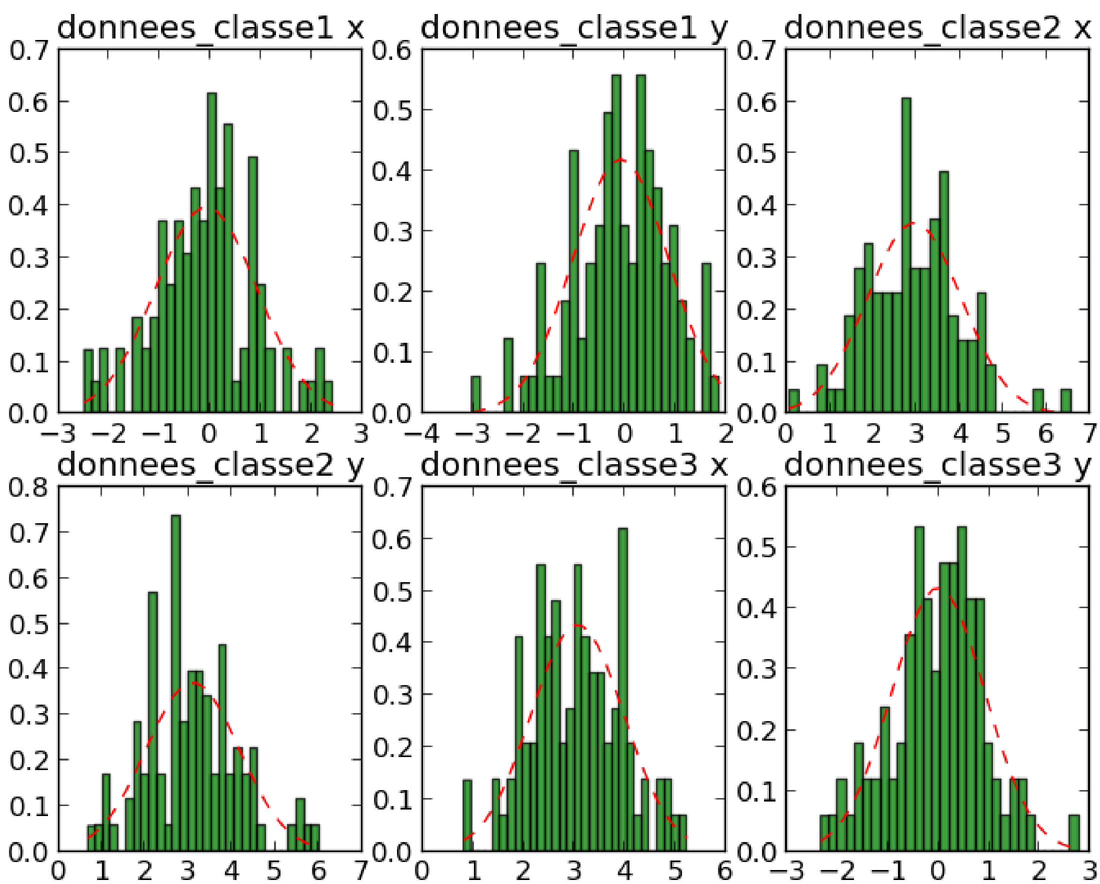
TP3
IAA

Classification par Lois Normales par approche Bayésienne

1.Analyse des données

Pour l'analyse des données, comme on a 3 classes de points, on obtient 6 histogrammes (1 pour chaque composante de chaque classe). On obtient les histogrammes ci-dessous :

x désignant la première coordonnée et y la deuxième coordonnées :



A priori, les données semblent bien suivre une loi normale. (visuellement l'histogramme est proche de la loi normale apprise tracée en rouge)

Les matrices de covariances trouvées sont :

Pour la classe 1:

[[1.0175364 -0.07091903]
 [-0.07091903 0.91352591]]

Pour la classe 2:

[[1.18931151 -0.03957458]
 [-0.03957458 1.16833718]]

Pour la classe 3:

[[0.8475625   0.11541681]
 [0.11541681 0.85057838]]

On s'aperçoit que les termes en dehors des diagonales sont négligeable , mis à part pour la classe 3 où on peut émettre un doute, mais les composantes semblent donc indépendantes entre elles.

