Université de Caen — **Analyse de données avancée**
UFR des Sciences — Examen TP du 11 décembre 2017. Durée 2h — par F. Chamroukhi
M2 Statistique/Informatique — Seuls le cours et les travaux personnels sont autorisés — 2017/2018

Consignes : Vous devez séparer pour chacun des exercices votre programme principal (l'équivalent du main) du reste du code (fonctions etc). Le dépôt de votre travail se fera sous la forme d'une archive à déposer sur ecampus (exclusivement) le 11/12/2017 avant 17h30, heure stricte.

Barème : Le barème est donné à titre indicatif. Les exercices pour lesquels les codes ne s'exécutent pas, seront notés sur la moitié des points du barème.

Ex1 : Analyse supervisée (10 pts) :

On dispose de l'échantillon d'apprentissage datatrain et de l'échantillon de test datatest issus d'une même population de trois classes hétérogènes de données de dimension 2. *Proposer et implémenter* un algorithme pour prédire les classes des données d'un échantillon test issu de cette même population et tester le sur l'échantillon test donné. La règle de prédiction doit être celle du maximum a posteriori (MAP) qui consiste à maximiser la probabilité a posteriori pour prédire la classe y_i de l'individu $\boldsymbol{x}_i$:

$$\hat{y}_i = \arg\max_k \mathbb{P}(y_i = k|\boldsymbol{x}_i; \boldsymbol{\theta}) \tag{1}$$

sur la base donc d'un modèle probabiliste de paramètres $\boldsymbol{\theta}$.
Calculer le taux d'erreur de prédiction pour l'échantillon de test étiqueté `datatest`.
Maintenant considérer l'échantillon AllXtest et prédire ses classes. *Afficher*, sur le même graphique, les données d'apprentissage et les données de test classées de ce nouvel échantillon.

Ex2 : Analyse non-supervisée (6 pts) :

On dispose d'un n-échantillon $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ issu d'une population hétérogène de K classes d'individus binaires multivariés décrits par d variables indépendantes : $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T \in \{0,1\}^d$ et distribués selon une loi mélange de lois de Bernoulli multivariées définie par :

$$\mathbb{P}(\boldsymbol{x}_i; \boldsymbol{\theta}) \quad = \quad \sum_{k=1}^{K} \pi_k \, \mathcal{B}(\boldsymbol{x}_i; \boldsymbol{p}_k) \tag{2}$$

où $\mathcal{B}(\boldsymbol{x}_i; \boldsymbol{p}_k) = \prod_{j=1}^{d} \mathcal{B}(x_{ij}; p_{kj})$ est la loi de Bernoulli multivariée de la classe k avec $\mathcal{B}(x_{ij}; p_{kj})$ la loi de Bernoulli univariée de paramètre $p_{kj} \in]0,1[$ associée à la variable x_{ij} et définie par $\mathcal{B}(x; p) = p^x(1-p)^{1-x}$.

Afin d'apprendre ce modèle en estimant les paramètres $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_K, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_K)$ à partir des données et faire la classification non-supervsiée des données, on utilise l'algorithme Espérance-Maximisation (EM). On montre que pour ce modèle l'algorithme EM consiste à partir d'un modèle initial de paramètres $\boldsymbol{\theta}^{(0)}$ et alterner à chaque itération t entre les deux étapes E- et M- suivantes jusqu'à ce qu'il n'y ait plus d'augmentation significative au sens d'un seuil préfixé de la log-vraisemblance du modèle :

1. Étape E : Calculer les probabilités a posteriori : $\tau_{ik}^{(t)} = \mathbb{P}(y_i = k|\boldsymbol{x}_i, \boldsymbol{\theta}^{(t)}) = \frac{\pi_k^{(t)} \mathcal{B}(\boldsymbol{x}_i; \boldsymbol{p}_k^{(t)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t)} \mathcal{B}(\boldsymbol{x}_i; \boldsymbol{p}_\ell^{(t)})}$

2. Étape M : Mettre à jour les paramètres du modèle $\boldsymbol{\theta}^{(t+1)}$: $\pi_k^{(t+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(t)}}{n}$ et $\boldsymbol{p}_k^{(t+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(t)} \boldsymbol{x}_i}{\sum_{i=1}^{n} \tau_{ik}^{(t)}}$.
Implémenter l'algorithme EM pour estimer les paramètres $\boldsymbol{\theta}$ du modèle (2) et les classes $\mathbf{y} = (y_1, \ldots, y_n)$

des données Xbin sur la base de ce modèle avec la règle du MAP (1). On pourra fixer la valeur de K à 3.

Afficher les valeurs de la log-vraisemblance enregistrée au cours de l'algorithme et les données classées. Pour ce dernier graphique, on utilisera la commande `imagesc(X(sort(y),:))` où `X` sont les données et `y` les classes obtenues.

Ex3 : Classification non-supervisée d'images de chiffres manuscrits (4 pts) :

On dispose de l'échantillon de test MNIST de $n = 10000$ images binarisées de chiffres manuscrits ($d = 28 \times 28 = 784$ pixels par image). Une image est considérée comme étant une réalisation d'une variable aléatoire $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T \in \{0,1\}^d$ binaire multidimensionnelle suivant le mélange de lois (2).

Utiliser l'algorithme EM implémenté précédemment pour classer ces images en K classes ($K \geq 10$). Pour faire face à d'éventuels problèmes numériques, calculer le logarithme du numérateur des probabilités a posteriori dans l'étape E- et utiliser ensuite la fonction log_normalize pour retrouver les valeurs effectives des probabilités a posteriori en passant par l'exponentielle.

Afficher, sur le même graphique (commande `subplot`), les K lois estimées $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K$ sous forme d'images. La commande `imagesc(reshape(toto,28,28)')` permet d'afficher le vecteur `toto` de dimension $d = 784$ sous forme d'une image de dimension 28×28.