

IFT6390-A-H23 - Fondements de l'apprentissage machine

Pierre-Luc Bacon

Références

Probabilistic Machine Learning: An Introduction par Kevin P. Murphy, disponible en ligne gratuitement et légalement. Au menu: retour sur les notions présentées au premier cours.
Dans le livre PML:

Chapitre 1

Chapitre 16

Apprentissage supervisé

Notre jeu de données $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ est un ensemble de paires d'entrées $x \in \mathcal{X}$ et de sorties $y \in \mathcal{Y}$. En classification (à m classes) nous voulons trouver une fonction $\hat{f} : \mathcal{X} \rightarrow \{0, \dots, m\}$ qui approxime bien les données. C'est-à-dire: que nous puissions faire de bonnes prédictions au-delà des données déjà observées: c'est ce que nous appelons la "généralisation".

Classification: exemple du jeux de données IRIS



(a) Setosa



(b) Versicolor



(c) Virginica

Données tabulaires

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
:	:	:	:

Table: Quelques rangées du **jeu d'entraînement**. Les colonnes représentent des **caractéristiques/traits** (features) choisies à priori pour leur valeur prédictive. Parfois, nous transformons aussi ces caractéristiques dans un autre espace, plus riche et expressif, que nous appelons l'**espace de redescription (feature space)**.

k plus proches voisins

Idée: pour classifier un exemple x (c'est-à-dire pour prédire son étiquette), il suffit de trouver les k ses proches voisins dans la géométrie donnée et en dériver une distribution empirique, localement:

$$p(y = c|x, \mathcal{D}) \doteq \frac{1}{k} \sum_{i \in \mathcal{N}(x)} \mathbb{1}_{y_i=c}$$

Le voisinage $\mathcal{N}(x)$ est déterminé par le choix de distance. Exemple: **distance de Mahalanobis** où M est une matrice définie positive. Si $M = I$, nous retrouvons la distance euclidienne.

$$d(x_i, x_j) \doteq \sqrt{(x_i - x_j)^\top M (x_i - x_j)}$$

k plus proches voisins: diagramme de Voronoï

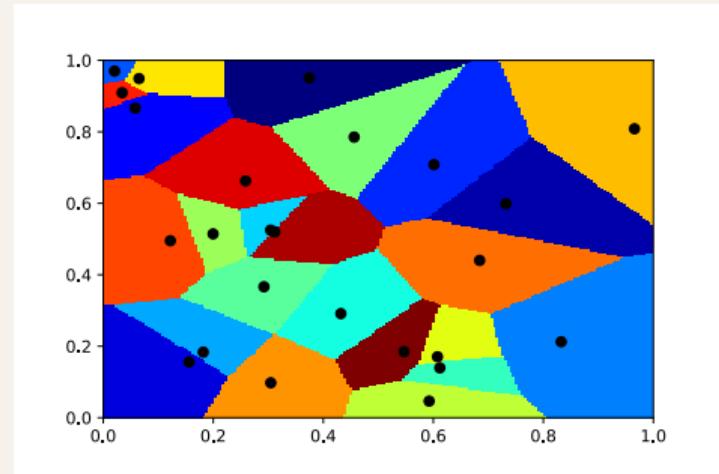
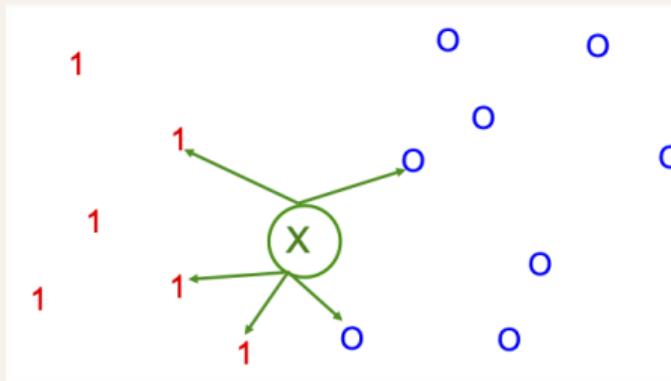
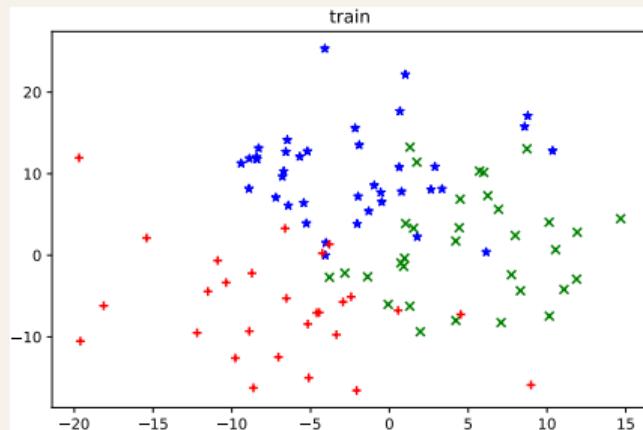
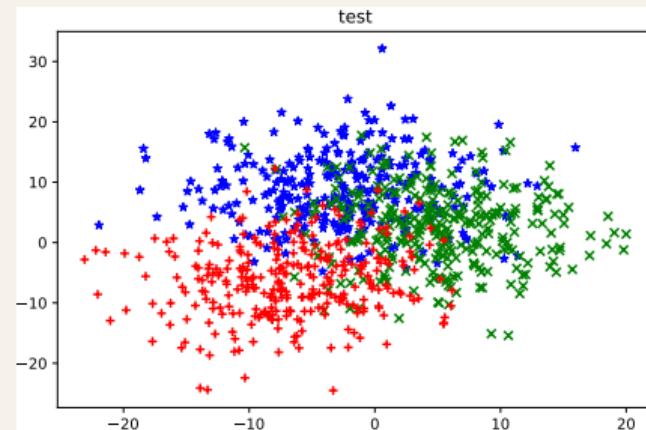