2.Passage du log

D'après le sujet du TP4:

$$\hat{c} = arg \max_{i \in Classes} P(x | N(\mu_i, \Sigma_i))$$

Et donc en dimension d pour une loi normale:

$$\hat{c} = arg \max_{i \in Classes} \frac{1}{(2\Pi)^{d/2} |\Sigma_i|^{1/2}} \exp((-1/2)(x-\mu_i)^t * \Sigma_i^{-1} * (x-\mu_i))$$

Remarque : pour utiliser cette formule il faut que det(Σ) ≠ 0

Le terme $(2\Pi)^{d/2}$ étant une constante positive (d>0), en divisant par ce terme l'expression devient:

$$\hat{c} = arg \max_{i \in Classes} \frac{1}{|\Sigma_i|^{1/2}} \exp((-1/2)(x-\mu_i)^t * \Sigma_i^{-1} * (x-\mu_i))$$

Le log étant une fonction croissante et continue sur R+, en l'utilisant sur la formule (on a le droit car exp(x)>0 pour tout x) on obtient :

$$\hat{c} = arg \max_{i \in Classes} \log(\frac{1}{|\Sigma_i|^{1/2}} \exp((-1/2)(x-\mu_i)^t * \Sigma_i^{-1} * (x-\mu_i)))$$

$$\hat{c}=arg\ \max_{i \in Classes} \log(\frac{1}{|\Sigma_i|^{1/2}})+\log(\exp((-1/2)(x-\mu_i)^t*\Sigma_i^{-1}*(x-\mu_i)))$$

$$\hat{c}=arg\ \max_{i \in Classes} -\log(|\Sigma_i|^{1/2})-(1/2)[(x-\mu_i)^t*\Sigma_i^{-1}*(x-\mu_i)]$$

$$\hat{c}=arg\ \max_{i \in Classes} -(1/2)\log(|\Sigma_i|)-(1/2)[(x-\mu_i)^t*\Sigma_i^{-1}*(x-\mu_i)]$$

La fonction y=-(1/2)x étant décroissante et continue sur R, en divisant la formule par –1/2 on obtient :

$$\hat{c}=arg\ \min_{i \in Classes} \log(|\Sigma_i|)+[(x-\mu_i)^t*\Sigma_i^{-1}*(x-\mu_i)]$$

C'est cette formule que l'on va appliquer dans le TP3 pour simplifier les calculs.

3.Classification : décision par maximum de vraisemblance

On obtient bien les résultats escomptés :

```
La donnee de classe 1 a ete reconnue comme une donnee de classe  1
Vraisemblances :  [0.93463494881375309, 21.894674407870522, 9.8778387519715753]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2
Vraisemblances :  [42.244538286117319, 3.3027127914791685, 27.220929179739468]
La donnee de classe 3 a ete reconnue comme une donnee de classe  3
Vraisemblances :  [11.698487597238886, 6.1313311763569915, -0.027379106581800317
]
```

Ici on prend le argmin des vraisemblances, car on a procédé comme indiqué au dessus dans le 2.

4.Probabilité à priori

En utilisant maintenant des probabilités différentes pour chaque classe, le terme $P(M_i)$ ne peut pas être simplifié. Reprenons donc les calculs à partir de là :

$$\hat{c}=arg\ \max_{i \in Classes} P(x|M_i)P(M_i)$$

$$\hat{c}=arg\ \max_{i \in Classes} P(x|N(\mu_i,\Sigma_i))P(M_i)$$

$$\hat{c}=arg\ \max_{i \in Classes} ([\frac{1}{(2\Pi)^{d/2}|\Sigma_i|^{1/2}}\exp((-1/2)(x-\mu_i)^t*\Sigma_i^{-1}*(x-\mu_i))]P(M_i))$$

En simplifiant les termes constants et en passant au log de la même manière que ci-dessus, on obtient :

$$\hat{c}=arg\ \max_{i \in Classes} \log(\frac{1}{|\Sigma_i|^{1/2}})+\log(\exp((-1/2)(x-\mu_i)^t*\Sigma_i^{-1}*(x-\mu_i)))+\log(P(M_i))$$

$$\hat{c} = arg \max_{i \in Classes} -(1/2)\log(|\Sigma_i|) - (1/2)[(x-\mu_i)^t * \Sigma_i^{-1} * (x-\mu_i)] + \log(P(M_i))$$

