UNIVERSITÉ DE MONTRÉAL

Rapport : Devoir1

Pierre Gérard
Mathieu Bouchard

IFT3395-6390 Fondements de l'apprentissage machine
Pascal Vincent, Alexandre de Brébisson et César Laurent

Année académique 2015 - 2016

Table des matières

1 Partie Théorique : Estimation de densité

1.1 Densité avec des fenêtres de Parzen a noyau Gaussien isotropique

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

b) Si on choisit de prendre tous les points du voisinages $p(x) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{N}_{x^{(i)},\sigma^2}$

c) Hyper-paramètre : $\sigma \in \mathbb{R}$

Paramètre : On mémorise l'ensemble des données d'entrainement. $(n \times \mathbb{R}^d)$

On considère cette méthode comme non-paramétrique car le nombre de paramètre varie avec la taille de l'ensemble de données.

d) Cette densité de probabilité est toujours positive. En effet, l'ensemble de paramètre est positif et on les multiplie et additionne.

L'intégral étant compliqué à calculer. Nous avons testé qu'elle somme à de manière empirique en calculant 2000 p(x) different sur l'interval de -10 à 10.

```
def parzenEqualToOne(self):
        sigma = 0.349
        dg = densite_fonction.DensiteParzen(1, sigma)
        dg.train(self.oneDimData)
        input = [ i/100. for i in range(-1000,1000)]
        fig = plt.figure()
        ax = fig.add_subplot(111)
        n, bins, rectangles = ax.hist(input, 50, normed=True)
        print("Resultat = " + str(numpy.sum(n * numpy.diff(bins))))
```

Ce code donne $Resultat = 1.0$

1.2 Densité paramétrique avec Gaussienne isotropique

a) $p(x) = \frac{1}{(2\pi)^{d/2}\sigma^d} e^{\frac{-||x-\mu||^2}{2\sigma^2}}$

μ est le est le vecteur des moyennes, de dimension d. Σ est la matrice de covariance, de dimension 1.

b) Dans le cas de l'estimation par fenêtre de Parzen, il nous faut effectuer une moyenne sur n densités Gaussiennes et, pour chacune d'elles, le point x(i) sert de moyenne $\mathcal{N}_{x^{(i)},\sigma^2}$ avec σ qui est un hyperparamètre. Dans le cas présent, on utilise une seule densité Gaussienne pour laquelle il faudra apprendre les valeurs de moyenne μ et d'écart-type σ.

c)

Paramètres :

– μ est le est le vecteur des moyennes, de dimension d.

– Σ est la matrice de covariance, de dimension 1.

Pas d'hyper-paramètre.

1.3 Densité paramétrique avec Gaussienne diagonale

a) $p(x) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} e^{\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$

– μ est le est le vecteur des moyennes, de dimension d.

– Σ est la matrice de covariance, de dimension d.

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_n \end{bmatrix}$$

b)Indépendance des composantes d'un vecteur aléatoire suivant une distribution Gaussienne diagonale :

Soit $x \in \mathbb{R}^d$ un vecteur aléatoire qui suit une distribution Gaussienne diagonale $\mathcal{N}_{\mu,\sigma(x)}$
Posons $x - \mu = k$ et $\frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} = s$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_d \end{bmatrix} \Sigma^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\sigma_2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{\sigma_d} \end{bmatrix}$$

