Université de Technologie de Compiègne
SY09:Analyse des données et Data-Mining , P17

Rendu TP3

Zineb Slam, Oumaima Talouka

6 juin 2017

Résumé

Dans ce rapport de TP3 nous allons étudier 2 méthodes d'apprentissage supervise : le Classifieur Euclidien et les K Plus Proches Voisins. Dans la première partie nous allons implémenter ces 2 méthodes en R. Ensuite nous allons évaluer leur performance avec différents jeux de données. Pour finir, nous allons calculer la règle de Bayes. Les fonctions ayant permis l'obtention de nos résultats sont jointes au rapport du TP.

1 Classifieur euclidien, K plus proches voisins

1.1 Programmation

Nous avons essaye dans notre implémentation d'éviter les boucles au maximum possible, car celle ci réduisent les performances. La fonction *ceuc.app* retourne les centres d'inertie de chaque classe pour l'ensemble d'apprentissage. La fonction *rowsum* sur R nous a été très utile pour sommer les lignes de chaque classe .
La fonction *ceuc.val* prédit la classe de chaque individu des données tests. Nous utilisons ici la fonction *distXY* pour calculer la distance qui sépare chaque individu de chaque centres d'inertie.

1.1.1 Test de fonctions

Pour tester nos fonctions nous allons représenter graphiquement les frontières de décision de chaque méthode en appelant les fonctions ***front.ceuc*** et ***front.val*** sur les données Synth1-40. On obtient les graphes représentés dans les figures ci-dessous.

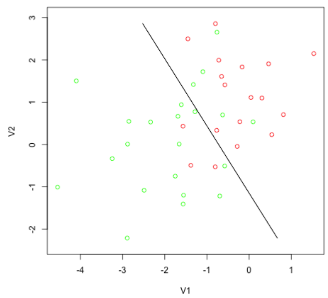

FIGURE 1 – Frontière de decision obtenue avec le Classifieur Eucledien

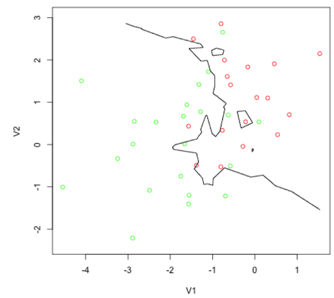

FIGURE 2 – Frontière de décision obtenue avec la methode des K plus proches voisins (K=3)

Remarquons que la frontière de décision du classifieur Euclidien est linéaire tandis que celle du KPP est non linéaire.

1.2 Évaluation des Performances

1.2.1 Estimation des paramètres

Dans cette partie on estime les moyennes μ_k, les matrices de covariances Σ_k ainsi que les proportions π_k pour chaque jeu de données et pour chaque classe k. Nous utilisons pour cela 2 fonctions sur R : *group_by* et *summarize*. La fonction *group_by* va grouper les données selon les classes z ensuite *summarize* va calculer les paramètres pour chaque groupement. Dans cet exemple on obtient les paramètres de la variable V1.

		Estimation des Paramètres		
Synth1	k	μ_k	$\sum_k$	π_k
40	1	$\begin{pmatrix} -0.32 \\ 1.09 \end{pmatrix}$	$\begin{bmatrix} 0.68 & 0.12 \\ 0.12 & 1.01 \end{bmatrix}$	0.45
	2	$\begin{pmatrix} -1.883 \\ 0.105 \end{pmatrix}$	$\begin{bmatrix} 1.37 & 0.32 \\ 0.32 & 1.44 \end{bmatrix}$	0.55
100	1	$\begin{pmatrix} 0.02 \\ 0.82 \end{pmatrix}$	$\begin{bmatrix} 0.88 & -0.13 \\ -0.13 & 1.12 \end{bmatrix}$	0.54
	2	$\begin{pmatrix} -1.96 \\ -0.13 \end{pmatrix}$	$\begin{bmatrix} 0.76 & -0.04 \\ -0.04 & 0.76 \end{bmatrix}$	0.46