En divisant par -(1/2) ie multiplier par -2 on obtient au final :

$$\hat{c} = arg \min_{i \in Classes} \log(|\Sigma_i|) + [(x-\mu_i)^t * \Sigma_i^{-1} * (x-\mu_i)] - 2\log(P(M_i))$$

On a donc le argmin du même résultat que précédemment auquel on enlève 2 fois le log de la probabilité de la classe.

Tests:

Avec les données initiales, les vecteur moyennes étant :

[-0.08334193 -0.08541702] pour la classe 1

(données sûrement généré à l'origine par une loi normale centrée sur (0,0))

[2.93982437 3.08713358] pour la classe 2

(données sûrement généré à l'origine par une loi normale centrée sur (3,3))

et [3.05691106 -0.00513458] pour la classe 3

(données sûrement généré à l'origine par une loi normale centrée sur (3,0))

Et les données tests étant très proche de ces moyennes,

donnee_test_classe1 = [0.18322726, -1.0297674999999999]

donnee_test_classe2 = [3.5376671000000002, 4.8338850000000004]

donnee_test_classe3 = [3.2696486, 0.49428706]

Il faut vraiment appliquer des gros coefficients pour les faire «changer» de décision.

Prenons comme probabilité

Probaclasses=[98./100.,1./100.,1./100.]

(ie 98% pour C1, 1% pour C2, 98% pour C3)

On obtient:

```
La donnee de classe 1 a ete reconnue comme une donnee de classe  1
Vraisemblances :  [0.93463494881375309, 21.894674407870522, 9.8778387519715753]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2
Vraisemblances :  [42.244538286117319, 3.3027127914791685, 27.220929179739468]
La donnee de classe 3 a ete reconnue comme une donnee de classe  3
Vraisemblances :  [11.698487597238886, 6.1313311763569915, -0.027379106581800317
]
La donnee de classe 1 a ete reconnue a priori comme une donnee de classe  1
Vraisemblances :  [0.97504036344879197, 31.105014779846705, 19.088179123947757]
La donnee de classe 2 a ete reconnue a priori comme une donnee de classe  2
Vraisemblances :  [42.284943700752358, 12.51305316345535, 36.431269551715651]
La donnee de classe 3 a ete reconnue a priori comme une donnee de classe  3
Vraisemblances :  [11.738893011873925, 15.341671548333174, 9.1829612653943808]
```

Les données sont toujours identifiées comme étant dans les mêmes classes, les probabilités ne sont pas assez différentes pour changer la décision d'un de ces points. (Ils sont vraiment près des moyennes de leurs classes)

On remarque cependant que les vraisemblances avec la classe C1 ont peut changé (en effet 98% c'est prêt de 1, et log(1) = 0, donc en enlevant (ou ajoutant) 2* le log ça change pas beaucoup)

Alors que les vraisemblances des autres classes ont augmenté.

Attention, le point appartient à la classe avec la plus petite « vraisemblance ».

Passons maintenant aux probabilités

Probaclasses=[9998./10000.,1./10000.,1./10000.]

On obtient :

```
La donnee de classe 1 a ete reconnue a priori comme une donnee de classe  1
Vraisemblances :  [0.9350349888190872, 40.315355151822885, 28.298519495923941]
La donnee de classe 2 a ete reconnue a priori comme une donnee de classe  2
Vraisemblances :  [42.24493832612265, 21.723393535431534, 45.641609923691831]
La donnee de classe 3 a ete reconnue a priori comme une donnee de classe  1
Vraisemblances :  [11.698887637244219, 24.552011920309354, 18.393301637370563]
```

Cette fois ci, la donnée de la classe 3 a basculé dans la classe 1. La donnée de la classe 2 est toujours bien identifié. On observe la même chose que précédemment, les scores de la classe 1 ne varient pas et ceux des autres classes augmentent.

On peut en conclure qu'une classe a un « rayon d'attraction » très fort vers sa moyenne mais qui diminue vite avec la distance.

