

IFT3390/6390


Fondements de l'apprentissage machine

<http://www.iro.umontreal.ca/~vincentp/ift3390>

Quatrième cours:

**Méthodes à base de voisinage: k-NN, Parzen
pour classification, régression, estimation de densité**

Professeur: Pascal Vincent

LISA  Laboratoire d'Informatique des Systèmes d'Apprentissage

Au programme aujourd'hui

- Rappel des types de problème en apprentissage
- Méthodes à base de voisinage: k-NN et Parzen pour classification binaire, régression, estimation de densité.
- Rappels de bases de probabilités.
- Classifieur de Bayes.
- Etapes de conception d'un algorithme

Les types de problèmes en apprentissage

	Classification	Régression	Estimation de densité
Signification de la cible y	indique une classe parmi c classes.	une valeur réelle à prédire.	pas de cible y !
Domaine de y	$y \in \{-1, 1\}$ ou $y \in \{1, \dots, c\}$ ou $y \in \{0, 1, \dots, c-1\}$	$y \in \mathbb{R}$	pas de cible y !
Ce que $f(x)$ vise à prédire	la classe de x (la classe la plus probablement associée à x)	la valeur espérée de y (le y "moyen") correspondant à x . $E[Y X=x]$	la densité $p(x)$ (l'observation x est-elle fort ou peu probable?)
Fonction de perte (ou coût) que l'on veut habituellement minimiser.	l'erreur de classification: $L((x, y), f) = I_{\{f(x) \neq y\}}$	l'erreur quadratique: $L((x, y), f) = (f(x) - y)^2$	la log-vraisemblance négative: $L(x, f) = -\log f(x)$

Méthodes à base de voisinage

- Une idée simple: faire voter les voisins du point de test.
- Ex. k-NN classification multiclass: "parmi mes k plus proches voisins, quelle classe est majoritaire?"
- Tout comme les méthodes de type histogramme (quadrillage de l'espace), les méthodes à base de voisinage sont des méthodes dites "non-paramétriques".

K-NN (k nearest neighbors)

K-PPV (k plus proches voisins)

Pour la classification binaire (avec $Y_i \in \{-1, 1\}$)

$$f(x) = \text{sign} \left(\frac{1}{k} \sum_{\{i \in 1 \dots n \mid X_i \in V(x)\}} Y_i \right) \quad \begin{array}{l} \text{signe de} \\ \text{la moyenne des valeurs cibles des} \\ \text{k voisins les plus proches de } x \end{array}$$

$$f(x) = \text{sign} \left(\frac{1}{k} \sum_{i=1}^n I_{\{X_i \in V(x)\}} Y_i \right) \quad \begin{array}{l} V(x) = \text{ensemble des } k \text{ plus} \\ \text{proches voisins de } x \text{ dans} \\ \text{l'ensemble d'apprentissage} \end{array}$$

$$f(x) = \text{sign} \left(\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i Y_i \right) \quad \text{avec } w_i = I_{\{X_i \in V(x)\}}$$

signe de la moyenne **pondérée** des valeurs cibles de **tous** les points d'entraînement, pondérées par un poids indiquant si le point d'entraînement est voisin de x.

K-NN (k nearest neighbors)

K-PPV (k plus proches voisins)

Pour la régression (avec $Y_i \in \mathbb{R}$)

$$f(x) = \cancel{\text{sign}} \left(\frac{1}{k} \sum_{\{i \in 1 \dots n \mid X_i \in V(x)\}} Y_i \right) \quad \begin{array}{l} \text{la moyenne des valeurs cibles des} \\ \text{k voisins les plus proches de } x \end{array}$$

$$f(x) = \cancel{\text{sign}} \left(\frac{1}{k} \sum_{i=1}^n I_{\{X_i \in V(x)\}} Y_i \right) \quad \begin{array}{l} V(x) = \text{ensemble des } k \text{ plus} \\ \text{proches voisins de } x \text{ dans} \\ \text{l'ensemble d'apprentissage} \end{array}$$

$$f(x) = \cancel{\text{sign}} \left(\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i Y_i \right) \quad \text{avec } w_i = I_{\{X_i \in V(x)\}}$$

la moyenne **pondérée** des valeurs cibles de **tous** les points d'entraînement, pondérées par un poids indiquant si le point d'entraînement est voisin de x.