Figure: Exemple en 2D pour $k = 5$. À gauche: La classe majoritaire est 1 et nous avons que $p(y = 1|x) = 3/5$. À droite: $k = 1$ induit un **pavage** (tesselation) de l'espace appelé "diagramme de Voronoï". Chaque cellule contient tous les points les plus près du "germe" donné. L'erreur d'entraînement est alors 0.

k plus proches voisins: autre exemple



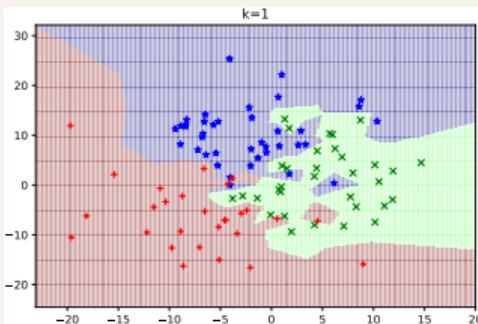
(a) Données d'entraînement $\mathcal{D}_{\text{train}}$



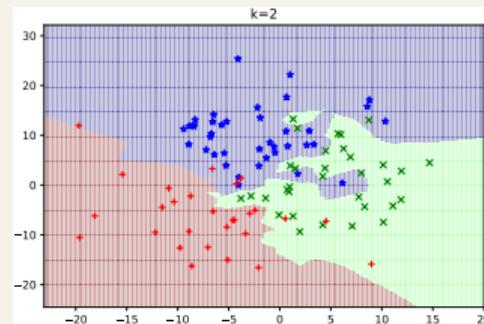
(b) Données de test $\mathcal{D}_{\text{test}}$

Figure: Données synthétiques en 2D. Un problème de classification à 3 classes

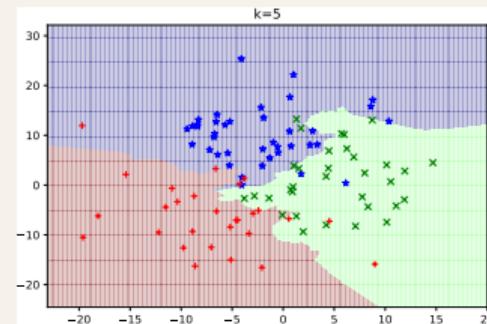
k plus proches voisins: effet du paramètre de voisinage



(a) $k = 1$



(b) $k = 2$



(c) $k = 5$

Les valeurs de k plus grandes ont un effet de lissage croissant. La **frontière de décision** est plus discontinue pour de petites valeurs de k .

