

## TP4 – Classification bayésienne

Dans ce TP, nous souhaitons résoudre un problème de classification binaire en utilisant la classification bayésienne. Chaque classe va être modélisée en utilisant la méthode « Kernel Density Estimation (KDE) »

### I. Chargement et visualisation des données

Charger les données (TP4.npy) et visualiser les en utilisant le code donné au TP3.

### II. Estimation des densités de probabilité

Ecrire une fonction `ddp_parzen(x, X, Cov)` qui estime la densité de probabilité en un point  $x$  à partir des points de la base d'apprentissage  $X$ , en utilisant un noyau 2D gaussien. On utilisera une matrice de covariance diagonale avec un écart type  $\sigma$  identique pour les deux dimensions. On pourra utiliser la fonction `norm2()` développée au TP3. On rappelle que :

$$\text{ddp\_parzen}(x, X, \text{Cov}) = \frac{1}{N} \sum_{i=1}^N \text{norm2}(x, X_i, \text{Cov})$$

#### Questions

Rappeler le principe de fonctionnement de l'estimation de densité de probabilité par noyau. Que représente le noyau ?

Estimer la densité de probabilité (ddp) pour les pixels de teinte chair  $p(x/\text{chair})$  et non chair  $p(x/\text{non chair})$ . Afin de visualiser ces ddp, estimer les en tout point  $x = \begin{bmatrix} Cb \\ Cr \end{bmatrix}$  d'une grille carrée ( $1 \leq Cb, Cr \leq 20$ ) et stocker les dans deux matrices  $p\_chair(Cb, Cr)$  et  $p\_non\_chair(Cb, Cr)$  que l'on pourra visualiser sous forme d'images (attention à retourner l'image avant de la visualiser, `np.flipud`). Visualiser la ddp de chacune des deux classes et faire varier  $\sigma$ .

#### Questions

Comment varient les ddp en fonction de  $\sigma$  ? Etait-ce prévisible ? *A priori* et sans faire le test, quelle valeur de  $\sigma$  pensez-vous adéquate pour faire la classification ?

### III. Classification bayésienne

Estimer les densités *a priori* de chaque classe  $P\_chair$  et  $P\_non\_chair$ .

Réaliser la classification des points de la base de test en utilisant la règle de bayes (on classifiera à tour de rôle chaque élément de la base de test). Afficher le taux de reconnaissance.

Faire varier  $\sigma$  et optimiser le taux de reconnaissance.

#### Questions

Pour quelle valeur de  $\sigma$  obtient-on la meilleure classification ? Etait-ce prévisible ? Expliquer le résultat.