Apprentissage et reconnaissance (GIF-4101 / GIF-7005)
Département de génie électrique et de génie informatique
Hiver 2014

UNIVERSITÉ LAVAL

EXAMEN FINAL

Instructions : – Une feuille aide-mémoire recto verso manuscrite est permise ;
– Durée de l'examen : 2 h 50.
Pondération : – Cet examen compte pour $35\,\%$ de la note finale ;
– La note est saturée à $100\,\%$ si le total des points avec bonus excède cette valeur.

Question 1 (20 points sur 100)

Soit un réseau de neurones de type RBF pour deux classes, composé d'une couche cachée de R neurones de type gaussien, suivi d'une couche de sortie d'un neurone avec fonction de transfert linéaire. La valeur de la sortie pour un tel réseau de neurones pour une valeur d'entrée $\mathbf{x}$ est donnée par l'équation suivante,

$$\mathrm{h}(\mathbf{x}) = \sum_{i=1}^{R} w_i \phi_i(\mathbf{x}) + w_0 = \sum_{i=1}^{R} w_i \exp\left[-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2s_i^2}\right] + w_0,$$

où :
— $\mathbf{m}_i$ est la valeur du centre du i-ème neurone gaussien de la couche cachée ;
— s_i est l'étalement du i-ème neurone gaussien ;
— w_i est le poids connectant le i-ème neurone gaussien de la couche cachée au neurone de sortie ;
— w_0 est le poids-biais du neurone de sortie.

Supposons que l'on fixe les étalements s_i à des valeurs prédéterminées et que l'on veut apprendre les valeurs w_i, w_0 et $\mathbf{m}_i$ par descente du gradient, en utilisant comme critère l'erreur quadratique moyenne,

$$E = \frac{1}{2N} \sum_{\mathbf{x}^t \in \mathcal{X}} (e^t)^2 = \frac{1}{2N} \sum_{\mathbf{x}^t \in \mathcal{X}} [r^t - \mathrm{h}(\mathbf{x}^t)]^2,$$

où :
— r^t est la valeur désirée pour le neurone de sortie du réseau ;
— $\mathcal{X}$ est l'ensemble des N données d'entraînement.

(10) (a) Développez les équations permettant de mettre à jour les poids w_i et w_0 du neurone de sortie par descente du gradient, en utilisant le critère de l'erreur quadratique moyenne.

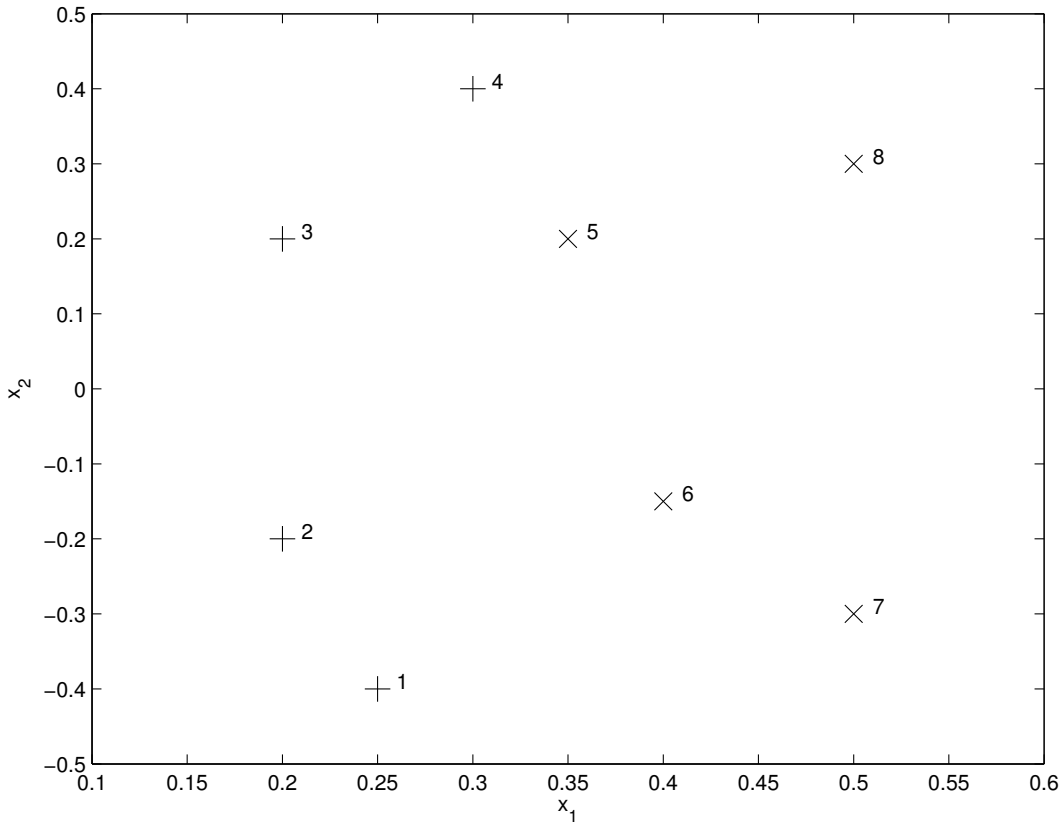
(10) (b) Développez les équations permettant de mettre à jour les valeurs des centres $\mathbf{m}_i$ des neurones gaussiens de la couche cachée par descente du gradient, en utilisant le critère de l'erreur quadratique moyenne.

