Apprentissage Automatique Numérique

M1 Informatique Miniprojet 1

L'objectif de ce mini-projet est de programmer un classifieur bayésien. L'exemple choisi est la classification de très beau specimens d'iris (la fleur). Chaque iris est représentée par 4 paramètres caractéristiques : la largeur et la longueur de leur pétale et de leur sépale. Chaque iris est donc codé par un vecteur de réels de dimension 4.

Le travail demandé consiste en deux phases :
— Apprentissage des paramètres du classifieur bayésien avec les données d'apprentissage
— Détermination des paramètres optimaux du classifieur avec les données de développement
— L'utilisation du classifieur pour tester ces performances sur les données de test

Les instructions Python suivantes permettent de charger le jeu de données Iris, et d'afficher la partie data (description des données en termes d'attributs) et la partie target (classe, cible, étiquette)

```
from sklearn import datasets
iris = datasets.load_iris()
print (iris.data)
print (iris.target)
```

1 Division de l'échantillon d'apprentissage

Étant donné que les données sont triées par classe, il faut d'abord les mélanger.

```
import numpy as np
Ciris = np.c_[iris.data.reshape(len(iris.data), -1), iris.target.reshape(len(iris.target), -1)]
np.random.seed(987654321)
np.random.shuffle(Ciris)
shuffledIrisData = Ciris[ :, :iris.data.size//len(iris.data)].reshape(iris.data.shape)
shuffledIrisTarget = Ciris[ :, iris.data.size//len(iris.data) :].reshape(iris.target.shape)
```

Utilisez la même valeur dans np.random.seed si vous voulez obtenir les mêmes corpus.

Pour assurer une comparaison correcte entre les différents classifieur nous divisons les données en trois parties : apprentissages, dev et test :
— Apprentissage : 100 premiers exemples
— Dev : 30 exemples suivants
— Test : 20 derniers exemples.

Dans un premier temps nous travaillerons uniquement avec deux dimensions décrivant nos fleurs.

Pour rappel, voici les étapes à réaliser ;

1. Utilisation des données d'apprentissage et leur classe afin de déterminer les paramètres de votre classifieur Bayésien.
2. on peut maintenant utiliser le système pour classer des données. Vous calculez le taux d'erreur de votre système sur les données développement puisque vous disposez des classes correctes pour cet ensemble.
3. Vous pouvez éventuellement répéter les étapes 1 et 2 pour créer différentes variantes de votre système. Vous garderez bien sûr la variante dont le taux d'erreur est minimal.
4. Vous appliquez votre meilleur système sur les données de Test.

Ces trois étapes sont détaillées par la suite.

2 Phase d'apprentissage

Il faut estimer deux types de probabilités :

— Les probabilités à priori $p(\omega_i)$ pour $i = \{0, 1, 2\}$.

— Les vraisemblances (probabilités conditionnelles) $P(x|\omega_i)$. On suppose que celles-ci suivent une loi Gaussienne de dimension 2. Il s'agit donc d'obtenir la moyenne μ_i (vecteur de dimension 2) et la matrice de variance-covariance Σ_i (matrice de dimension 2×2), **séparément** pour chaque classe ω_i.

Rappel : fonction de densité de probabilité pour une loi Gaussienne multi-dimensionnelle de taille N :

$$p(x) = \frac{1}{(2 \cdot \pi)^{\frac{N}{2}} \|\Sigma\|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^t \Sigma^{-1}(\mathbf{x}-\mu)}$$

Une fois que toutes ces valeurs sont calculées, on peut les être utiliser dans la phase de classification.

3 Phase de développement

On traite séquentiellement tous les exemples du corpus de développement. Pour chaque exemple on calcule la probabilité *a posteriori* $P(\omega_i|x) = P(x|\omega_i)p(\omega_i)$ avec les paramètres déterminés lors de la phase d'apprentissage, et ceci pour toutes les 3 classes $i = 0..9$. La classe pour laquelle la probabilité *a posteriori* est maximum est la classe reconnue. Puisque vous connaissez la bonne classe pour chaque exemple du corpus de développement, vous pouvez compter le nombre d'erreurs.

Comparer les performances de plusieurs variantes de votre classifieur.

4 Phase d'évaluation

Vous utiliserez votre meilleur système pour classer tous les exemples du corpus de Test de la même manière que ci-dessus. Les résultats calculés sur le corpus de Test ne doivent pas être utilisés pour déterminer votre meilleur système.

5 Travail à rendre

— Un Notebook contenant

1. Le descriptif des étapes suivies pour obtenir votre meilleur système
2. La matrice de confusion de votre meilleur système (les confusions faites par votre meilleur classifieur).