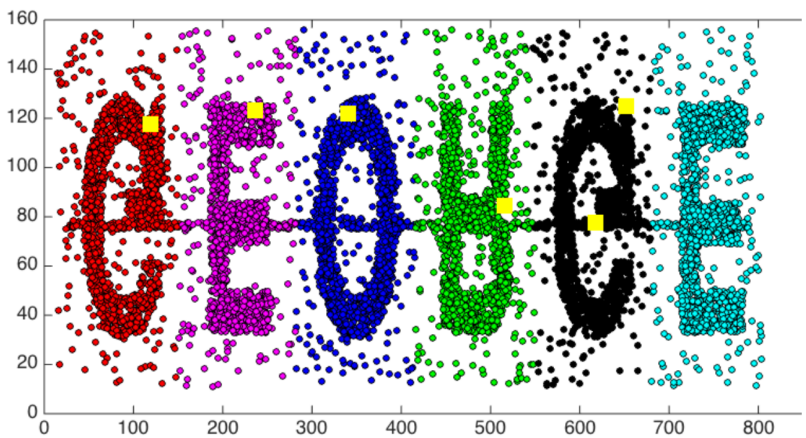


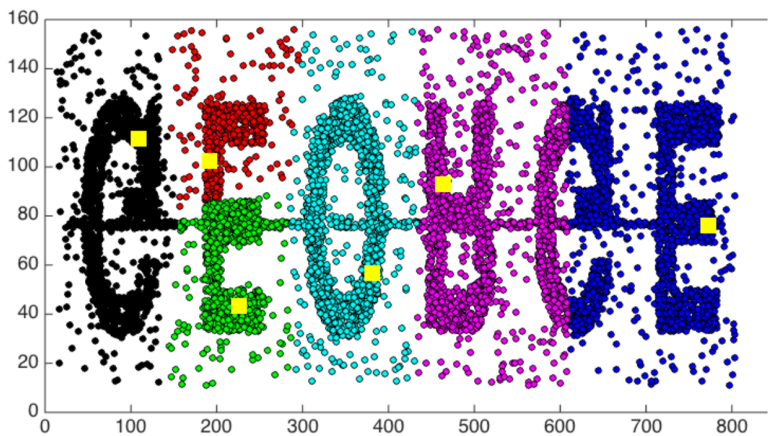


K-means: some weaknesses

k-means is simple but ...

- Solution depends on initialization
- Need to know *K* in advance
- Can't handle noise or outliers : *non-robust*
- Fails with clusters of *non-convex* shapes





Today's course

1. Background on ML

2. Reminders of probability and estimation
    1. Method of Moments
    2. Maximum Likelihood Estimation
    3. Bayesian estimation - MAP and MMSE

3. Supervised learning
    1. Discriminant Analysis

4. Unsupervised learning
    1. Clustering with k-means
    2. Dimension reduction and PCA

PCA principle

Orthogonal projection $\boldsymbol{P}$ of the data $\boldsymbol{X} \in \mathbb{R}^{N,d}$ into a subspace of dimension $q < d$ s.t the variance of the projected data is maximized

$$\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{P}^T + noise$$

Classic matrix computation tools

- Eigen values of the covariance matrix : capture variance direction and scale
- Singular Value Decomposition

Two main approaches

- Minimization of the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{x}$
- Maximization of variance: q largest eigen values of Σ

Find directions of maximum variance!