

TD 2 — Erreur de Bayes gaussienne, Bayes naïf, LDA et QDA

Séance 2

Christophe Ambroise

2026-09-29

Objectifs

Calculer l'erreur de Bayes d'un modèle gaussien à deux classes dans le plan, où tout est explicite, puis estimer par validation croisée l'erreur de Bayes naïf, de LDA et de QDA sur Iris. Confronter ce que la théorie garantit à ce que l'échantillon permet d'atteindre.

Le modèle de la partie théorique :

$$X \mid Y = 0 \sim \mathcal{N}_2(\mu_0, I), \quad X \mid Y = 1 \sim \mathcal{N}_2(\mu_1, I), \quad \mu_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mu_1 = \begin{pmatrix} 1 \\ \varepsilon \end{pmatrix}.$$

Les deux classes sont supposées **équiprobables** ($\pi_0 = \pi_1 = 1/2$), sauf mention contraire. La première variable sépare toujours les deux classes d'une unité ; la seconde n'apporte quelque chose que si $\varepsilon \neq 0$.

Partie théorique

Exercice 1 — Règle de Bayes et frontière linéaire

1. Écrire $\delta_k(x) = \log \pi_k + \log f_k(x)$ pour ce modèle, et simplifier.
2. Montrer que la règle de Bayes décide « classe 1 » lorsque $w^\top x > c$, et donner w et c explicitement en fonction de $\mu_0, \mu_1, \pi_0, \pi_1$.
3. Quelle est la direction de w ici ? Tracer la frontière pour $\varepsilon = 0$ puis $\varepsilon = 1$, sur le carré $[-3, 4]^2$.
4. Pourquoi la frontière est-elle une **droite** et non une conique ?

Exercice 2 — Erreur de Bayes et distance de Mahalanobis

1. Poser $U = w^\top X$. Donner la loi de U sous chaque classe. Que vaut $\text{Var}(U)$?
2. En déduire les deux erreurs conditionnelles, puis montrer que, sous $\pi_0 = \pi_1 = 1/2$:

$$R_B = \Phi\left(-\frac{\Delta}{2}\right), \quad \Delta^2 = (\mu_1 - \mu_0)^\top \Sigma^{-1}(\mu_1 - \mu_0).$$

3. Que vaut Δ ici ? Calculer R_B pour $\varepsilon \in \{0; 0,5; 1; 2; 4\}$.
4. Reprendre le calcul avec des proportions quelconques $\pi_0 \neq \pi_1$. Vérifier la formule obtenue sur le TD 1 : une seule dimension, $\mu_0 = -1, \mu_1 = 1,5, \sigma = 1, \pi_1 = 0,35$.

Exercice 3 — Ce qu’apporte la seconde variable

1. Un observateur ne dispose que de x_1 . Quel est Δ ? Quelle erreur de Bayes ?
2. Comparer à la question 3 de l’exercice 2 et remplir la colonne « gain ».
3. Démontrer que, dans ce modèle, ajouter une variable ne peut **jamais** augmenter l’erreur de Bayes. Le résultat vaut-il encore si la variable ajoutée est du bruit pur ($\varepsilon = 0$) ?
4. En pratique, ajouter une variable inutile dégrade pourtant les modèles. Où est la contradiction ?

Partie empirique

Exercice 4 — Bayes naïf, LDA et QDA sur Iris

Le squelette du code est fourni. Le compléter, puis répondre.

