Model-based Statistical Learning

Pr. Charles BOUVEYRON

Professor of Statistics
Chair of the Institut 3IA Côte d'Azur
Université Côte d'Azur & Inria

charles.bouveyron@univ-cotedazur.fr
@cbouveyron

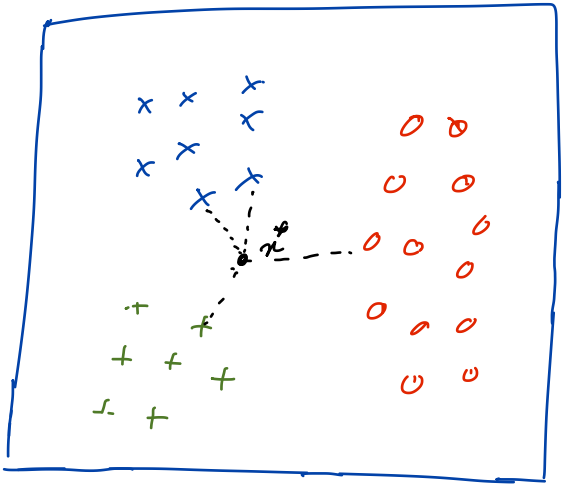
k-nearest neighbours (kNN):

k-NN is a very basic but still reference algorithm for classification. The spirit of k-NN is to compare any new observation x^* to the learning data and classify it according to its neighbors.

Algorithm: input x^*, k

(i) find in the learning data set (X, Y) the k nearest neighbors of x^*

(ii) assign x^* to the class which is the most represented among the neighbors.



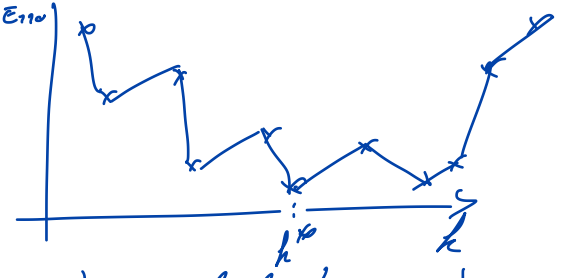
$k = 4$

$\hat{y}^{\text{test}}_{k\text{-NN}} = \text{"blue"}$.

Rmk: the choice of k in k-NN is a very sensitive question since it may end up with different classifications for different values of k.

$\Rightarrow$ the appropriate way to choose k is to rely on cross-validation.

Rmk: even with CV, the choice of k may be quite difficult because the curve of the classif. error according to k will be quite irregular



Rmk: k-NN can be applied to any types of data; if we are able to define a distance

- continuous data:
  - euclidean
  - Manhattan
- graph:
  - shortest path
  - average random walk
- text:
  - ...

Summary of k-NN:

⊕ single and easy to understand method

⊕ it can be applied to different data types

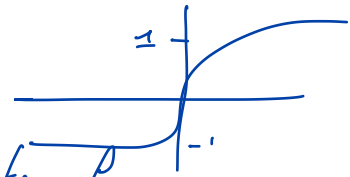
⊖ the choice of k is sensitive and not easy even with CV.

⊖ need to keep the whole learning data set and to find the neighbors in this large data set (which may be long!).

Logistic regression:

Logistic regression is, as LDA, one of the reference method for classification and is still very popular in many domains (even GAFAs). This method is however limited to the binary classification case.

The rational of logistic regression is to transform a classification problem as a regression one thanks to the logistic function.



Rmk: Logistic regression can also be seen as a very simple Neural Network with only an activation function.

The statistical model of Log. reg is:

$$\text{Logit}\left(P(Y=1|X=x)\right) = \alpha + \beta^t X + \varepsilon \quad (*)$$

and $\text{Logit}(p) = \log\left(\frac{p}{1-p}\right)$ | with $\varepsilon \sim N(0,\sigma^2)$

$\Rightarrow$ Learning this model from data resumes to estimating the model parameters $\theta = \{\alpha, \beta, \sigma^2\}$.

This can be done thanks to the maximum likelihood (ML) technique.

Unfortunately, the direct optimization of the likelihood is not possible and we have to rely to a numerical optimization procedure (stochastic gradient, ...) to get $\hat{\theta}_{ML}$

$\Rightarrow$ Once the model is learned, it is possible to use it to make a prediction for any new observation x^*.

