IFT6390 - Devoir 1

Présenté par

Nicolas Daoust

Pierre-Olivier Brosseau

Le mardi 4 octobre 2011

# 1 – Fléau de la dimensionnalité et intuition géométrique en haute dimension

## Question 1

Le volume d’un hyper-cube de dimension *d* et de côté *a* est : *V* = *ad*

## Question 2

Supposons que l’hyper-cube de dimension d se situe à l’origine et est de côté a.

Si une fonction f est la densité de probabilité d'une variable aléatoire à valeur dans Rd, cette fonction vérifie les propriétés suivantes :

1. f est intégrable sur R*d.*



1. est positive ou nulle sur R*d*.



Nous proposons la densité p(x) = 1/ad, qui respecte ces propriétés.

## Question 3

Soit V1 = ad le volume de l’hyper-cube.

V2 = (0,96a)d le volume de l’hyper-cube intérieur.

V3 = V1 – V2 = ad - (0,96a)d le volume de la bordure.

Comme nous avons une distribution uniforme, la probabilité qu’il tombe dans l’hyper-cube plus petit est le volume de l’hyper-cube plus petit divisé par celui du plus grand. Nous avons un calcul similaire pour la probabilité que x soit dans la zone de bordure.

P(x ε hyper-cube plus petit) = (0,96a)d / ad = 0,96d

P(x ε zone de bordure) = ad – (0,96a)d / ad = 1-(0,96)d

## Question 4

Pour d=1, on a : a1 – (0,96a)1 / a1 = 0,04a/a = 0,04 = 4%

Pour d=2, on a : a2 – (0,96a)2 / a2 = 0,0784a2/a2 = 0,0784 = 7,84%

Pour d=3, on a : 11,53%

Pour d=5, on a : 18,46%

Pour d=10, on a : 33,52%

Pour d=100, on a : 98,31%

Pour d=1000, on a : 99,9999999999999998% (presque 100%)

## Question 5

Les petites différences en faible dimension peuvent augmenter de manière exponentielle pour devenir des différences titanesques à haute dimension.

# 2 – Petit exercice de probabilités

## Question 1

Soit A1 : « la patiente est atteinte du cancer »

Soit A2 : « la patiente n’est pas atteinte du cancer »

Soit B : « le test est positif »

P (B|A1) = 0.80

P (B|A2) = 0.096

P (A1) = 0.01

P (A2) = 0.99

Selon la règle de Bayes on a:

P(A|B) = \_\_\_\_\_\_\_P(A1) x P(B|A1)\_\_\_\_\_\_\_\_ = \_\_\_\_\_\_0.01 x 0.80\_\_\_\_\_\_

P(A1) x P(B|A1) + P(A2) x P(B|A2) 0.01 x 0.80 + 0.99 x 0.096

≈ 0,0776 soit 7,76%

Ici la réponse intuitive est erronée, car même si le test a 80% de chances d’être positif lorsqu’une femme a le cancer, la probabilité qu’une femme de cette tranche d’âge ait le cancer lorsque le test est positif est seulement de **7,76%.** Ceci est dû au fait qu’on a un très petit nombre de femmes qui ont vraiment le cancer (une sur cent) et que pratiquement tous les tests positifs désignent des porteurs qui n’ont pas le cancer. Ce biais cognitif est aussi connu sous le nom de l’oubli de la fréquence de base.

# 3 – Fenêtres de Parzen (régression, classification et estimation de densité) et classifieur de Bayes, Compréhension théorique.