Question 2 (22 points sur 100)

Soit le jeu de données suivant, en deux dimensions :

$$\mathbf{x}^1 = [0{,}25 \;\; -0{,}4]^T, \quad \mathbf{x}^2 = [0{,}2 \;\; -0{,}2]^T, \quad \mathbf{x}^3 = [0{,}2 \;\; 0{,}2]^T, \quad \mathbf{x}^4 = [0{,}3 \;\; 0{,}4]^T,$$
$$\mathbf{x}^5 = [0{,}35 \;\; 0{,}2]^T, \quad \mathbf{x}^6 = [0{,}4 \;\; -0{,}15]^T, \quad \mathbf{x}^7 = [0{,}5 \;\; -0{,}3]^T, \quad \mathbf{x}^8 = [0{,}5 \;\; 0{,}3]^T.$$

Les étiquettes de ces données sont $r^1 = r^2 = r^3 = r^4 = -1$ et $r^5 = r^6 = r^7 = r^8 = 1$.
Le graphique ici bas présente le tracé de ces données.



Nous obtenons le résultat suivant en effectuant l'entraînement d'un SVM linéaire à **marge douce** avec ces données, en utilisant comme valeur de paramètre de régularisation $C = 300$:

$$\alpha^1 = 73{,}541, \quad \alpha^2 = 0, \quad \alpha^3 = 0, \quad \alpha^4 = 226{,}459, \quad \alpha^5 = 300, \quad \alpha^6 = 0, \quad \alpha^7 = 0, \quad \alpha^8 = 0,$$
$$w_0 = -6{,}136.$$

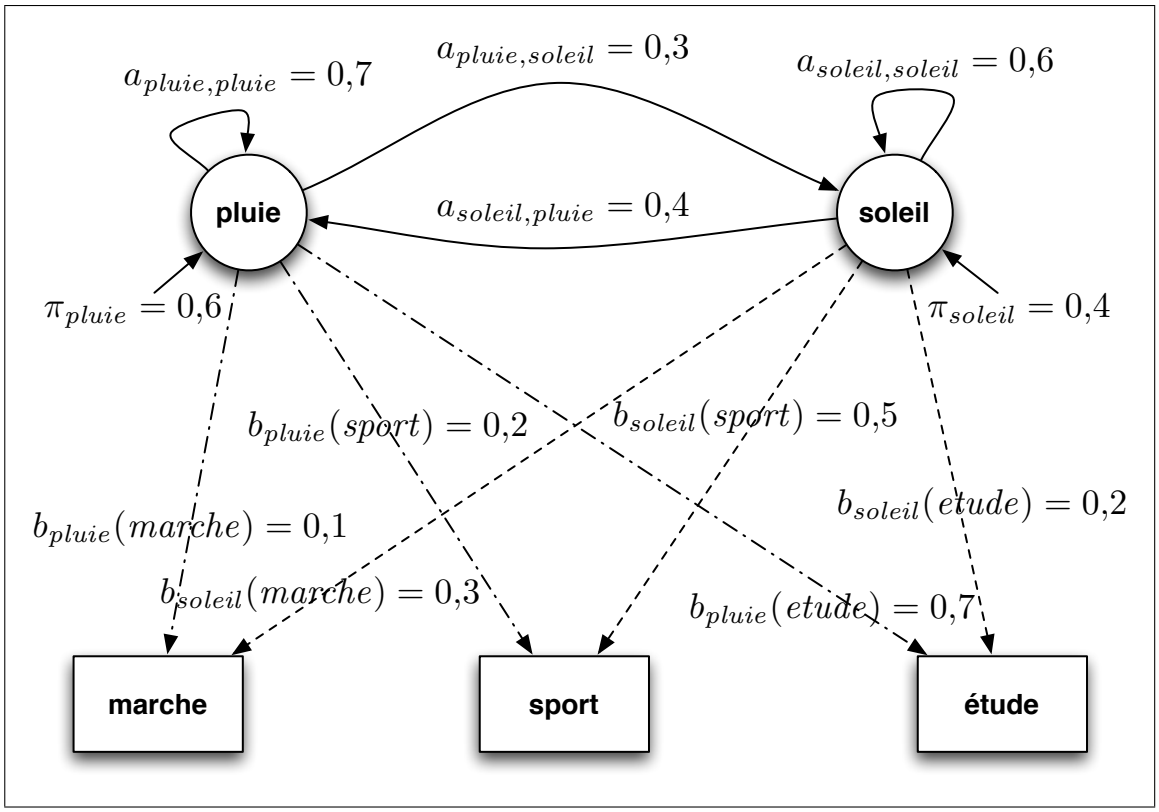
(5) (a) Calculez les valeurs du vecteur $\mathbf{w}$ de l'hyperplan séparateur de ce classifieur.