		Estimation des Paramètres		
Synth1	k	μ_k	$\sum_k$	π_k
500	1	$\begin{pmatrix} 0.13 \\ 0.88 \end{pmatrix}$	$\begin{bmatrix} 1.05 & 0.052 \\ 0.052 & 0.98 \end{bmatrix}$	0.53
	2	$\begin{pmatrix} -1.88 \\ -0.08 \end{pmatrix}$	$\begin{bmatrix} 0.97 & -0.11 \\ -0.11 & 0.98 \end{bmatrix}$	0.47
1000	1	$\begin{pmatrix} -0.01 \\ 0.91 \end{pmatrix}$	$\begin{bmatrix} 0.97 & -0.06 \\ -0.06 & 1.08 \end{bmatrix}$	0.50
	2	$\begin{pmatrix} -1.96 \\ 0.02 \end{pmatrix}$	$\begin{bmatrix} 0.99 & 0.02 \\ 0.02 & 0.94 \end{bmatrix}$	0.50

D'après les résultats du tableau des paramètres de ces jeux de données, nous pouvons remarquer que les matrices $\sum_k$ se rapprochent de la forme globale : $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ pour les deux classes, ainsi que des proportions π_k qui sont quasiment toutes à peu près égales à 0,5. Ainsi, nous pouvons en déduire que les nuages de points sont approximativement sphériques et de même volume. Il est donc possible de séparer les classes par un hyperplan avec le classifieur euclidien.

1.2.2 Calcul du taux d'erreur du classifieur Euclidien

Dans cette partie nous allons tester les performances de chacun des algorithmes programmes precedement en utilisant le critère de l'erreur qu'on exprime comme :

$$Erreur = \frac{1}{N}\sum_{i=1}^{n} \mathbb{1}\hat{z}_i \neq z_i$$

Avec z_i la vrai valeur de z et $\hat{z}_i$ la valeur calculée a partir de l'algorithme. N est le nombre d'individus. $\mathbb{1}\hat{z}_i \neq z_i = 1$ si $\hat{z}_i = z_i$, 0 sinon. Après avoir calculé l'erreur qu'on suppose suit une loi normale de moyenne μ_ϵ et de variance σ_ϵ, on peut calculer l'intervalle de confiance exprimé ci-dessous.

$$Ic = [\mu_\epsilon - t\frac{\sigma_\epsilon^2}{\sqrt{N}}, \mu_\epsilon + t\frac{\sigma_\epsilon^2}{\sqrt{N}}]$$

On choisit un niveau de confiance t de 95%. Notons ici que μ et σ sont ceux calculés pour les différentes valeurs de ϵ et non ceux du jeu de données. Autre point a remarquer est le N qui est ici le nombre d'individus , or comme on a séparé nos jeux de données en un ensemble d'application et un ensemble de test, on veillera a mettre le nombre d'individus correspondant en fonction si c'est l'intervalle de confiance d'application ou test. D'après l'énoncé en utilisant la fonction *separ1* napp $=\frac{2n}{3}$ et ntst$=\frac{n}{3}$ n étant le nombre total d'individus du jeu de donnée.

On fait tourner 20 fois le classifier Euclidien ainsi que le KPP puis on calcules les erreurs obtenus. Les résultats obtenus pour les jeux de données sont affichés dans Le tableau qui suit.

	Performance du Classifieur Euclidien			
Synth1	ϵ_{app}	Ic_{app}	ϵ_{test}	Ic_{test}
40	0.23	$[0.218, 0.234]$	0.18	$[0.155, 0.205]$
100	0.085	$[0.083, 0.088]$	0.102	$[0.093, 0.111]$
500	0.132	$[0.131, 0.132]$	0.134	$[0.132, 0.136]$
1000	0.140	$[0.140, 0.141]$	0.147	$[0.146, 0.147]$

En remarquant les intervalles de confiance on remarque que l'erreur d'apprentissage est inférieure a celle du test ce qui est normal vu qu'on entrainé le modèle avec ses données. On remarque aussi que plus le nombre d'individus augmente plus l'intervalle de confiance diminue diminue ce qui est logique car avec plus de données dans le modèle d'apprentissage apprend mieux les données et donc pourra mieux classer l'ensemble de test. Notons ici le faible taux d'erreur qu'on a obtenu avec les donnees **Synth1-100**, ce qui peut revenir a la génération de données.

1.2.3 Le nombre Optimal de voisins du KPP

Le nombre optimal de voisins qu'en peut obtenir en ayant l'ensemble d'apprentissage comme ensemble de validation (Xval = Xapp) est toujours de 1 puisque le plus proche voisin d'un individu est lui même et donc chaque individu sera attribue a sa vraie classe (zval = zapp). L'erreur sera donc nulle dans ce cas et le modèle sera alors biaisé, d'ou l'intérêt d'avoir un ensemble d'apprentissage, un ensemble de validation pour déterminer le meilleur k puis un ensemble de test.