Fenêtres de Parzen

à voisinage dur

Pour la classification binaire (avec $Y_i \in \{-1, 1\}$)

$V(x) =$ ensemble des points de l'ensemble d'apprentissage situés à moins d'une distance h de x .

signe de la moyenne des valeurs cibles des voisins de x situés à distance $\leq h$

$$f(x) = \text{sign} \left(\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i Y_i \right) \quad \text{avec } \begin{array}{l} w_i = I_{\{X_i \in V(x)\}} \\ w_i = I_{\{d(X_i, x) < h\}} \\ w_i = I_{\{\frac{d(X_i, x)}{h} < 1\}} \end{array}$$

signe de la moyenne **pondérée** des valeurs cibles de **tous** les points d'entraînement, pondérées par un poids indiquant si le point d'entraînement est voisin de x.

Fenêtres de Parzen

à voisinage dur

Pour la régression (avec $Y_i \in \mathbb{R}$)

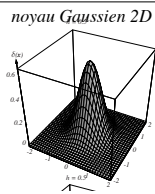
$V(x) =$ ensemble des points de l'ensemble d'apprentissage situés à moins d'une distance h de x .

la moyenne des valeurs cibles des voisins de x situés à distance $\leq h$

$$f(x) = \left(\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i Y_i \right) \quad \text{avec } \begin{array}{l} w_i = I_{\{X_i \in V(x)\}} \\ w_i = I_{\{d(X_i, x) < h\}} \\ w_i = I_{\{\frac{d(X_i, x)}{h} < 1\}} \end{array}$$

la moyenne **pondérée** des valeurs cibles de **tous** les points d'entraînement, pondérées par un poids indiquant si le point d'entraînement est voisin de x.

Fenêtres de Parzen à voisinage mou (soft)



Pour la régression (avec $Y_i \in \mathbb{R}$)

$$f(x) = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i Y_i \quad \text{avec} \quad w_i = K(X_i, x)$$

la moyenne **pondérée** des valeurs cibles de **tous** les points d'entraînement, pondérées par un poids indiquant à quel "degré" le point est voisin de x .

K est un noyau (Kernel)

notez que $w_i = I_{\{\frac{d(X_i, x)}{h} < 1\}}$ correspond à un K particulier (un noyau "dur")

Comme noyau "mou" on choisit souvent un noyau Gaussien (correspond à une densité Normale)

$$K(X_i, x) = \mathcal{N}_{x, \sigma^2}(X_i) = \mathcal{N}_{X_i, \sigma^2}(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} e^{-\frac{1}{2} \frac{d(X_i, x)^2}{\sigma^2}}$$

Fenêtres de Parzen (en résumé)

Pour la régression ($Y_i \in \mathbb{R}$) :

$$f(x) = \frac{1}{\sum_{i=1}^n K(X_i, x)} \sum_{i=1}^n K(X_i, x) Y_i$$

Pour la classification binaire ($Y_i \in \{-1, 1\}$) :

$$f(x) = \text{sign} \left(\frac{1}{\sum_{i=1}^n K(X_i, x)} \sum_{i=1}^n K(X_i, x) Y_i \right)$$

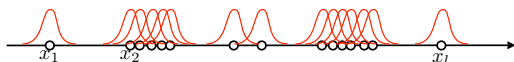
Pour l'estimation de densité :

$$f(x) = \hat{p}(x) = \frac{1}{n} \sum_{i=1}^n K(X_i; x)$$

à condition que

$K_{X_i}(x) = K(X_i; x)$ soit bien une fonction de densité de probabilité.
Ex: une Gaussienne centrée en X_i