(12) (b) Déterminez les données qui sont des vecteurs de support (mais pas dans la marge) ainsi que les données qui sont dans la marge ou mal classées. Tracez ensuite un graphique représentant toutes les données du jeu, en encerclant les vecteurs de support et en encadrant les données dans la marge ou mal classées. Tracez également la droite représentant l'hyperplan séparateur ainsi que deux droites pointillées représentant les limites de la marge. **N'utilisez pas** le graphique du préambule de l'énoncé de la question pour donner votre réponse, tracez vous-même un nouveau graphique dans votre cahier de réponse.

(5) (c) Supposons maintenant que l'on veut classer une nouvelle donnée $\mathbf{x} = [0{,}35 \quad -0{,}1]^T$ avec ce SVM. Calculez la valeur $\mathrm{h}(\mathbf{x})$ correspondante (valeur réelle avant seuillage de la sortie).

Question 3 (16 points sur 100)

Soit le modèle de Markov caché (MMC) suivant, correspondant à l'exemple présenté en classe.



(4) (a) Supposons la séquence d'états $S = \{pluie, pluie, soleil, pluie\}$. Calculez la probabilité d'obtenir cette séquence d'états avec le MMC présenté ici haut.

(8) (b) Supposons maintenant une séquence d'observations $O = \{$*étude*,*marche*,*étude*$\}$. Calculez la probabilité d'obtenir cette séquence d'observations avec le MMC présenté ici haut.

(4) (c) Expliquez en quoi consistent les valeurs $\gamma_t^k(i)$ et $\xi_t^k(i,j)$ utilisées par l'algorithme Baum-Welch.

Question 4 (42 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

(3) (a) Expliquez en quoi le classement logistique vu en classe fait un pont, conceptuellement parlant, entre les méthodes génératives et les méthodes discriminatives de classement.

(3) (b) Expliquez pourquoi il est possible de traiter avec succès des données non linéairement séparables avec un discriminant linéaire en utilisant des fonctions de base.

(3) (c) Expliquez pourquoi il est intéressant d'effectuer plusieurs exécutions d'un entraînement avec perceptron multicouches sur un même jeu de données, en utilisant les mêmes hyperparamètres.

(3) (d) Expliquez la relation entre la taille du jeu de données d'entraînement et la complexité algorithmique de méthodes à noyau telles que les séparateurs à vaste marge (SVM).

(3) (e) Donnez le principal avantage et le principal désavantage d'un apprentissage en ligne comparativement à un apprentissage par lots avec une optimisation de classifieurs basée sur la descente du gradient (incluant la rétropropagation des erreurs).

(3) (f) Expliquez ce que l'on entend par la diversité des membres dans un contexte d'ensembles de classifieurs et pourquoi cette diversité est souhaitable.

(3) (g) Complétez la matrice de décision suivante, basée sur un code à correction d'erreurs pour un ensemble de $L = 5$ classifieurs, pour une décision d'ensemble à $K = 3$ classes, afin de maximiser la robustesse de l'ensemble relativement à une erreur des membres :

$$\begin{bmatrix} -1 & -1 & -1 & +1 & ? \\ -1 & +1 & +1 & -1 & ? \\ +1 & -1 & +1 & -1 & ? \end{bmatrix}.$$

(3) (h) Dans l'algorithme AdaBoost présenté en classe, il est indiqué que l'on interrompt l'algorithme lorsque le taux d'erreur ϵ_j est supérieur à $0{,}5$. Expliquez en quoi consiste ce taux d'erreur, en précisant l'équation utilisée pour son calcul.