$p(x) = \mathcal{N}_{\mu,\Sigma(x)} = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} e^{\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} = s \times e^{\frac{-1}{2}k^t\Sigma^{-1}k}$
$p(x) = s \times e^{\frac{-1}{2}(\frac{k_1}{\sigma_1^2}\frac{k_2}{\sigma_2^2}\cdots\frac{k_d}{\sigma_d^2})k} = s \times e^{\frac{-1}{2}\sum_{i=1}^d \frac{k_i^2}{\sigma_i^2}}$
$p(x) = s\prod_{i=1}^d e^{\frac{-1}{2}\frac{k_i^2}{\sigma_i^2}} = s\prod_{i=1}^d e^{\frac{-1}{2}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu)}$ où x^i est un vecteur de taille d avec la i^{eme} valeur de x en position i et des 0 partout ailleurs.
$p(x) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}}\prod_{i=1}^d e^{\frac{-1}{2}(x^i-\mu)^T\Sigma^{-1}(x^i-\mu)}$
$p(x) = \prod_{i=1}^d p(x^i)$
La densité estimée pour à partir d'un vecteur aléatoire $x \in R$ qui suit une distribution gaussienne diagonale est égale au produit des densités estimées des composantes de ce vecteur aléatoire. On peut en conclure que ces composantes sont des variables aléatoires indépendantes.
c) $J(\theta) = \widehat{R}(f, D) = \frac{1}{n}\sum_{i=1}^n -log(p(x^i))$
On cherche donc θ^* tel que $\theta^* = argmin_\theta \widehat{R}(f, D)$
$\theta = (\mu, \sigma)$
d) On calcul le gradient $\nabla J(\theta) =$
$\widehat{R}(f, D) = \frac{1}{n}\sum_{i=1}^n -log(\frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}} e^{\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)})$
$\widehat{R}(f, D) = \frac{1}{n}\sum_{i=1}^n -(-log((2\pi)^{d/2}\sqrt{|\ \Sigma\ |}) + (\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)))$
$\widehat{R}(f, D) = \frac{1}{n}\sum_{i=1}^n log((2\pi)^{d/2}) + log(\sqrt{|\ \Sigma\ |}) - (\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$
Pour une matrice diagonale, le déterminant vaut le produit des éléments sur la diagonale.
$\sqrt{|\ \Sigma\ |} = \sqrt{\prod_{k=1}^d \sigma_k^2} = \prod_{k=1}^d \sqrt{\sigma_k^2} = \prod_{k=1}^d \sigma_k$ car σ positif
$\widehat{R}(f, D) = \frac{1}{n}\sum_{i=1}^n log((2\pi)^{d/2}) + log(\prod_{k=1}^d \sigma_k) - (\frac{-1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$
Dérivé partielle pour μ
$\frac{\partial J}{\partial \mu} = 0 + 0 + \frac{\partial}{\partial \mu}\sum_{i=1}^n -(\frac{-1}{2n}(x-\mu)^T\Sigma^{-1}(x-\mu))$

1.4 Choix du modèle

La première tâche à effectuer pour choisir le modèle le plus approprié est de sélectionner un ensemble de tous les modèles que nous pourrions effectuer. Ici, il s'agit des approches présentées aux sections 1.1, 1.2 et 1.3. Ensuite, pour chacun des modèles sélectionnées, on procède ainsi :

1. Soit A , le modèle courant.
2. Si le modèle contient des hyper-paramètres, répéter la procédure pour toutes les valeurs acceptables que ces hyper-paramètres peuvent prendre. Appelons cette configuration λ. Rappelons que l'approche des fenêtres de Parzen à noyau isotropique contient un hyper- paramètre (l'écart-type σ) tandis que les approches 2 et 3 n'en n'ont aucun ($\lambda = \emptyset$).
3. Mélanger les éléments de l'ensemble D s'ils étaient ordonnés.
4. Séparer l'ensemble d'entraînement D en deux sous-ensembles D_{train} et D_{valid}. L'ensemble d'entraînement devrait être plus grand que l'ensemble de validation. $x \in \mathbb{R}$ qui suit une distribution Gaussienne
5. Entraîner A avec la configuration d'hyper-paramètres choisis sur l'ensemble d'entraînement. On obtient alors : $\widehat{f}(A_\lambda) = A_\lambda(D_{train})$
6. Evaluer le résultat à l'aide de l'ensemble de validation. $e_{(A_\lambda)} = \widehat{R}(\widehat{f}_{(A_\lambda)}, D_{valid})$

