Reconnaissance de langues écrites

Le projet sera envoyé par email (à `christophe.ambroise@univ-evry.fr`) sous forme de fichier `pdf` avec le notebook (`python` ou `Rmd` correspondant). Décrivez succinctement le problème, écrivez les calculs que vous programmez. Le projet est à réaliser en binôme ou seul.

Exercices de base

Une procédure possible pour définir le classifieur consiste à considérer que les densités conditionnelles aux classes appartiennent à une famille de densités définies par peu de paramètres. Cette approche paramétrique comporte alors deux étapes :

1. le choix d'un modèle ;
2. l'estimation des paramètres de ce modèle.

Cette démarche revient à approximer les densités *a posteriori* par

$$\hat{\pi}(k|\boldsymbol{x}) = \frac{\hat{p}_k \cdot f_k(\boldsymbol{x}|\hat{\theta}_k)}{\sum_{\ell=1}^{K} \hat{p}_k \cdot f_\ell(\boldsymbol{x}|\hat{\theta}_\ell)}.$$

Exercice 1 Base de textes

1. Constituer une base de fichiers de textes bruts (au moins 30) $\boldsymbol{x}_i$ dont certains en français $y_i = -1$ et d'autres en anglais ($y_i = 1$). Nous noterons

   $$\mathcal{D} = ((\boldsymbol{x}_i, y_i))_{i=1,\cdots,n}$$

   la base de textes étiquetés par leur langue.
2. À partir de $\mathcal{D}$, construire un tableau $X = (x_{ij})_{i=1,\cdots,n;\ j=1,\cdots,p}$ où $x_{ij} = \log(1 + f_{ij})$, f_{ij} étant la fréquence du symbole j dans le texte i.
3. Représenter pour chacune des deux classes (anglais-français) un histogramme des log-fréquences des symboles. Commentez.

Attention cet exercice est plus complexe qu'il n'y paraît car il faut écrire un fonction de nettoyage et normalisation des textes.

Exercice 2 Classifieur de Bayes naif

Supposons que

$$f_k(\boldsymbol{x}|\theta_k) = \mathcal{N}_p(\boldsymbol{x}; \boldsymbol{\mu}_k, \sigma_k^2 I_p).$$

1. Estimer les paramètres de moyennes et variances des classes.

2. Programmer votre classifieur .

3. Évaluer ses performances par validation croisée.

Exercice 3 Classifieur markovien

Supposons que

$$f_k(\boldsymbol{x}|\theta_k) = \mathcal{MC}(\boldsymbol{x}; \boldsymbol{\pi}_k, A_k)$$

. où $\mathcal{MC}$ est une chaîne de Markov de matrice de transition A_k et de probabilité d'état initial $\boldsymbol{\pi}_k$ sur l'ensemble des symboles.

1. Estimer les paramètres des deux chaines de Markov.

2. Programmer votre classifieur markovien.

3. Évaluer ses performances par validation croisée.

Exercice 4 Décodage de langue par Viterbi

1. Créer un court texte d'au plus 1000 caractères enchainant de manière aléatoire des phrase en français et en anglais tirées de vos textes initiaux

2. Utiliser l'algorithme de Viterbi pour trouver les passages en français et en anglais du texte fabriqué à l'aide des chaines de Markov estimées à l'exercice précédent.

3. Commentez.

Exercices facultatifs

Vous pouvez faire 0 ou 1 exercice facultatif.

Exercice 5 Algorithme de Baum-Welch (***)

1. Programmer un algorithme de Baum-Welch à deux états cachés.

2. Exécuter votre code sur le texte fabriqué à l'exercice précédent (Viterbi) en utilisant comme initialisation les paramètres des chaines utilisées pour Viterbi.

3. Exécuter votre code sur le texte fabriqué à l'exercice précédent (Viterbi) en utilisant comme initialisation les paramètres des chaines utilisées pour Viterbi.

4. Exécuter votre code sur le texte fabriqué à l'exercice précédent (Viterbi) en utilisant une initialisation aléatoire des paramètres.

5. Illustrez et commentez.

Exercice 6 Clustering de textes(**)

1. À partir de la matrice X calculez la distance euclidienne $d(i, j)$ entre tous les textes deux à deux. Nous noterons m_d la médiane des distances calculées.

2. Créer une matrice binaire $K = (\mathbb{I}_{d(i,j)<m_d})_{i,j}$.

3. Utiliser un modèle à blocs stochastiques pour trouver deux classes de textes à partir de la matrice K.