Erreurs d'entraînement et de test

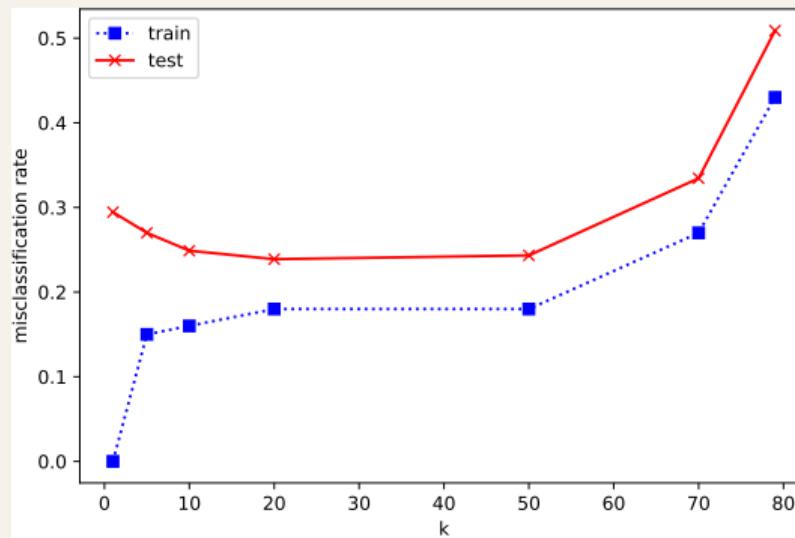


Figure: L'erreur d'entraînement (en utilisant $\mathcal{D}_{\text{train}}$) peut être réduite à zéro avec $k = 1$. Toutefois, les capacités de généralisation du modèle risquent d'être mauvaises. Le choix de valeur pour k ainsi que du type métrique introduit le "dilemme biais-variance". $k = 20$ semble être la valeur idéale ici

Fléau de la dimensionnalité

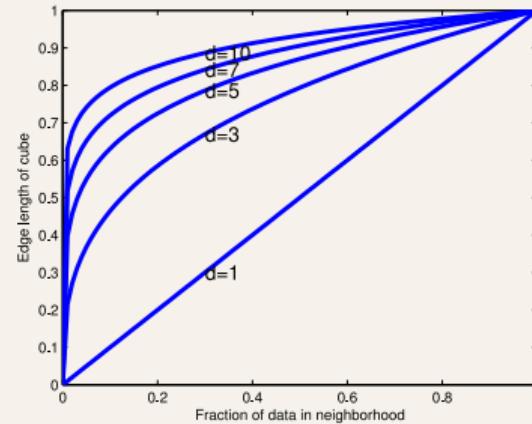
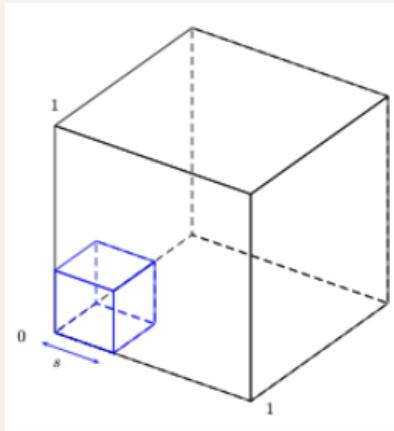


Figure: Le volume croît de manière exponentielle en fonction du nombre de dimensions. Les voisins deviennent alors plus éloignés les uns des autres. Cette grande distance mène à un manque de localité, qui est mauvais pour la généralisation. Si les points sont distribués uniformément dans l'espace, il nous faudra un hypercube dont le côté mesure $p^{1/d}$ pour contenir une fraction p de points. L'espace devient "peu peuplé" (sparsely populated).

Noyau de densité

Un noyau de densité est une fonction $\mathcal{K} : \mathcal{R} \rightarrow \mathcal{R}_+$ tel que:

1. $\int \mathcal{K}(x)dx = 1$
2. $\mathcal{K}(-x) = \mathcal{K}(x)$

de sorte que $\int x\mathcal{K}(x - x_i) = x_i$. Exemples:

Noyau Boxcar: $\mathcal{K}(x) \doteq (1/2)\mathbb{1}(|x| \leq 1)$

Noyau gaussien: $\mathcal{K}(x) \doteq \frac{1}{(2\pi)^{1/2}} \exp^{-x^2/2}$

Si $x \in \mathbb{R}^d$ est un vecteur (plutôt qu'un scalaire), nous pouvons utiliser un **noyau de fonction de base radiale** (radial basis function, RBF), qui pour le cas gaussien, est:

$$\mathcal{K}_\lambda(x) \doteq \frac{1}{\lambda^d (2\pi)^{d/2}} \prod_{i=1}^d \exp\left(-\frac{1}{2h^2} x_i^2\right)$$