1.2.4 Calcul du taux d'erreur du KPP

De la même manière qu'avec le Classifieur Euclidien on obtient les résultats des taux d'erreurs ci-dessous. Notons ici que la notion d'erreur d'apprentissage n'a pas de sens ici car le modèle n'a pas besoin d'apprendre d'abord sur des données puis tester, il fait tout en un seul appel de fonction.

	Performance du KPP	
Jeu de données Synth1	ϵ_{test}	Ic_{test}
40	0.295	$[0.253, 0.337]$
100	0.104	$[0.094, 0.114]$
500	0.152	$[0.149, 0.154]$
1000	0.146	$[0.144, 0.147]$

1.3 Analyse des résultats

En comparant les 2 méthodes on remarque que l'erreur ainsi que les intervalles de confiance du Classifier Euclidien sont bien inférieur a ceux de la méthode des KPP. On a donc peut être des...

1.3.1 Jeux de données Synth2-1000

Estimation des Paramètres				
Jeu de données	k	μ_k	$\sum_k$	π_k
Synth2-1000	1	$\begin{pmatrix} 3.018 \\ -0.0063 \end{pmatrix}$	$\begin{bmatrix} 0.9904 & 0.1131 \\ 0.1131 & 1.092 \end{bmatrix}$	0.52
	2	$\begin{pmatrix} -2.142 \\ -0.0265 \end{pmatrix}$	$\begin{bmatrix} 4.435 & -0.154 \\ -0.154 & 1.030 \end{bmatrix}$	0.48

Grâce a cette estimation on peut remarquer que

$$\lim \pi_1 = \lim \pi_1 = 0.5 \quad \mu_1 = (3,0)^T \quad \mu_2 = (-2,0)^T \quad \Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$$

	Performance du Classifier Euclidien			
Jeu de données	ϵ_{app}	Ic_{app}	ϵ_{test}	Ic_{test}
Synth2-1000	0.061	$[0.061, 0.062]$	0.063	$[0.062, 0.064]$

	Performance du KPP	
Jeu de données	ϵ_{test}	Ic_{test}
Synth2-1000	0.062	$[0.061, 0.063]$

Contrairement aux jeux de données *Synth1-n* on remarque le KPP semble donner des résultats légèrement meilleures avec *Synth2*.

1.3.2 Jeux de données réelles : Pima & Breast Cancer

	Performance du Classifier Euclidien			
Jeu de données	ϵ_{app}	Ic_{app}	ϵ_{test}	Ic_{test}
Pima	0.252	$[0.251, 0.252]$	0.236	$[0.234, 0.238]$
Cancer	0.040	$[0.040, 0.0407]$	0.039	$[0.038, 0.039]$

	Performance du KPP	
Jeu de données	ϵ_{test}	Ic_{test}
Pima	0.248	$[0.245, 0.250]$
Cancer	0.040	$[0.039, 0.042]$

2 Règle de Bayes

2.1 Loi Marginale

La première remarque qu'on peur faire concernant les paramètres des données de Synth1n et Synth2 est que les matrices de variances $\sum_1$ et $\sum_2$ sont diagonales, donc les covariances COV(V1, V2) =0. Ainsi on peut en déduire que les variables sont indépendantes. Cette remarque va nous être utile dans le calcul de la loi marginale de chaque classe ω_1 et ω_2. Ainsi on peut écrire :

$$\begin{aligned} f(X^1, X^2|\omega_1) &= f(X^1|\omega_1)f(X^2|\omega_1) \\ f(X^1, X^2|\omega_2) &= f(X^1|\omega_2)f(X^2|\omega_2) \end{aligned} \tag{1}$$

Les individus de la classe 1 et 2 suivent une loi normale bivariee, donc on peut écrire :

$$f(X|\omega_k) = \frac{1}{\sqrt{2\pi|\Sigma_k|}} exp(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)) \quad avec \quad x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{2}$$

2.1.1 Synth1

Dans le cas des jeux de données Synth1 nous avons

$$\mu 1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \quad \Sigma = \Sigma_1 = \Sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad et \quad \pi_1 = \pi_2 = 0.5$$