Une fois que ce travail a été effectuer pour tous les algorithmes et, pour chacun, sur chaque configurations d'hyper-paramètres λ jugés admissibles, on regarde les valeurs de $e_{(A_\lambda)}$ obtenues. Le modèle A_λ qui retourne une erreur de validation minimale $e*_{A_\lambda}$ sera alors sélectionner. Nous aurions également pu séparer D en 3 sous-ensembles plutôt que 2 et ainsi ajouter un ensemble de test. Une fois l'algorithme optimal trouvé à l'aide de l'ensemble d'entraînement et de l'ensemble de validation, nous aurions pu calculer un estimé non-biaisé de la performance de généralisation de l'algorithme choisit en calculant l'erreur empirique de validation sur l'ensemble de test ; soit un ensemble de données dont le valeurs n'ont jamais été utilisées pour sélectionner l'algorithme dans les étapes précédantes. Cette étape nous permet de nous assurer (si nous avons assez de donner pour pouvoir effectuer un test concluent) que l'algorithme sélectionné est capable de généraliser sur de nouvelles données et qu'il n'est pas juste efficace sur les données de D_{train} et D_{valid}.

2 Partie Théorique : Classifieurs de Bayes

2.1 Classifieur de Bayes obtenu avec des densités paramétriques Gaussiennes diagonales

a)Pour la probabilité a postériori.

– μ est le est le vecteur des moyennes, de dimension d.

– Σ est la matrice de covariance, de dimension d.

Pour la probabilité a priori.

– La proportion d'element de chaque classe, de dimension m.

b)

c) $P(Y = c|X = x) = \frac{P(X=x|Y=c)P(Y=c)}{P(X=x)}$

$P(Y = c|X = x) = \frac{\hat{p}_c(x)\hat{P}_c}{\sum_{c'=1}^{m} \hat{p}_{c'}\hat{P}_{c'}}$

avec $\hat{p}_c(x)$ la densité de probabilité apprise (1.2 a) sur tout les points de la classe c.

2.2 Classifieur de Bayes obtenu avec des fenêtres de Parzen à noyau Gaussien isotropique

a) Pour la probabilité à postériori

– L'ensemble des données d'entrainement.

Pour la probabilité a priori.

– La proportion d'element de chaque classe, de dimension m.

b) $P(Y = c|X = x) = \frac{P(X=x|Y=c)P(Y=c)}{P(X=x)}$

$P(Y = c|X = x) = \frac{\hat{p}_c(x)\hat{P}_c}{\sum_{c'=1}^{m} \hat{p}_{c'}\hat{P}_{c'}}$

avec $\hat{p}_c(x)$ la densité de probabilité apprise (1.3 a) sur tout les points de la classe c.

2.3 Classifieur de Parzen obtenu avec des fenêtres de Parzen à noyau Gaussien isotropique

a)

$P(Y = c|X = x) = \frac{\hat{p}_c(x)}{\sum_{c'=1}^{m} \hat{p}_{c'}}$

avec $\hat{p}_c(x) = \frac{1}{\sum_{i=1}^{n} K(x_i,x)} \sum_{i=1}^{n} K(x_i, x) onehot_m(Y_i)$

et avec $K(x_i, x) = \frac{1}{(2\pi)^{d/2}\sigma^d} e^{\frac{-1}{2}\frac{d(x^i,x)^2}{\sigma^2}}$

b)

3 Implémentation

Pour tester notre code et l'implémentation, exécuter simplement dans le dossier *src* :

```
python main.py
```

L'ensemble des graphiques vont être sauvé comme image.

Note : Les modules suivants utilisé en labo sont nécessaire : numpy, matplotlib, pylab.