Estimation de densité

Une autre approche non paramétrique, cette fois pour l'estimation de densité. Le **paramètre de largeur** (ici identifié par la lettre λ) contrôle le degré de lissage (et donc de généralisation) du modèle. L'estimateur de densité à base de **fenêtres de Parzen** associe un noyau pour chaque exemple de l'ensemble d'entraînement:

$$p(x|\mathcal{D}) \doteq \frac{1}{N} \sum_{i=1}^N K_\lambda(x - x_i)$$

Fenêtres de Parzen

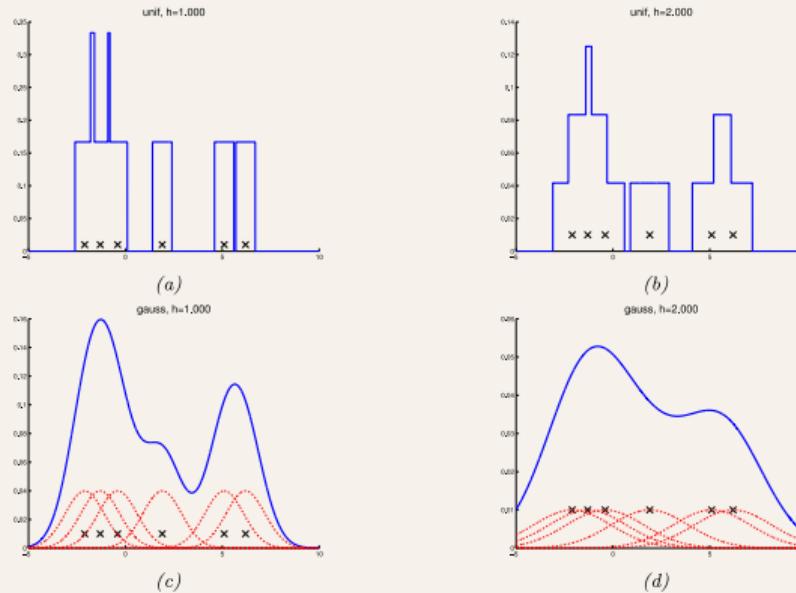


Figure: Rangée du haut: noyau uniforme. Rangée du bas: noyau gaussien

k-ppv: un cas spécial des fenêtres de Parzen

Nous pouvons montrer que la méthode des k plus proches voisins pour la classification peut s'exprimer dans le langage des fenêtres de Parzen. Nous voulons ici modéliser la distribution conditionnelle sur l'étiquette étant donnée une entrée:

$$p(x|y=c, \mathcal{D}) = \frac{N_c(x)}{N_c V(x)},$$

Gaussien où $\mathcal{N}_c(x)$ est le nombre d'exemples de class c autour de x dans un volume $V(x)$ (un "ballon") contenant k exemples (**estimateur de densité à ballon**) et \mathcal{N}_c est le nombre d'exemples de classe c dans tout l'ensemble d'entraînement. En utilisant le théorème de Bayes et $p(y=c) = N_c/N$, nous avons:

$$p(y=c|x, \mathcal{D}) = \frac{\frac{N_c(x)}{N_c V(x)} \frac{N_c}{N}}{\sum_{c'} \frac{N_{c'}(x)}{N_{c'} V(x)} \frac{N_{c'}}{N}} = \frac{N_c(x)}{\sum_{c'} N_{c'}(x)} = \frac{N_c(x)}{K} = \frac{1}{K} \sum_{i \in \mathcal{N}(x)} \mathbb{1}(y_i = c)$$

Régression par noyau

Dans le problème de régression, nous voulons estimer l'espérance de la cible associée à chaque exemple, c'est-à-dire:

$$\mathbb{E}[Y|x, \mathcal{D}] = \int yp(y|x, \mathcal{D})dy = \frac{\int yp(x, y|\mathcal{D})dy}{\int p(x, y|\mathcal{D})dy}$$

Nous pouvons utiliser l'idée d'estimation de densité à nouveau, mais cette fois pour des fins de régression en modélisant $p(x, y|\mathcal{D})$ par:

