TP3 – Performance d'un classifieur

On travaille sur des images de la base Pratheepan Dataset (http://cs-chan.com/ downloads_skin_ dataset.html) dont on souhaite détecter les pixels de teinte chaire afin d'accélérer un algorithme de détection de visages. Pour cela, la vérité de terrain a été déterminée pour chaque image. On a récupéré dans X_train, y_train, les pixels des 26 premières images et dans X_test, y_test, ceux des 4 dernières images. Seul un pixel sur 2000 a été conservé pour avoir des temps de calcul raisonnables. X_ est composé des composantes chromatiques Cb, Cr de chaque pixel et y_ est composé de la classe du pixel, 1 si couleur peau, 0 sinon.

I. Chargement et visualisation des données

Observer les images dans l'explorateur de fichier puis charger les données et visualiser les points de chaque classe avec :

```
[X_train, y_train, X_test, y_test] = np.load("TP3.npy",allow_pickle=True)
#Pixel peau
T_train = X_train[np.where(y_train==1),:]
T_train = np.reshape(T_train,(T_train.shape[1],T_train.shape[2] ))
#Pixel non peau
F_train = X_train[np.where(y_train==0),:]
F_train = np.reshape(F_train,(F_train.shape[1],F_train.shape[2] ))

plt.plot(F_train[:,0], F_train[:,1], '.b')
plt.show
plt.plot(T_train[:,0], T_train[:,1], '.r')
plt.show
```

Questions

Combien y a-t-il de pixels de teinte chaire ? de teinte non chaire ? Quelle est la dimension des données ?

II. Modélisation de la densité de probabilité a priori de la teinte chaire par une loi normale 2D avec des dimensions décorrélées

Comme il est difficile de modéliser tout ce qui n'est pas teinte chaire, on décide de travailler avec une seule classe, la teinte chaire, dont on estime la densité de probabilité *a priori*. $p(\boldsymbol{x}/chair)$.

a. Estimation de la densité de probabilité a priori des pixels de teinte chaire

On modélise la densité de probabilité *a priori* des pixels de teinte chaire $p(\boldsymbol{x}/chair), \boldsymbol{x} = \begin{bmatrix} Cb \\ Cr \end{bmatrix}$ sur la base d'apprentissage par une loi normale 2D avec des dimensions décorrélées :

$$p(\boldsymbol{x}/chair) = \frac{1}{\sqrt{2\pi}\sigma_{Cb}} exp\left(-\frac{(Cb - m_{Cb})^2}{2\sigma_{Cb}^2}\right) * \frac{1}{\sqrt{2\pi}\sigma_{Cr}} exp\left(-\frac{(Cr - m_{Cr})^2}{2\sigma_{Cr}^2}\right)$$

Où m_{Cb}, m_{Cr}, σ_{Cb}, σ_{Cr} représentent les moyennes et écarts-type de chacune des composantes.

Déterminer m_{Cb}, m_{Cr}, σ_{Cb}, σ_{Cr}.

Utiliser la fonction norm1(x, m, σ) pour estimer la probabilité de tous les pixels de la base d'apprentissage X_train d'appartenir à la teinte chaire :

$p\left(\begin{bmatrix}Cb\\Cr\end{bmatrix}/chair\right)$ =norm1(Cb, m_{Cb}, σ_{Cb}) * norm1(Cr, m_{Cr}, σ_{Cr})

On stockera ces valeurs dans un vecteur p1_train.

```
def norm1(x, m, s):
    p =1/(math.sqrt(2*math.pi)*s)*math.exp(-(x-m)*(x-m)/(2*s*s))
    return p
```

Questions
Quelle est la dimension de m_{Cb}, m_{Cr}, σ_{Cb}, σ_{Cr} ?
Pour un pixel $\boldsymbol{x}$ de teinte chaire donnée, quelle est la dimension de $p(\boldsymbol{x}/chair)$?
Quelle est la dimension du vecteur p1_train ?
Quelle hypothèse nous permet d'estimer la valeur de la loi normale à partir de l'équation précédente ?

b. Classification

Afin de classer les pixels de test X_test comme teinte chaire ou non, on estime la valeur de $p(\boldsymbol{x}/chair)$ en ces points que l'on stocke dans P1_test. Pour réaliser la classification, on seuille P1_test en utilisant comme seuil initial la valeur moyenne de P1_train.

Sur la base de test, estimer :

- TP : le nombre de vrai positif
- TN : le nombre de vrai négatif
- FP : le nombre de faux positif
- FN : le nombre de faux négatif

puis la sensibilité et la spécificité. Quel est le taux de bonne reconnaissance ?

Questions
Que représentent TP, TN, FP et FN ? Comment estimer la sensibilité et la spécificité ? Comment estimer le taux de bonne classification ? Pourquoi avoir choisi ce seuil initial ?

c. Courbe ROC

Plutôt que de choisir un seuil arbitraire, on choisit 20 valeurs de seuils régulièrement réparties entre min(P1_train) et max(P1_train). Pour chaque valeur de seuil, estimer la précision et le rappel et tracer la courbe ROC.

```
NB = 20
step = (np.max(p1_train) - np.min(p1_train) ) / NB
SEUILS = np.arange (np.min(p1_train), np.max(p1_train), step)
```

Questions
Déterminer sur la courbe ROC le point de fonctionnement tel qu'il y ait autant de faux positifs que de faux négatifs. Que vaut le taux de reconnaissance pour ce point ?

III. Modélisation de la densité de probabilité a priori de la teinte chaire par une loi normale 2D

Reprendre la partie II en utilisent une loi normale 2D définie par :

$$p(\boldsymbol{x}) = \frac{1}{2\pi \det(\boldsymbol{\Sigma})} exp(-0.5(\boldsymbol{x}-\boldsymbol{m})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{m}))$$

Où $\boldsymbol{m}$ et Σ sont les moyenne et matrice de covariance estimées sur les pixels de peau de la base d'apprentissage. On appellera norm2($\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{\Sigma}$) la fonction qui renvoie la probabilité en un point $\boldsymbol{x}$.

```
def norm2(x, m, cov):
        a = np.dot(np.transpose((x-m)), np.linalg.inv(cov))
        a = np.dot(a, (x-m))
        p =1/(math.sqrt(2*math.pi*np.linalg.det(cov)))*math.exp(-0.5*a)
        return p
```

Tracer la nouvelle courbe ROC et comparer les deux modélisations de la densité de probabilité.

Questions
Quelle méthode amène aux meilleurs résultats ?