4 Partie Pratique : Estimation de densité

4.1 Densité 1D

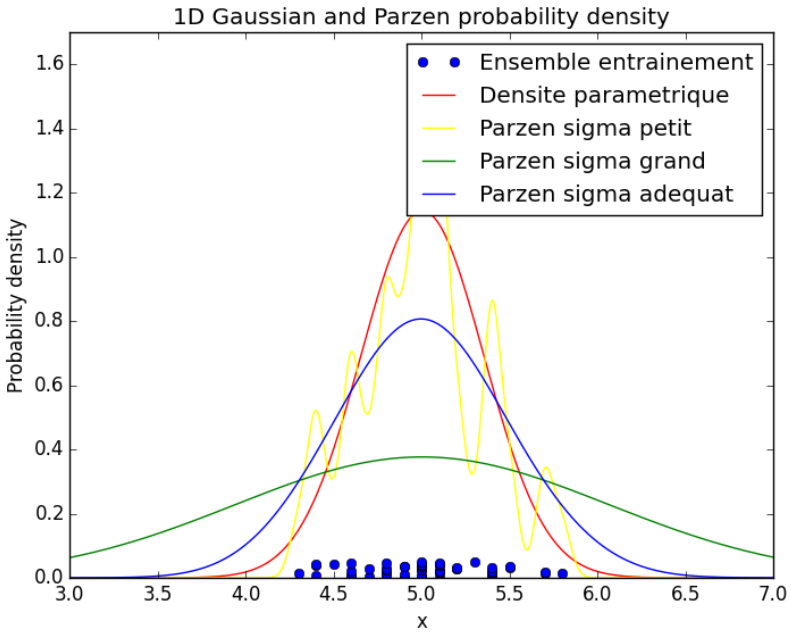

FIGURE 1 – Comparaison de la densité paramétrique et la densité de Parzen avec différents sigma