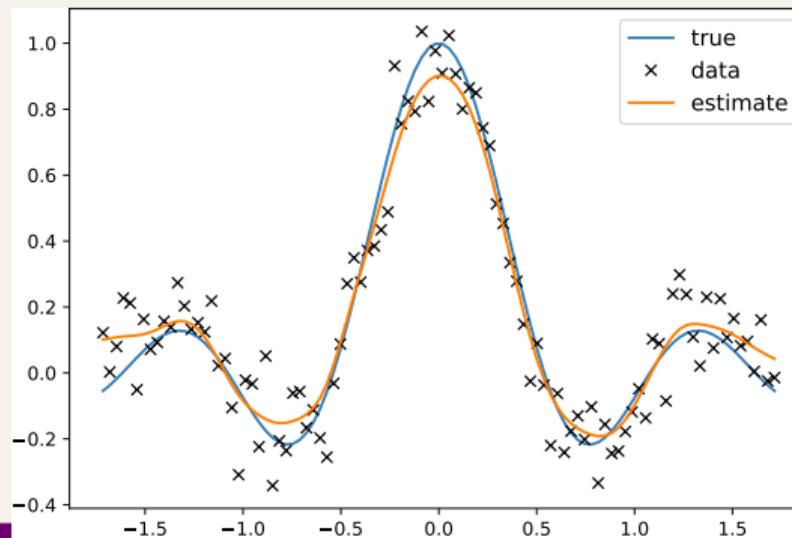
$$p(y, x|\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathcal{K}_\lambda(x - x_i) \mathcal{K}_\lambda(y - y_i) .$$

Il en découle que:

$$\mathbb{E}[Y|x, \mathcal{D}] = \sum_{i=1}^N y_i w_i(x), \quad w_i(x) \doteq \frac{\mathcal{K}(x - x_i)}{\sum_{i=1}^N \mathcal{K}_\lambda(x - x_i)} .$$

Méthode de Nadaraya-Watson

La méthode présentée ci-haut est celle de **Nadaraya-Watson**. Nous pouvons voir qu'un prédition est faite en calculant une somme pondérée sur l'ensemble complet d'entraînement. La pondération dépend de chaque exemple et est calculée à l'aide d'un noyau.



Risque d'un modèle

La performance d'un modèle est évaluée d'après une **fonction de perte** (loss function) donnée. Le **risque** associé à un modèle f est l'espérance de cette fonction de perte sur la distribution intrinsèque des données.

$$\mathcal{R}(f) \doteq \mathbb{E} [l(Y, f(X))] .$$

R est une fonctionnelle et l est une fonction de perte. Le risque empirique (empirical risk) est une approximation du "vrai" risque calculée d'après la perte moyenne sur un échantillon:

$$\hat{\mathcal{R}}(f, \mathcal{D}) \doteq \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} l(y_i, f(x_i))$$

Minimisation du risque

Vapnik (1992) propose un cadre théorique pour formaliser l'idée de l'apprentissage. Sous cette perspective (celle que nous avons déjà adoptée au cours des dernières heures), l'apprentissage s'apparente à un problème d'estimation d'une fonction. Cette approximation est celle qui minimise le risque, qui est définie comme étant le risque espéré:

$$\text{minimiser}_{f \in \mathcal{F}} R(f)$$

où \mathcal{F} est une famille de fonctions. Puisque le risque est une espérance sur un espace continu $\int l(y, f(x))p(x, y)dxdy$, nous faisons face à un problème d'intégration numérique coûteux. De plus, la distribution jointe $p(x, y)$ n'est pas connue et nous n'y avons accès qu'indirectement via un échantillon. Que faire?

Minimisation du risque empirique

Le "principe d'induction" de la minimisation du risque empirique soutient que le minimum du risque empirique s'approche de celui du "vrai" risque: une question de **cohérence statistique** (consistency). Formellement, si f^* est un minimiseur du risque R , et que \hat{f}_N^* est un minimiseur du risque empirique sur N exemples:

Est-ce que $R(\hat{f}^N)$ peut atteindre $R(f^*)$ lorsque $N \rightarrow \infty$?

À quel "vitesse" (quantité de données) ?

Nous répondons à ces questions en établissant que le risque empirique \hat{R} **converge uniformément** vers le risque \hat{R} :

$$\mathbb{P} \left(\sup_{f \in \mathcal{F}} |R(f) - \hat{R}_N(f)| > \epsilon \right) \rightarrow 0 \text{ lorsque } N \rightarrow \infty$$

Maximum de vraisemblance

Un choix possible de fonction de perte dans un contexte d'apprentissage de modèles probabilistes est celui du logarithme négatif de la probabilité conditionnelle:

$$l(y, f(x)) = -\log p(y|f(x))$$

Le risque empirique correspondant à cette fonction de perte nous donne:

$$\hat{R}(f, N) = -\frac{1}{N} \sum_{i=1}^N \log p(y_i|f(x_i)) ,$$

et le minimiseur de \hat{R} ci-haut coïncide avec ce qui est appelé **l'estimateur de vraisemblance**.