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import (LinearDiscriminantAnalysis,
                                           QuadraticDiscriminantAnalysis)
from sklearn.model_selection import StratifiedKFold, cross_val_score

iris = pd.read_csv("../data/iris.csv")
X, y = iris.drop(columns="species", iris["species"])
petale = ["petal length (cm)", "petal width (cm)"]

modeles = {"Bayes naïf": GaussianNB(),
           "LDA": LinearDiscriminantAnalysis(),
           "QDA": QuadraticDiscriminantAnalysis()}

plis = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)
for nom, modele in modeles.items():
    scores = cross_val_score(modele, X[petale], y, cv=plis)
    print(nom, 1 - scores.mean(), scores.std())
```

```
library(e1071)
library(MASS)

iris_df <- read.csv("../data/iris.csv")
iris_df$species <- factor(iris_df$species)
petale <- grep("petal", names(iris_df), value = TRUE)

# Plis stratifiés : chaque classe est répartie également entre les K plis.
plis_stratifies <- function(y, K = 10, graine = 0) {
  set.seed(graine)
  pli <- integer(length(y))
  for (cls in levels(y)) {
    idx <- sample(which(y == cls))
    pli[idx] <- rep_len(seq_len(K), length(idx))
  }
  pli
}

modeles <- list(
```

```

"Bayes naïf" = list(ajuste = function(d) naiveBayes(species ~ ., data = d),
                    predict = function(m, d) predict(m, d)),
"LDA"        = list(ajuste = function(d) lda(species ~ ., data = d),
                    predict = function(m, d) predict(m, d)$class),
"QDA"        = list(ajuste = function(d) qda(species ~ ., data = d),
                    predict = function(m, d) predict(m, d)$class))

erreur_vc <- function(donnees, modele, K = 10, graine = 0) {
  pli <- plis_stratifies(donnees$species, K, graine)
  mean(sapply(seq_len(K), function(k) {
    ajuste <- modele$ajuste(donnees[pli != k, ])
    mean(modele$predit(ajuste, donnees[pli == k, ]) != donnees$species[pli == k])
  })))
}

donnees <- iris_df[, c(petale, "species")]
sapply(modeles, function(m) erreur_vc(donnees, m))

```

1. Estimer l'erreur des trois modèles par validation croisée stratifiée à dix plis, d'abord sur les **deux** variables de pétale, puis sur les **quatre** variables.
2. Pourquoi stratifier les plis ? Que se passerait-il sans mélange préalable, le fichier étant trié par espèce ?
3. Le classement des trois modèles est-il le même dans les deux cas ? Proposer une explication géométrique.
4. Relancer avec `random_state=1`, puis 2. De combien l'estimation bouge-t-elle ? Peut-on départager deux modèles séparés de 0,7 point ?

Exercice 5 — Variables de bruit : théorie contre pratique

L'exercice 3 a établi que l'erreur de Bayes ne remonte jamais quand on ajoute une variable. Vérifions ce qu'il en est de l'erreur *estimée*.

```

import numpy as np

def ajoute_bruit(X, n_bruit, graine=0):
    """Ajoute n_bruit variables gaussiennes sans aucun lien avec la classe."""
    rng = np.random.default_rng(graine)
    return np.hstack([np.asarray(X), rng.normal(size=(len(X), n_bruit))])

```

```

ajoute_bruit <- function(donnees, n_bruit, graine = 0) {
  set.seed(graine)
  bruit <- matrix(rnorm(nrow(donnees) * n_bruit), nrow = nrow(donnees))
  colnames(bruit) <- paste0("bruit", seq_len(n_bruit))
  cbind(donnees, bruit)
}

```

1. Reprendre les quatre variables d'Iris et leur ajouter 5, puis 20, puis 50 variables de bruit. Réestimer l'erreur des trois modèles à chaque étape.
2. Quel modèle se dégrade le plus vite ? Le rapprocher du nombre de paramètres de covariance de chacun.
3. À partir de combien de variables de bruit QDA devient-il inutilisable ? Que dit scikit-learn au passage ?
4. Conclure sur le rapport entre l'exercice 3 et celui-ci.

Remarque finale

! À retenir

Bayes naïf, LDA et QDA sont trois contraintes sur la même matrice Σ . La théorie dit que la plus riche ne peut pas être moins bonne ; l'échantillon dit souvent le contraire. Entre les deux, il n'y a que le nombre de paramètres à estimer et la taille de n .

Les deux scripts exécutables correspondants sont `python/td02_classification.py` et `r/td02_classification.R`.