4.2 Densité 2D

4.2.1 Densité paramétrique gaussienne

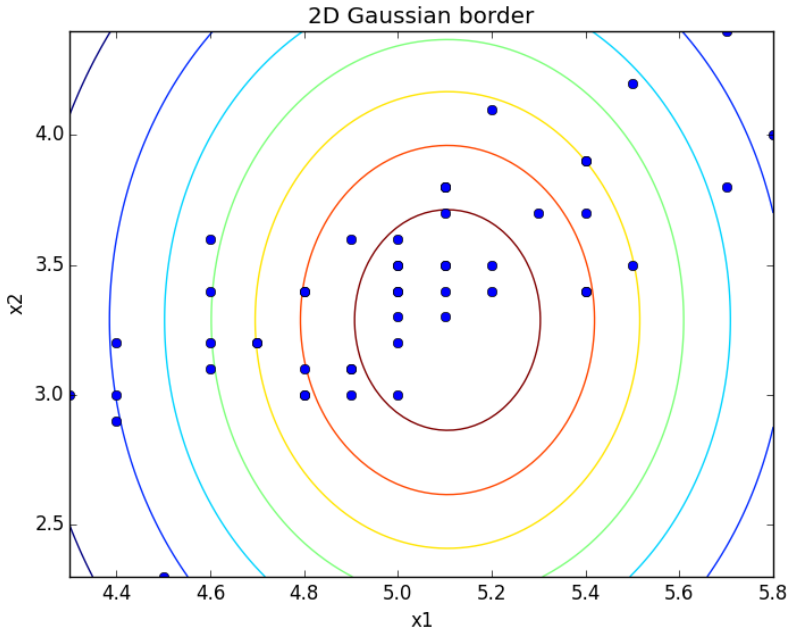

FIGURE 2 – Densité paramétrique gaussienne

4.2.2 Densité de Parzen : sigma trop petit

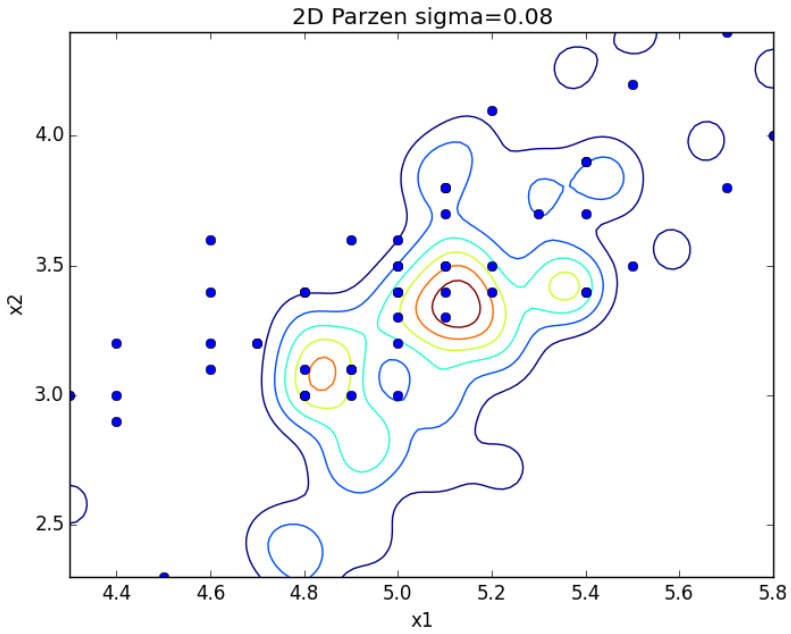

FIGURE 3 – Densité de Parzen : sigma trop petit

4.2.3 Densité de Parzen : sigma trop grand

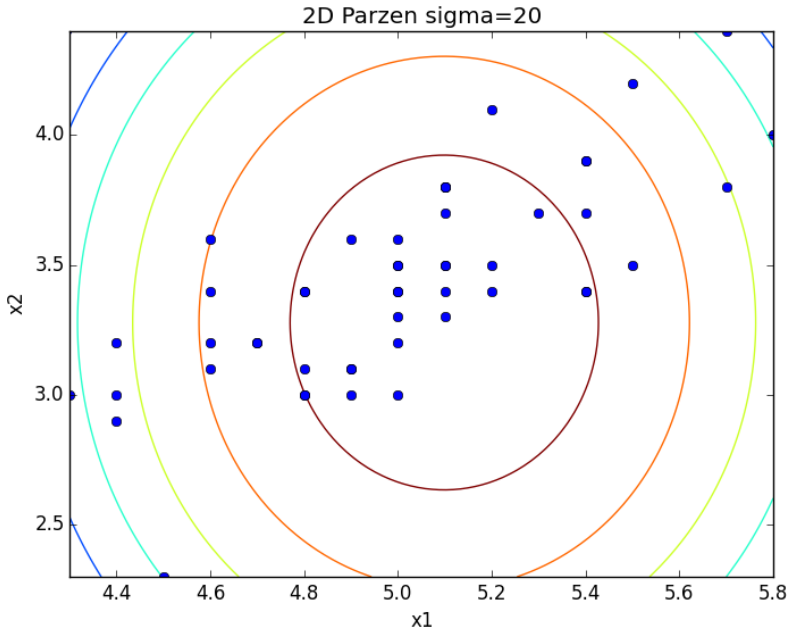

FIGURE 4 – Densité de Parzen : sigma trop grand

4.2.4 Densité de Parzen : sigma adéquat

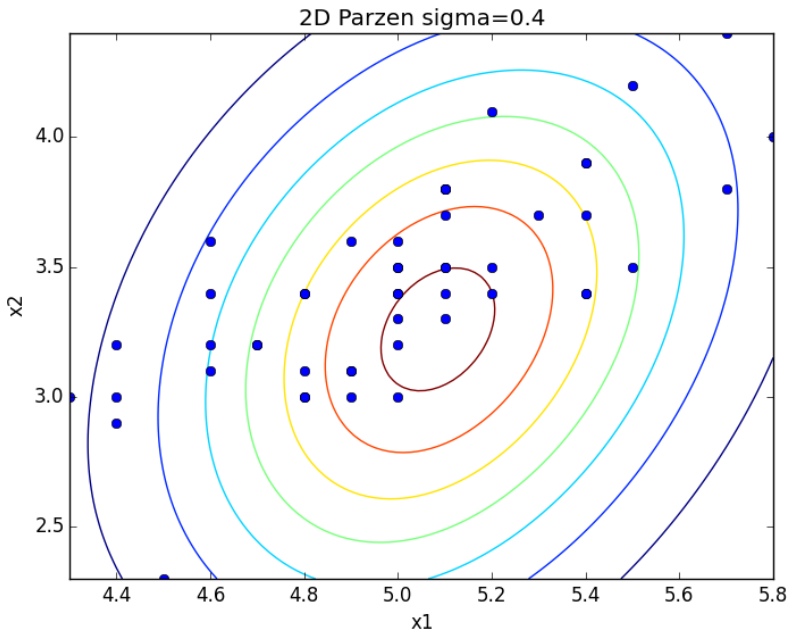