5.Généralisation

On va ici générer des données de 4 classes en dimension 3.

Prenons comme centres (0.,0.,0.) ,(3.,3.,3.),(0.,0.,3) et (0.,3.,0.)

prenons comme écart-type 2 pour chaque loi sur chaque coordonnés.

Nous allons donc utiliser la fonction « normal » du module numpy.normal pour générer ces données.

Les données son générés à chaque appel de la fonction dans la fonction genere_donnee(). On aura donc jamais 2 fois les même résultats.

J'ai pris comme points tests (0.,0.,0.) ,(3.,3.,3.),(0.,0.,3) et (0.,3.,0.)

Pour une exécution, j'obtiens le résultat suivant :

```
moyenne= [-0.06890084 -0.0304742  -0.5285495 ]
moyenne= [ 3.00145654  2.68379248  2.76944133]
moyenne= [-0.18446923 -0.02738854  3.13591418]
moyenne= [-0.08636705  3.42447817 -0.13365268]
La donnee de classe 1 a ete reconnue comme une donnee de classe  1
Vraisemblances :  [4.3652635480499304, 9.6664390168200267, 7.0852001576549437, 6.8216694174157642]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2
Vraisemblances :  [11.378463362212553, 3.8273291263255236, 8.1613385195284938, 9.3800740488034791]
La donnee de classe 3 a ete reconnue comme une donnee de classe  3
Vraisemblances :  [7.8745145605953955, 8.0323542964129242, 4.3187420250294712, 9.4393818816971411]
La donnee de classe 4 a ete reconnue comme une donnee de classe  4
Vraisemblances :  [6.1651015583625215, 8.3251599809177854, 8.9519259195108365, 4.1804777488174381]
```

6.Optimisation

D'après wikipedia,

Une matrice diagonale est inversible si et seulement si **son déterminant est non nul**, c'est-à-dire si et seulement si tous ses éléments diagonaux sont non nuls. Dans ce cas, l'inverse d'une matrice diagonale est une matrice diagonale où les coefficients diagonaux sont les inverses des coefficients diagonaux de la matrice de départ.

Les coefficients situés sur la diagonales sont les variances (l'écart type au carrée) de chaque classe.

Si la variance d'une classe est nulle, c'est que toutes les distances des points à la moyenne sont nulles (par définition), donc que tous les points de la classe sont la moyenne, ce qui est peu probable.

On peut donc mettre en hypothèse que tous les coeffs diagonaux sont non nuls et donc que le déterminant de la matrice de variance-covariance est non nulle.

En ce cas là, det(∑) = var(X1) * var(X2)...* var(X3) = $\sigma_{X1}^2 * \sigma_{X2}^2 * \ldots * \sigma_{Xd}^2$

et

$$[(x-\mu_i)^t * \Sigma_i^{-1} * (x-\mu_i)]$$

devient :

$$\frac{(x_1-\mu_{i,1})^2}{var(X^1)} + \frac{(x_2-\mu_{i,2})^2}{var(X^2)} + \ldots$$

On a donc :

$$\hat{c} = arg \min_{i \in Classes} \log(var(X1_i) * var(X2)_i * \ldots) + \frac{(x_1-\mu_{i,1})^2}{var(X^1)} + \frac{(x_2-\mu_{i,2})^2}{var(X^2)} + \ldots$$

Pour optimiser les calculs, on va donc n'utiliser qu'un vecteur de variances au lieu de la matrice de covariance.

Avec les valeurs de tests standard données au début, on a des résultats très similaires :

```
La donnee de classe 1 a ete reconnue comme une donnee de classe  1
Vraisemblances :  [0.96335972872599929, 21.417977641931262, 10.742090073090608]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2
Vraisemblances :  [39.684421756207783, 3.2504134818586601, 27.738491386427881]
La donnee de classe 3 a ete reconnue comme une donnee de classe  3
Vraisemblances :  [11.439790073056649, 6.2139753571435961, 0.002637977729866403
1]
```

Contact

Si il y a des erreurs, des remarques, des ajouts à faire, etc.

Veuillez m'en faire part à cette adresse :

wedg@hotmail.fr