Chapitre 6 : Algorithmes de classification élémentaires

24 février 2026

Plan du chapitre

Au-delà de la restriction a priori

Jusqu'ici, pour éviter le surapprentissage, nous avons restreint *a priori* la classe des prédicteurs $\mathscr{H}$ et minimisé un risque empirique de comptage d'erreurs.

Il existe d'autres grandes approches en Machine Learning :

1. **Modèles génératifs (QDA, LDA, Naïf Bayes) :** On contraint la distribution des données plutôt que la frontière.
2. **Modèles basés sur les cas (k-NN) :** Pas de minimisation de risque, apprentissage par cœur et voisinage.
3. **Approches itératives (Perceptron) :** Algorithme heuristique géométrique.
4. **Fonctions de coût souples (Régression Logistique) :** On abandonne le comptage strict pour un risque convexe et dérivable.

L'approche de Bayes

Le prédicteur optimal de Bayes choisit la classe k maximisant la probabilité a posteriori :

$$\mathbb{P}(y = k \mid \mathbf{x}) = \frac{\pi_k p_k(\mathbf{x})}{\sum_j \pi_j p_j(\mathbf{x})}$$

- $\pi_k = \mathbb{P}(y = k)$: Probabilité a priori de la classe.
- $p_k(\mathbf{x}) = \mathbb{P}(\mathbf{x} \mid y = k)$: Distribution des données de la classe k.

Problème : Ces lois sont inconnues !

Solution Générative : Fixer une famille de lois *a priori* (ex : lois normales) et estimer leurs paramètres. Cela agit comme un régularisateur naturel.

Estimation des paramètres (Lois Normales)

Supposons que $p_k(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

À partir de la base d'apprentissage $\mathscr{S}$, on estime :

- **Fréquence des classes :** $\hat{\pi}_k = \frac{n_k}{n}$
- **Moyennes empiriques :** $\hat{\boldsymbol{\mu}}_k = \frac{1}{n_k} \sum_{y_i=k} \mathbf{x}_i$
- **Covariances empiriques :**

$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k - 1} \sum_{y_i=k} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T$$

QDA : Analyse Discriminante Quadratique

En injectant ces estimations dans la formule de Bayes et en passant au logarithme, la règle de décision $\arg\max_k \delta_k(\mathbf{x})$ génère des frontières définies par : $\delta_k(\mathbf{x}) = \delta_l(\mathbf{x})$.

Cette équation est quadratique :

$$c_{k,l} + \mathbf{l}_{k,l}^T \mathbf{x} + \mathbf{x}^T \mathbf{Q}_{k,l} \mathbf{x} = 0$$

avec le terme quadratique $\mathbf{Q}_{k,l} = \frac{1}{2}(\hat{\mathbf{\Sigma}}_l^{-1} - \hat{\mathbf{\Sigma}}_k^{-1})$.

Conclusion QDA

Si chaque classe possède sa propre matrice de covariance, les frontières de décision sont des **courbes quadratiques** (ellipses, paraboles, hyperboles).

LDA et Bayes Naïf

LDA (Analyse Discriminante Linéaire) :
On contraint le modèle en supposant que **toutes les classes ont la même covariance** ($\mathbf{\Sigma}_k = \mathbf{\Sigma}$).

- Le terme quadratique $\mathbf{Q}_{k,l}$ s'annule !
- L'équation de la frontière devient purement linéaire : $c_{k,l} + \mathbf{l}_{k,l}^T \mathbf{x} = 0$.
- *Avantage :* Moins de paramètres à estimer (plus robuste en grande dimension face à QDA).

Bayes Naïf :
On va encore plus loin : on suppose que la covariance est un multiple de la matrice identité (les caractéristiques sont supposées indépendantes conditionnellement à la classe).

La méthode des k-NN (Nearest Neighbors)

Algorithme basé sur les instances (pas de phase d'optimisation de paramètres). La régularisation se fait en contraignant des points géométriquement proches à avoir des étiquettes proches.