D'après la formule 2 on peut alors exprimer les densités comme suit :

$$\begin{array}{c|c} f_{X^1}(x_1|\omega_1|) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x_1^2) & f_{X^1}(x_1|\omega_2|) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}(x_1+2)^2) \\ f_{X^2}(x_2|\omega_1|) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}(x_2-1)^2) & f_{X^2}(x_2|\omega_2|) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x_2^2) \end{array} \tag{3}$$

On obtient ainsi selon (1) les densités jointes de chacune des classes ci-dessous :

$$\begin{aligned} f_{X^1,X^2}(x|\omega_1) &= \frac{1}{2\pi} exp(-\frac{1}{2}(x_1^2 + (x_2-1)^2)) \\ f_{X^1,X^2}(x|\omega_2) &= \frac{1}{2\pi} exp(-\frac{1}{2}(x_2^2 + (x_1+2)^2)) \end{aligned} \tag{4}$$

2.1.2 Synth2

Dans le cas des jeux de donnes Synth2 nous avons :

$$\mu 1 = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \quad \mu_2 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \quad \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 5 & 0 \\ 0 & 1 \end{pmatrix} et \quad \pi_1 = \pi_2 = 0.5$$

Les densités des variable pour chaque classe sont alors :

$$\begin{array}{c|c} f_{X^1}(x_1|\omega_1|) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}(x_1-3)^2) & f_{X^1}(x_1|\omega_2|) = \frac{1}{\sqrt{10\pi}} exp(-\frac{1}{10}(x_1+2)^2) \\ f_{X^2}(x_2|\omega_1|) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x_2^2) & f_{X^2}(x_2|\omega_2|) = \frac{1}{\sqrt{10\pi}} exp(-\frac{1}{2}x_2^2) \end{array} \tag{5}$$

On obtient ainsi selon (1) les densités jointes de chacune des classes ci-dessous :

$$\begin{aligned} f_{X^1,X^2}(x|\omega_1) &= \frac{1}{2\pi} exp(-\frac{1}{2}(x_1-3)^2 - \frac{1}{2}x_2^2) \\ f_{X^1,X^2}(x|\omega_2) &= \frac{1}{10\pi} exp(-\frac{1}{10}(x_1+2)^2 - \frac{1}{2}x_2^2) \end{aligned} \tag{6}$$

2.2 Courbe d'iso-densité

Une courbes d'iso-densité correspond a la régions ou la densité est constante.

2.2.1 iso-densité de Synth1

Pour la classe ω_1 $f(X^1, X^2|\omega_1) = C_1 \quad avec \quad C_1 constante$

$$\begin{aligned} exp(-\frac{1}{2}(x_1-\mu_1)^2 - \frac{1}{2}(x_2-\mu 1)^2) &= 2\pi\sigma_{11}^2\sigma_{22}^2 C_1 \\ (x_1-\mu_1)^2 + (x_2-\mu_1)^2 &= -2ln(2\pi\sigma_{11}^2\sigma_{22}^2 C_1) \\ x_1^2 + (x_2-1)^2 &= -2ln(2\pi C_1) \end{aligned} \tag{7}$$

L'équation (5) obtenue est une équation de cercle de centre $\mu_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ et de rayon $R^2 = -2ln(2\pi C_1)$.

De même pour la classe ω_2 $f(X^1, X^2|\omega_2) = C_2 \quad avec \quad C_2 constante$

$$\begin{aligned} (x_1-\mu_2)^2 + (x_2-\mu_2)^2 &= -2ln(2\pi\sigma_{11}^2\sigma_{22}^2 C_2) \\ (x_1+2)^2 + x_2^2 &= -2ln(2\pi C_2) \end{aligned} \tag{8}$$

On obtient encore une fois l'équation de cercle de centre $\mu_2 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$ et de rayon $R^2 = -2ln(2\pi C_2)$.

2.2.2 iso-densité de Synth2

Pour la classe ω_1 $f(X^1, X^2|\omega_1) = C_3 \quad avec \quad C_3 constante$

$$\begin{aligned} exp(-\frac{1}{2}(x_1-\mu_1)^2 - \frac{1}{2}x_2^2) &= 2\pi\sigma_{11}^2\sigma_{22}^2 C_3 \\ (x_1-\mu_1)^2 + x_2^2 &= -2ln(2\pi\sigma_{11}^2\sigma_{22}^2 C_3) \\ (x_1-3)^2 + x_2^2 &= -2ln(2\pi C_3) \end{aligned} \tag{9}$$

L'équation obtenue est une équation de cercle de centre $\mu_1 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$ et de rayon $R^2 = -2ln(2\pi C_2)$.