As for LDA, we will rely on the maximum a posteriori rule (MAP):

MAP rule:

$$\hat{y}^* = \underset{k=1,2}{\operatorname{argmax}} \; P(Y=k \mid X=x^*)$$

where $P(Y=1 \mid X=x^*)$ can be obtained from equation (*): $\operatorname{logit}\left(P(Y=1 \mid X=x^*)\right) = \hat{\alpha} + \hat{\beta}^t x^*$

$$\Leftrightarrow \log\left(\frac{P(Y=1)}{P(Y=-1)}\right) = \hat{\alpha} + \hat{\beta}^t x^*$$

$$\Leftrightarrow \frac{P(Y=1)}{P(Y=-1)} = \exp\left(\hat{\alpha} + \hat{\beta}^t x^*\right)$$

$$\Leftrightarrow P(Y=1) = \left(1 - P(Y=1)\right) \exp(\text{---})$$

$\Leftrightarrow \quad P(Y=1) = \exp(\text{---}) - P(Y=1)\exp(\text{---})$

$\Leftrightarrow \quad P(Y=1)\left(1+\exp(\text{---})\right) = \exp(\text{---})$

$\Leftrightarrow \quad P(Y=1 \mid X=x^*) = \dfrac{\exp\left(\hat{\alpha}+\hat{\beta}^t x^*\right)}{1+\exp\left(\hat{\alpha}+\hat{\beta}^t x^*\right)}$

if $P(Y=1 \mid X=x^*) > 0.5 \Rightarrow \hat{y}^* = 1$

otherwise $\hat{y}^* = -1$.

Summary on logistic regression:

⊕ a simple method without tuning param.

⊕ no assumption on the class distributions

⊕ the method outputs probabilities to belong to the classes.

⊖ limited to only two classes.

⊖ it requires the use of an external optimization algorithm.

Outline

1. Introduction

2. Reminder on the learning process

3. Model-based statistical learning

4. Linear models for classification

5. Mixture models and the EM algorithm

(...)

The task of clustering is the same as in supervised classification, i.e. predict a categorical target variable y, but without having access to examples of both x and y for learning.

(Sup.) classification

$(x, y) \xrightarrow{\text{learn}} \hat{f} \xrightarrow{\text{predict}} \hat{f}(x^{(*)}) = \hat{y}^{(*)}$

(Unsup.) clustering

$x \xrightarrow{\text{learn}} \hat{f}(x) = \hat{y}$

$\Rightarrow$ Clustering is by nature a more challenging task compared to classification.

Definition: the goal of clustering is to form groups such that:

- data in the same group should behave in a similar way
- data in different groups should behave in different manners.

This task is simple but of course heavily combinatorial.

↳ this is why we needed to develop clustering algorithm which approximate the best solution.

Among the possible algorithms for clustering:

- k-means.
- DB-scan
- spectral clustering.

} non model-based techniques

- model-based clustering.

↳ most of those techniques rely on two ingredients:

- the mixture model
- the EM algorithm.

The mixture model

The mixture model is a family of statistical models that assume that the overall p.d.f is a mixture of several prob. density functions:

$$p(x) = \sum_{k=1}^{K} \pi_k \, p_k(x)$$

where $p_k()$ is a specific p.d.f and the weights π_k are such that $\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$.

The mixture model

The mixture model is a class of models which extremely flexible. Some examples:

* $p(x) = 0.6\, \mathcal{N}(x; 0, 1) + 0.4\, \mathcal{U}_{[0,1]}(x)$
* $p(x) = 0.3\, \mathcal{N}(x; 0, 1) + 0.3\, \mathcal{N}(x; 2, 1) + 0.4\, St(x, \dots)$
* $p(x) = \sum_{k=1}^{K} \pi_k\, \mathcal{N}(x; \mu_k, \Sigma_k)$ } This model is known as the Gaussian Mixture model (GMM)
* $p(x) = \sum_{k=1}^{K} \pi_k\, \mathcal{P}(x; \lambda_k)$ Poisson Mixture Model
* $p(x) = \sum_k \pi_k\, B(\ ) \dots$

The Gaussian mixture model (GMM)

The GMM model is clearly the most popular mixture model:

$$p(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x; \mu_k, \Sigma_k)$$

where μ_k is the mean of the k^{th} component and Σ_k is the covariance matrix of this same component.

Rmk: When using this model for clustering, we naturally associate the notion of mixture component to the groups that we are looking for.

we indeed recognize that :

- the model behind QDA is the Gaussian mixture model :

$$\begin{cases} p(Y=k) = \pi_k \\ p(X \mid Y=k) = \mathcal{N}(x; \mu_k, \Sigma_k) \end{cases}$$

marginalization over Y $\Rightarrow$

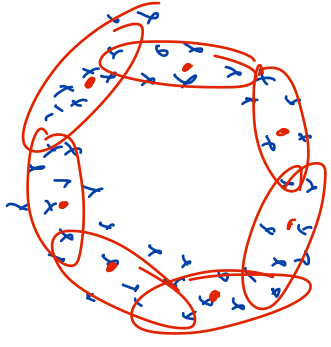
$$p(X) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x; \mu_k, \Sigma_k) .$$

- LDA :

$$p(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x; \mu_k, \Sigma)$$

↳ it is a restricted / constrained GMM.

Even though the GMM model is based on a single specific density, it can be used to fit data that are very complex and extremely far to be Gaussian:



GMM for clustering

From now, we are going to focus on the specific task of clustering with a specific Mixt. model which is the GMM.

Clustering with GMM is a two-step procedure:

1) fit the GMM model from data with the EM algorithm. (estimate model parameters)
2) deduce the clustering from the fitted model.

Remark: let's assume for the moment that K is known.