Fenêtres de Parzen en 1D



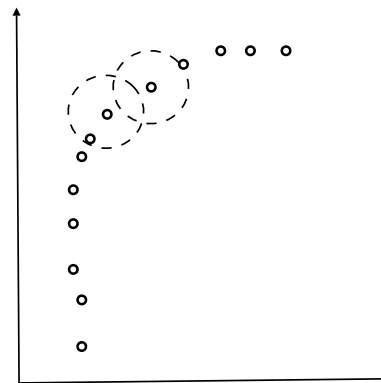
Gaussienne en dimension 1 :

$$\mathcal{N}_{\mu, \sigma}(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Estimateur de densité de Parzen:

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}_{X_i, \sigma}(x)$$

Fenêtres de Parzen en 2D



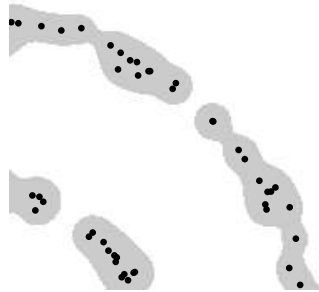
Gaussienne isotropique en dimension d :

$$\mathcal{N}_{\mu, \sigma}(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \sigma^d} e^{-\frac{1}{2} \frac{\|x-\mu\|^2}{\sigma^2}}$$

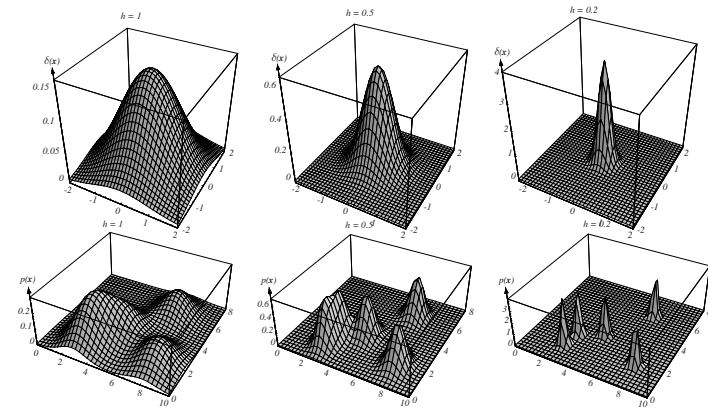
Exemple d'estimation de densité 2D

PARZEN WINDOWS

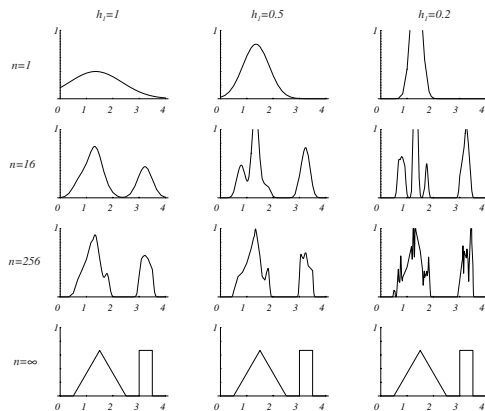
La zone grisée est la région où la densité estimée par l'estimateur de Parzen est supérieure à une certaine valeur seuil.



• L'effet de la largeur de fenêtre h_n



- Exemple: $p(x) \sim \text{triangle} + \text{uniform}$, $\phi(u) = \frac{1}{\sqrt{2\pi}}e^{-u^2/2}$



Etapes de conception d'un algorithme d'apprentissage

- Compréhension intuitive de l'algorithme. Savoir l'expliquer en français!
- Formalisation mathématique de l'algorithme.
- Ecriture de l'algo sous forme de pseudo-code.
- Implémentation dans un langage/environnement de programmation.
- Entraînement/test de l'algo sur des problèmes simples en faible dimension, où on peut vérifier graphiquement si ça fait bien ce qu'on veut.
- Evaluation de performance sur des problèmes réels, et comparaison avec d'autres algorithmes concurrents.