De même pour la classe ω_2 $f(X^1, X^2|\omega_2) = C_4 \quad avec \quad C_4 constante$

$$\begin{aligned} exp(-\frac{1}{10}(x_1-\mu_1)^2 - \frac{1}{2}(x_2-\mu_2)^2) &= 10\pi\sigma_{11}^2\sigma_{22}^2 C_4 \\ (x_1-\mu_1)^2 + 5(x_2-\mu_2)^2 &= -10ln(10\pi\sigma_{11}^2\sigma_{22}^2 C_3) \\ (x_1-3)^2 + 5x_2^2 &= -10ln(50\pi C_3) \\ (x_1+2)^2 + x_2^2 &= -10ln(50\pi C_2) \end{aligned} \tag{10}$$

On obtient encore une fois l'équation de cercle de centre $\mu_2 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$ et de rayon $R^2 = -2ln(2\pi C_2)$.

2.3 Règle de Bayes

La règle de Bayes s'exprime de la manière :

$$
\begin{aligned}
\delta^*(x) = &\begin{cases} \omega_1 & si \quad \Pr(\omega_1|x) > \Pr(\omega_2|x) \\ \omega_2 & sinon \end{cases} \\
\Longleftrightarrow \ \delta^*(x) = \omega_1 \quad si \quad & \frac{f_X(x|\omega_1)}{f_X(x)}\pi_1 > \frac{f_X(x|\omega_2)}{f_X(x)}\pi_2 \\
\Longleftrightarrow \ \delta^*(x) = \omega_1 \quad si \quad & \frac{f_X(x|\omega_1)}{f_X|\omega_2(x)} > \frac{\pi_2}{\pi_1}
\end{aligned} \tag{11}
$$

Dans la suite nous notons $f_1(x) = f_X(x|\omega_1)$ et $f_2(x) = f_X(x|\omega_2)$

2.3.1 Synth1

D'après les densités calculées dans (4) on obtient :

$$\frac{f_1(x)}{f_2(x)} = exp(2x_1 + x_2 + \frac{3}{2})$$

On peut donc exprimer l'inéquation 7 de facon linaire en introduisant le *log* , ainsi on a

$$
\begin{aligned}
2x_1 + x_2 > ln\frac{\pi_2}{\pi_1} - \frac{3}{2} \quad \frac{\pi_2}{\pi_1} = 1 \\
x_2 > -2x_1 - \frac{3}{2}
\end{aligned} \tag{12}
$$

Ainsi La règle de Bayes attribuera l'individu a la classe 1 si (12) est vérifiée si non a la classe 2. Nous obtenons donc une frontière linéaire si les 2 matrices de covariances sont égales. Nous avons représenté cela de manière graphique sur les données Synth1 avec n=40 et n=50 en utilisant ggplot.

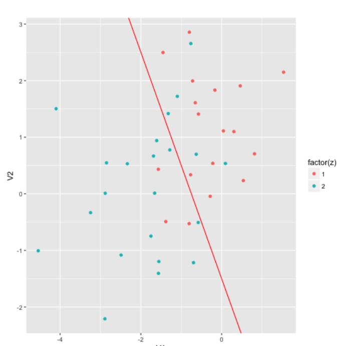

FIGURE 3 – Frontière de decision Lineaire pour Synth140

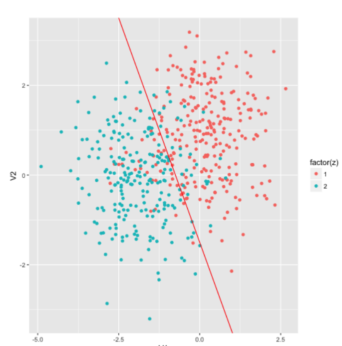

FIGURE 4 – Frontière de decision Lineaire pour Synth1500

2.3.2 Synth2

D'après les densités calculées dans (4) on obtient :

$$\frac{f_1(x)}{f_2(x)} = 5exp(-\frac{4}{10}x_1^2 + \frac{34}{10}x_1 - \frac{41}{10})$$

On obtient ainsi l'inéquation :

$$-4x_1^2 - 34x_1 - 41 > -10ln5$$

En résolvant cette inéquation de second degrés par $\Delta = b^2 - 4ac$ on obtient 2 solutions

$$k_1 = 0,342 \quad k_2 = -8.842$$

Ainsi lorsqu'on a des matrices de covariances différentes on obtient une frontière de décision quadratique. Cette frontière est représentée ci-dessous de 2 manières différentes pour les jeux de données Synth2 en utilisant **ggplot** sur R et stat_function pour dessiner la fonction quadratique.