FIGURE 5 – Densité de Parzen : sigma adéquat

5 Partie Pratique : Classifieur de Bayes

5.1 Avec densités paramétriques gaussienne diagonales

5.1.1 Graphique pour d=2

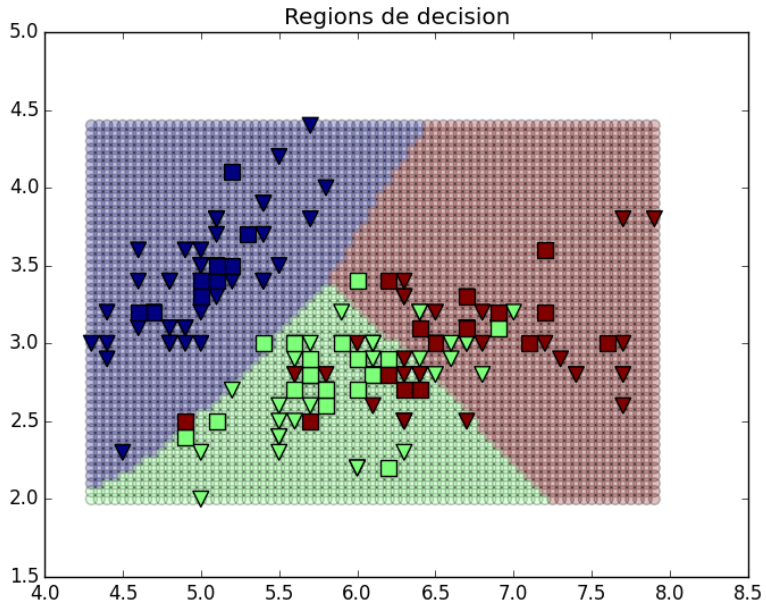

FIGURE 6 – Graphique pour d=2

5.1.2 Calcul d'erreur pour d=2

```
d=2
Taux d'erreur sur l'ensemble d'entrainement: 24.07%
Taux d'erreur sur l'ensemble de validation: 19.05%
```

5.1.3 Calcul d'erreur pour d=4

```
d=4
Taux d'erreur sur l'ensemble d'entrainement: 4.63%
Taux d'erreur sur l'ensemble de validation: 4.76%
```