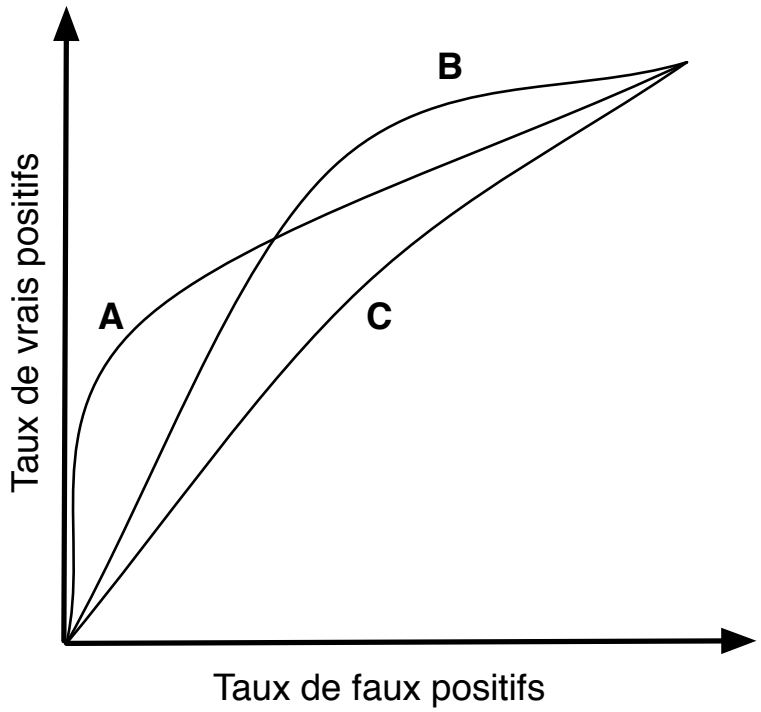
(3) (i) Expliquez la différence principale entre une mixture d'experts et un vote pondéré dans un contexte de méthodes par ensemble.

(3) (j) Expliquez en quoi consiste l'hypothèse H_0 (hypothèse nulle) évaluée par le test statistique ANOVA et de quelle façon on procède pour la tester.

(3) (k) D'un point de vue des processus d'expérimentations, si l'on fait l'entraînement de perceptrons multicouches sur un jeu de données particulier, donnez des exemples de facteurs contrôlables que l'on voudrait étudier.

(3) (l) Expliquez en quoi consiste une validation croisée de type 5×2.

(3) (m) Soit la courbe ROC suivante, présentant les performances de trois classifieurs (A, B et C) opérant selon deux classes (données positives et négatives).



Expliquez en termes clairs et généraux quels classifieurs seraient les plus intéressants à utiliser selon les circonstances rencontrées.

(3) (n) Expliquez pourquoi le classement paramétrique avec lois de probabilité multivariées peut performer relativement bien sur de petits jeux de données, mais performe généralement moins bien que d'autres méthodes sur de grands jeux de données, comparativement à des approches telles que le classement par les k-plus proches voisins et le perceptron multicouche.

Question 5 (15 points bonus)

Supposons que l'on veuille traiter des données provenant selon un flot continu, et qu'il n'est pas possible de stocker ces données de façon permanente. Ces données forment un jeu $\mathcal{X} = \{(\mathbf{x}^1,r^1),(\mathbf{x}^2,r^2),\ldots)\}$, où la paire $(\mathbf{x}^t,r^t)$ reçue au temps t comporte la mesure $\mathbf{x}^t$ et la classe correspondante r^t. On suppose que les données sont indépendantes et identiquement distribuées (iid), donc qu'elles sont reçues dans un ordre aléatoire relativement aux mesures $\mathbf{x}^t$ et classes r^t. Ceci implique, par exemple, que les données de chacune des classes sont bien réparties dans le temps relativement au flot de données.

On veut utiliser des méthodes de classement capables de faire un apprentissage à la volée de

ces données. On considère utiliser la procédure suivante pour évaluer les performances des différentes méthodes :

1. On reçoit une paire $(\mathbf{x}^t,r^t)$ correspondant à la donnée au temps t ;
2. On évalue la performance de notre classifieur $\mathrm{h}(\cdot|\theta^{t-1})$ de l'itération précédente sur la donnée $(\mathbf{x}^t,r^t)$;
3. On entraîne le classifieur sur cette donnée, ce qui nous donne $\mathrm{h}(\cdot|\theta^t)$;
4. Tant que l'on reçoit de nouvelles données du flot, on retourne à l'étape 1.

À l'étape 2, on calcule le taux d'erreur de classement en ligne selon l'équation suivante :

$$E^t = t - \sum_{j=1}^{t} I(r^j,\mathrm{h}(\mathbf{x}^j|\theta^{j-1})),$$

où :
— $I(a,b)$ est une fonction de perte 0-1 retournant 1 lorsque $a = b$, sinon 0 ;
— $\mathrm{h}(\cdot|\theta^t)$ est le classifieur obtenu après entraînement sur la donnée de l'itération t.
D'après vous, est-ce que cette approche suit une méthodologie qui est valide ? Justifiez clairement et de façon convaincante votre réponse, sans verbiage inutile.