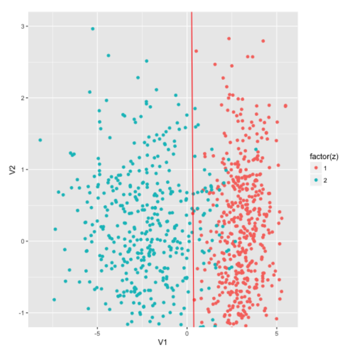

Figure 5 – Frontière de decision Quadratique pour Synth2

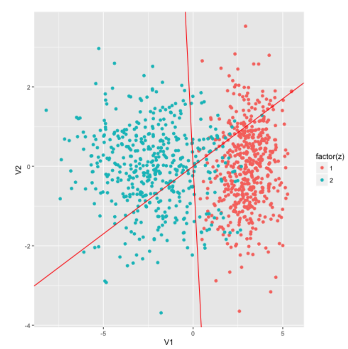

Figure 6 – Frontière de decision pour Synth2

La figure 5 montre la frontière de décision entre les 2 classes. Cette frontière qui est en effet une courbe apparait ici comme linéaire car on a restreint l'échelle a celle de nos données.

2.4 Erreur de Bayes

L'erreur de Bayes est la plus petite erreur possible que peut atteindre une règle δ utilisant uniquement X.

2.4.1 Synth1

Dans le cas des jeux de données de *Synth1n* les matrices de covariances sont égales ($\sigma_k = \sigma$) et les probabilités a priori de chaque classe aussi ($\pi_1 = \pi 2$). Avec ces conditions, l'erreur de Bayes s'écrit de la forme :

$$\epsilon^* = \phi(-\frac{\Delta}{2}).$$

ou ϕ est la fonction de répartition de la loi normale (univariee) centrée réduite et $\Delta^2 = (\mu_2 - \mu_1)^T \sigma^{-1}(\mu_2 - \mu_1)$ est la *distance au carrée de Mahalanobis.* On obtient alors $\Delta^2 = 5$, d'ou $\Delta = \sqrt{5}$ Ainsi :

$$\begin{aligned}
\epsilon^* &= \phi(-\frac{\sqrt{5}}{2}) \\
&= Pr(x \leq -\frac{\sqrt{5}}{2}) \\
&= 1 - \Pr(x \leq \frac{\sqrt{5}}{2}) \\
&= 1 - 0.86 \\
\epsilon^* &= 0.13
\end{aligned} \tag{13}$$

On remarque ainsi que cette erreur est relativement proche a l'erreur de test ϵ_{test} du classifieur Euclidien calculée dans la section 1.2

2.4.2 Synth2

Dans le cas general ou les matrices de covariances ne sont pas égales on peut déduire une borne supérieure de l'erreur de Bayes. Dans la page 98 du poly on trouve l'expression de cette borne dans le cas de 2 classes :

$$\epsilon^* \leq \sqrt{\pi_1 \pi_2} \exp -\Delta_B^2$$

Avec Δ_B^2 la distance de Bhattacharyya entre les 2 classes et qui s'exprime comme

$$\Delta_B^2 = \frac{1}{8}(\mu_2 - \mu_1)^T (\frac{\Sigma_1 + \Sigma_2}{2})^{-1}(\mu_2 - \mu_1) + \frac{1}{2} ln \frac{|\frac{\Sigma_1 + \Sigma_2}{2}|}{\sqrt{|\Sigma_1||\Sigma_2|}}$$

On obtient alors $\Delta_B^2 = \frac{25}{24} + \frac{1}{2} ln(\frac{3}{5}) = 0.79$

$$\epsilon^* \leq 0.45$$

Ainsi l'erreur maximale que peut atteindre l'erreur de Bayes est 0.45

3 Conclusion