5.2 Avec densités de Parzen gaussienne isotropique

5.2.1 Classifieur de Parzen : sigma trop petit

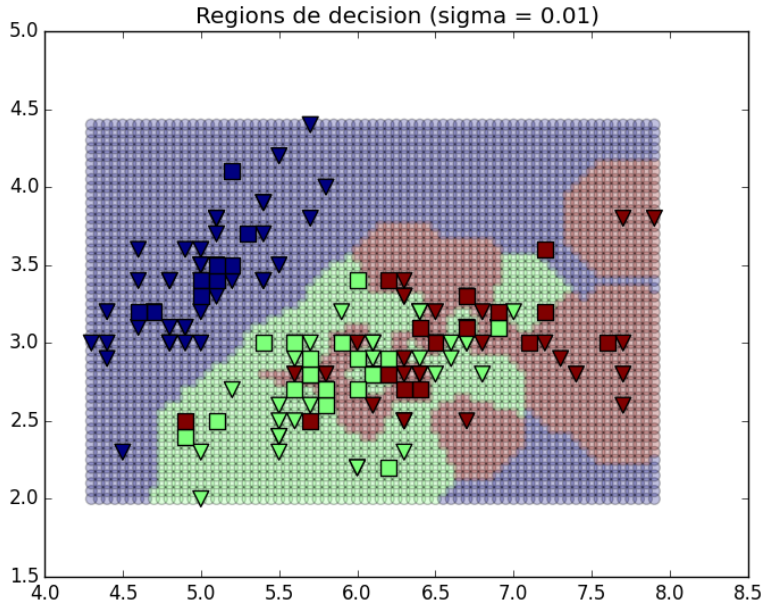

FIGURE 7 – Classifieur de Parzen : sigma trop petit

5.2.2 Classifieur de Parzen : sigma trop grand

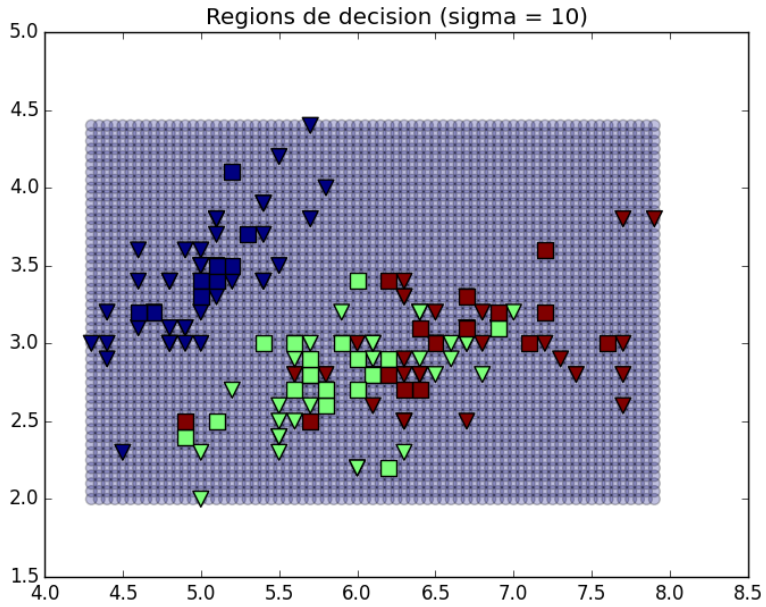

FIGURE 8 – Classifieur de Parzen : sigma trop grand

5.2.3 Classifieur de Parzen : sigma adéquat

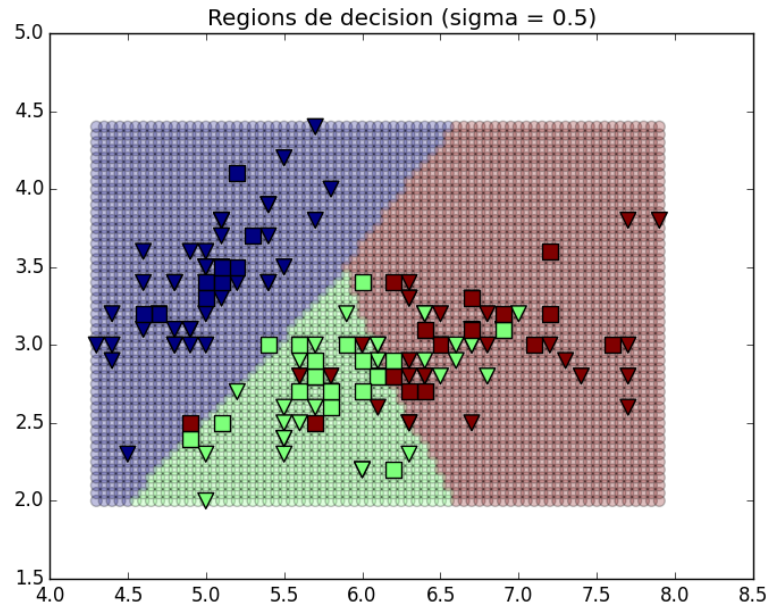

FIGURE 9 – Classifieur de Parzen : sigma adéquat

5.2.4 Courbe d'apprentissage 2D

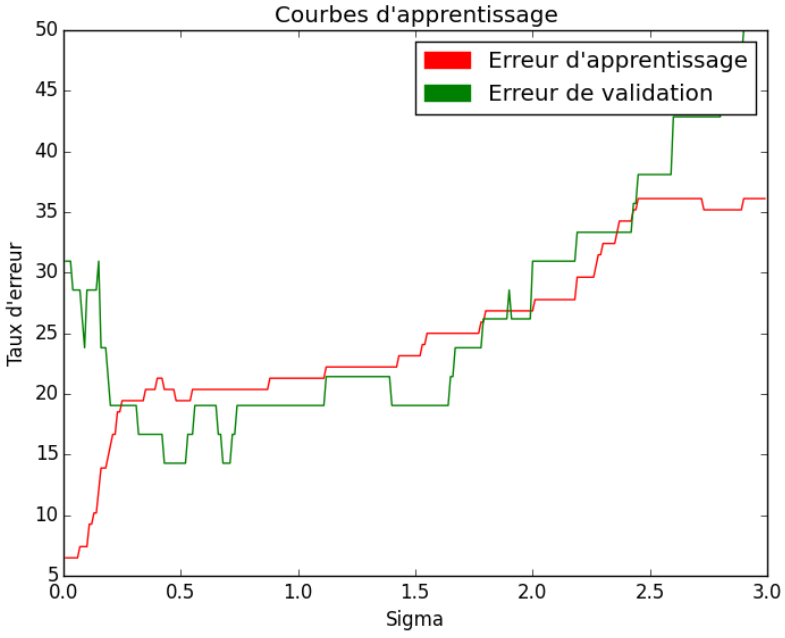

FIGURE 10 – Courbe d'apprentissage 2D

Meilleur valeur : $\sigma = 0.43$

5.2.5 Courbe d'apprentissage 4D

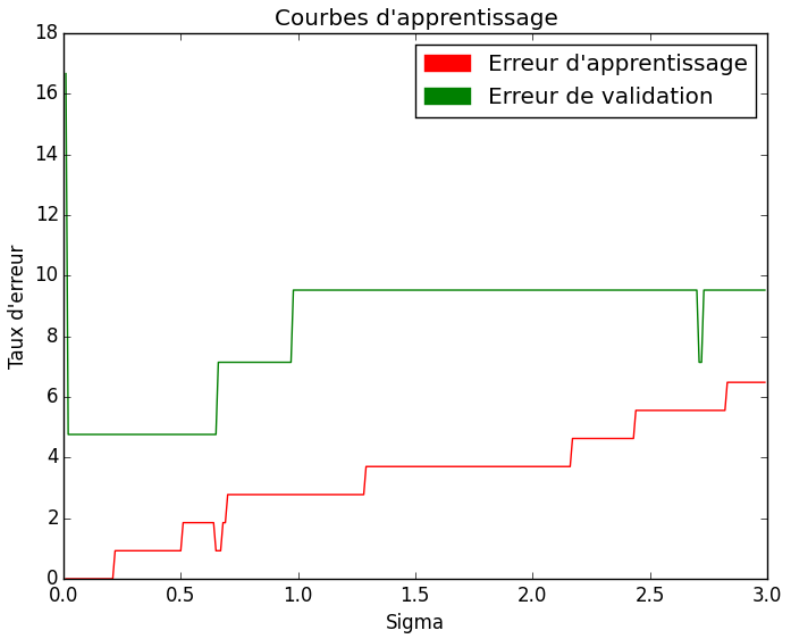

FIGURE 11 – Courbe d'apprentissage 4D

Meilleur valeur : $\sigma = 0.02$

5.2.6 Meilleur choix

Cette exécution avec cette division particulière montre que :
- La classification en dimension 4 est meilleur que la classification en dimension 2,
- La classification avec un noyau de Parzen donne le même résultat en dimension 4 et un meilleur en dimension 2,
- En dimension 2, l'erreur de validation avec gaussienne diagonale est de 19.05 % et de 14.29 % pour Parzen,
- En dimension 4, l'erreur de validation avec gaussienne diagonale et Parzen valent tous les deux 4.76 %,
- La meilleur valeur de σ en dimension 4 est de 0.02,
- La meilleur valeur de σ en dimension 2 est de 0.43.

On peut essayer de généraliser les résultats. Ce choix sera réalisé de manière empirique en testant plusieurs *seeds* différents de manière a faire ressortir une "tendance général".

Ces exécutions différentes semblent montrer que :
- La classification en dimension 4 est meilleur que la classification en dimension 2,
- La classification avec un noyau de Parzen semble donner de meilleurs résultats,
- La valeur de sigma "optimal" varie grandement en fonction des données.
- L'erreur de classification avec gaussienne diagonale semble varier entre 2 et 6 % et entre 0 et 4 % pour Parzen.