Pour une nouvelle donnée $\mathbf{x}$:

1. Calculer la distance (ex : euclidienne) entre $\mathbf{x}$ et tous les points de $\mathscr{S}$.
2. Identifier l'ensemble V_k des k voisins les plus proches.
3. **Classification :** Assigner l'étiquette majoritaire dans V_k.
4. **Régression :** Calculer la moyenne des étiquettes de V_k.

Performances et Limites du k-NN

Sous l'hypothèse que la probabilité d'étiquette $\eta(\mathbf{x})$ est continue (Lipschitzienne), le k-NN converge vers le risque de Bayes lorsque $n \to \infty$.

La malédiction de la dimension frappe à nouveau !

Pour garantir une erreur ϵ, le nombre de données nécessaires explose avec la dimension d :

$$n \geq \left(\frac{4c\sqrt{d}}{\epsilon} \right)^{d+1}$$

Rappel : En grande dimension, tout le monde est éloigné de tout le monde. La notion même de "voisin" perd son sens.

Le Perceptron (Rosenblatt, 1958)

Le premier algorithme de classification binaire linéaire ($\mathscr{Y} = \{-1; +1\}$). Il cherche itérativement un hyperplan séparateur $\text{sign}(\langle \mathbf{w}_t, \mathbf{x} \rangle) = y$.

Algorithme

1. Initialisation : $\mathbf{w}_1 = \mathbf{0}$.
2. Pour chaque exemple $\mathbf{x}_i$ mal classé (i.e., $y_i \langle \mathbf{w}_t, \mathbf{x}_i \rangle < 0$) :

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_i \mathbf{x}_i$$

3. Répéter jusqu'à ce que plus aucun exemple ne soit mal classé.

Intuition et Théorème de Convergence

Pourquoi ça marche ?

Lorsqu'un point est mal classé, l'ajout vectoriel $y_i \mathbf{x}_i$ corrige le poids $\mathbf{w}$. À l'itération suivante, la prédiction $y_i \langle \mathbf{w}_{t+1}, \mathbf{x}_i \rangle$ aura augmenté strictement de $\|\mathbf{x}_i\|^2$, allant dans le bon sens.

Théorème de convergence (Marge)

Si les données sont linéairement séparables, limitées dans une sphère de rayon R, et qu'il existe un hyperplan solution de norme minimale B, le Perceptron s'arrêtera en un nombre fini d'itérations T tel que :

$$T \leq (RB)^2$$

Limite : Ne fonctionne pas si les données ne sont pas parfaitement séparables. La solution finale dépend fortement de l'ordre de présentation des données.

De la frontière dure à la frontière souple

L'optimisation avec la fonction $\mathrm{sign}()$ est difficile (combinatoire, gradient nul). La **Régression Logistique** remplace la fonction signe par une fonction lisse : la Sigmoïde.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- Pour $x \to +\infty$, $\sigma(x) \to 1$
- Pour $x \to -\infty$, $\sigma(x) \to 0$
- $\sigma(0) = 1/2$

Le modèle prédit alors une **probabilité** : $h_{\mathbf{w}}(\mathbf{x}) = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle)$.

La Fonction de Coût Logistique

Si $y_i = +1$, on veut que $\sigma(\langle \mathbf{w}, \mathbf{x}_i \rangle)$ soit proche de 1.
Si $y_i = -1$, on veut que $\sigma(\langle \mathbf{w}, \mathbf{x}_i \rangle)$ soit proche de 0.

Mathématiquement, cela revient à demander que la quantité $(1 + e^{-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle})^{-1}$ soit proche de 1.

Minimisation du Risque Logistique (Log-Loss)

En passant au logarithme (pour des raisons de convexité et de vraisemblance), on obtient la fonction de coût à minimiser :

$$\mathbf{w}_\star \in \arg\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \log \left[1 + e^{-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle} \right]$$

Avantage majeur : Cette fonction est strictement convexe et dérivable partout ! L'optimisation par descente de gradient est donc garantie de trouver l'optimum global.