## Question 1

La fonction de prédiction en un point x, notée f(x) :



Avec un noyau Gaussien :



## Question 2

Soit xpred le point x choisi comme point de test.

sig2 = 1 ## Variance

def K (xi, x, sig2, d) :

### Loi normale en dimension d

Math.e\*\*[-(x-xi)\*\*2/(2\*sig2)] / (2\*Math.Pi\*sig2)\*\*(d/2)

def fonctionDePrediction (x, y, xpred)

num = zeros(m) ## Somme pondérée des observations

den = 0 ## Pondération totale

for i in range(n) :

Ktemp = K(x[i], xpred, sig2, len(x))

num += Ktemp\*y[i]

den += Ktemp

return Num/Den

## Question 3

a) La complexité algorithmique d’apprentissage de l’algorithme est de O(n\*(d+m)).

b) La consommation mémoire de l’algorithme est de O(n\*(d+m)), soit l’ensemble des données d’entraînement.

c) La complexité de prédiction de l’algorithme est de O(n\*(d+m)).

Pour une base dimension l’histogramme est beaucoup moins complexe.

## Question 4

def g (xpred) :

v = zeros(n, m)

for i in range(n) :

v[i, y[i]-1] = 1

return fonctionDePrediction (x, v, xpred)

def f (xpred) :

return argmax(g(xpred))+1

## Question 5

Le poids de chaque observation y serait multiplié par β, mais le poids total serait lui aussi multiplié par β et les constantes s’annuleraient donc lors du calcul du vecteur dans la fonctionDePrediction.

## Question 6 ------

L’estimateur de densité:



## Question 7

Il est évident que si tous les noyaux K sont positifs et intégrables, leur sommation le sera également. Prouvons que s’intègre à 1.



= = = 1



## Question 8

Comme K’ n’est pas une densité, notre ne sera pas non plus une densité. Par contre, il est facile de normaliser par la suite en divisant par β.



## Question 9

Premièrement, on sépare l’ensemble d’entraînement en m sous-ensembles différents contenant chacun tous les points d’une même classe. Ensuite, on utilise l’estimateur de densité de Parzen sur chaque sous-ensemble, puis on normalise chacune de ces densités selon le théorème de Bayes en utilisant leurs probabilités à priori. Finalement, on choisit la classe la plus probable.



## Question 10

Fonction de décision :

1. L’estimateur de densité de Parzen est :



1. On approxime :



1. La fonction de décision est :



Équivalence :

Ce classifieur de Bayes regroupe les densités de Parzen de toutes les observations de chaque classe, puis les pondère selon leur représentation.

Dans l’algorithme de classification multiclasse de Parzen, le codage « one-hot » cumule séparément les densités de chaque classe dans un vecteur et les pondère à la fois.

Les deux méthodes sont donc équivalentes.

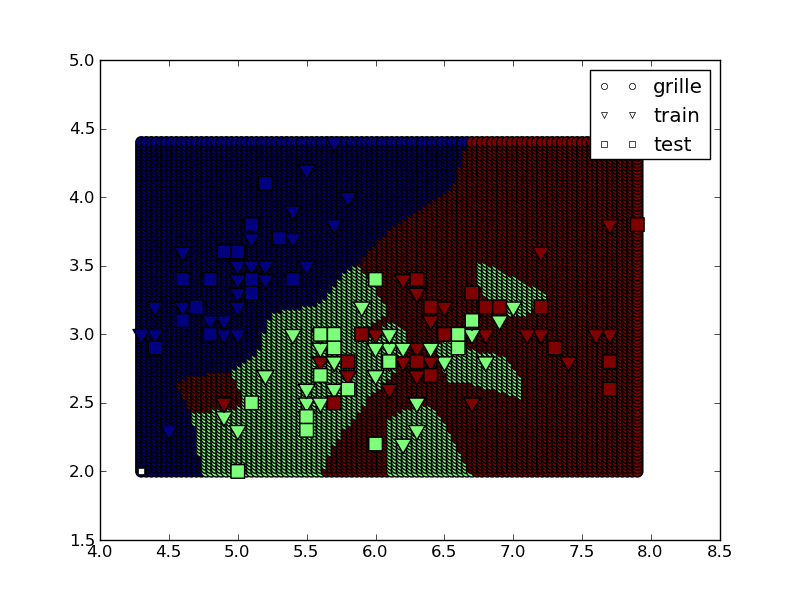
# 4 – Classifieur multi-classe de Parzen: implémentation en Python et expérimentation

## Question 1, 2 et 3

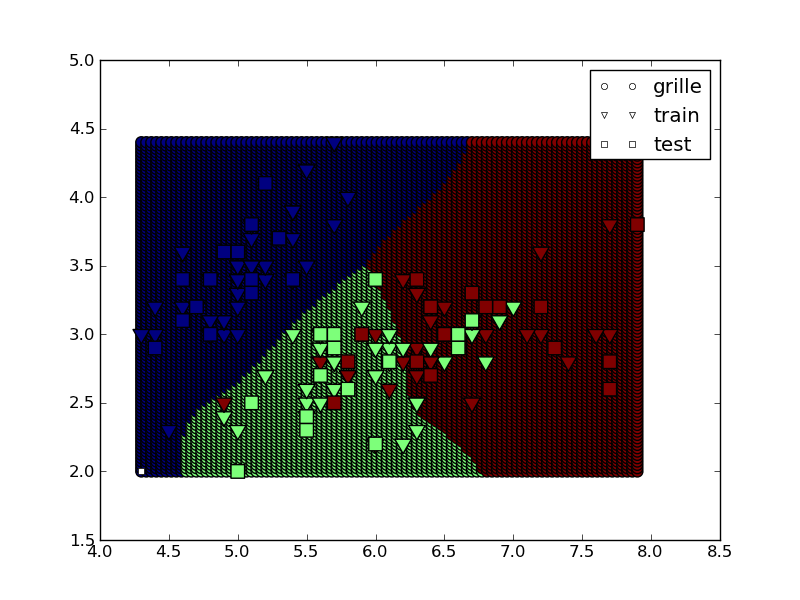
Voir commentaires dans le fichier parzen.py

## Question 4

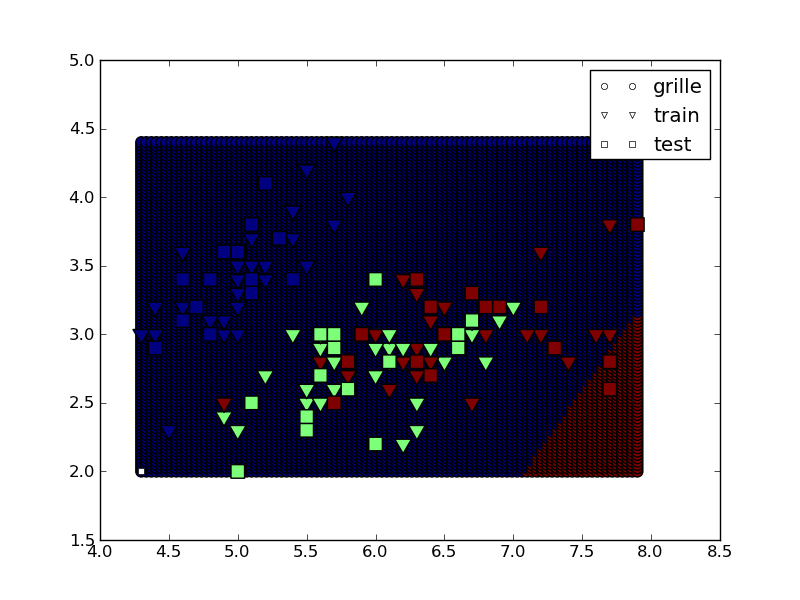
Trop petit : σ = 0.1



Satisfaisant : σ = 0.3



Trop grand : σ = 5.0



## Question 5 ------ (ces courbes d’apprentissage)

## Question 6

Selon le graphique précédent la valeur de l’hyper-paramètre σ devrait se situer autour de 0.5 qui minimise le taux d’erreur.