Apprentissage et reconnaissance (GIF-4101 / GIF-7005)
Département de génie électrique et de génie informatique
Automne 2009

UNIVERSITÉ LAVAL

EXAMEN FINAL

Instructions : – Aucune documentation sauf aide mémoire d'une page recto-verso manuscrit
– Ce questionnaire comporte 6 pages et 5 questions
– Durée de l'examen : 2 heure 30 minutes

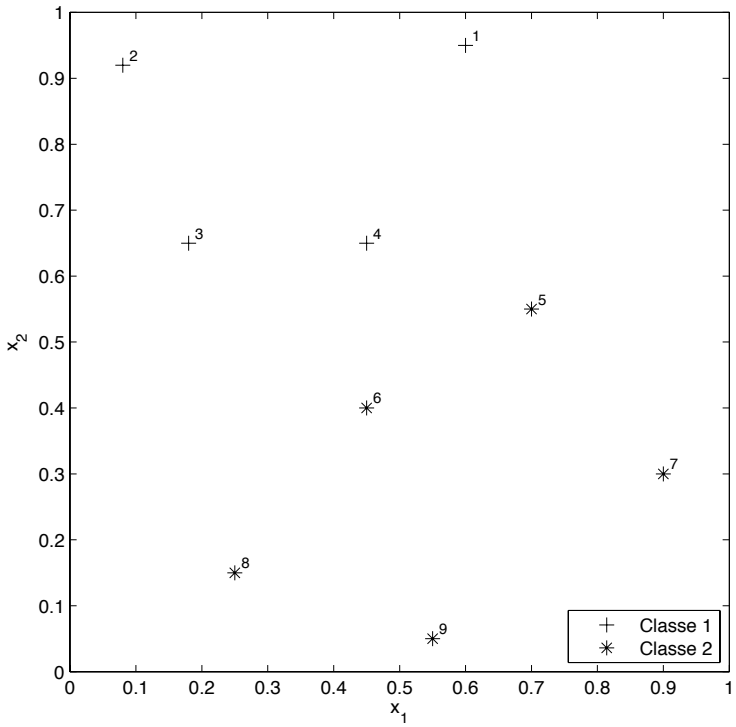
Pondération : Cet examen compte pour 35% de la note finale.

---

1. Soit le jeu de données $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^{9}$ présenté ci-bas.

$$\mathbf{x^1} = \begin{bmatrix} 0.6 \\ 0.95 \end{bmatrix}, \quad r^1 = -1, \quad \mathbf{x^2} = \begin{bmatrix} 0.08 \\ 0.92 \end{bmatrix}, \quad r^2 = -1, \quad \mathbf{x^3} = \begin{bmatrix} 0.18 \\ 0.65 \end{bmatrix}, \quad r^3 = -1,$$
$$\mathbf{x^4} = \begin{bmatrix} 0.45 \\ 0.65 \end{bmatrix}, \quad r^4 = -1, \quad \mathbf{x^5} = \begin{bmatrix} 0.7 \\ 0.55 \end{bmatrix}, \quad r^5 = 1, \quad \mathbf{x^6} = \begin{bmatrix} 0.45 \\ 0.4 \end{bmatrix}, \quad r^6 = 1,$$
$$\mathbf{x^7} = \begin{bmatrix} 0.9 \\ 0.3 \end{bmatrix}, \quad r^7 = 1, \quad \mathbf{x^8} = \begin{bmatrix} 0.25 \\ 0.15 \end{bmatrix}, \quad r^8 = 1, \quad \mathbf{x^9} = \begin{bmatrix} 0.55 \\ 0.05 \end{bmatrix}, \quad r^9 = 1$$

La figure suivante trace ces points en deux dimensions.



(a) (5pts) Supposons que l'on veut classer ces données avec un classifieur de type Séparateur à Vastes Marges (SVM) utilisant un noyau linéaire ($K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$), sans marge floue. Dans votre cahier de réponse, tracez les données du jeu $\mathcal{X}$, les marges géométriques maximales obtenues avec le SVM, l'hyperplan séparateur correspondant, et encerclez les données agissant comme vecteurs de support.

(b) (10pts) Donnez les valeurs des poids $\mathbf{w}$ et biais w_0 correspondant au discriminant linéaire maximisant les marges géométriques tracées en a).

2. Les réseaux *Radial Basis Function* (RBF) sont parfois considérés comme des réseaux de neurones de type perceptron multi-couches. En effet, on peut définir un réseau RBF comme un perceptron multi-couches avec une couche cachée, où chaque neurone sur la couche de sortie utilise une fonction de transfert linéaire, $\mathrm{f}(a) = a$. Sur la couche cachée, on retrouve des neurones d'un type particulier, paramétrés par les variables $\mathbf{m}_i$ et s_i (plutôt que $\mathbf{w}$ et w_0), où la sortie y_j^t du j-ième neurone de la couche cachée pour une donnée $\mathbf{x}^t$ est calculée selon l'équation suivante.

$$y_j^t = \exp\left(-\frac{\|\mathbf{x}^t - \mathbf{m}_j\|^2}{2s_j^2}\right) = \exp\left(-\frac{\sum_{i=1}^{D}(x_i^t - m_{j,i})^2}{2s_j^2}\right)$$

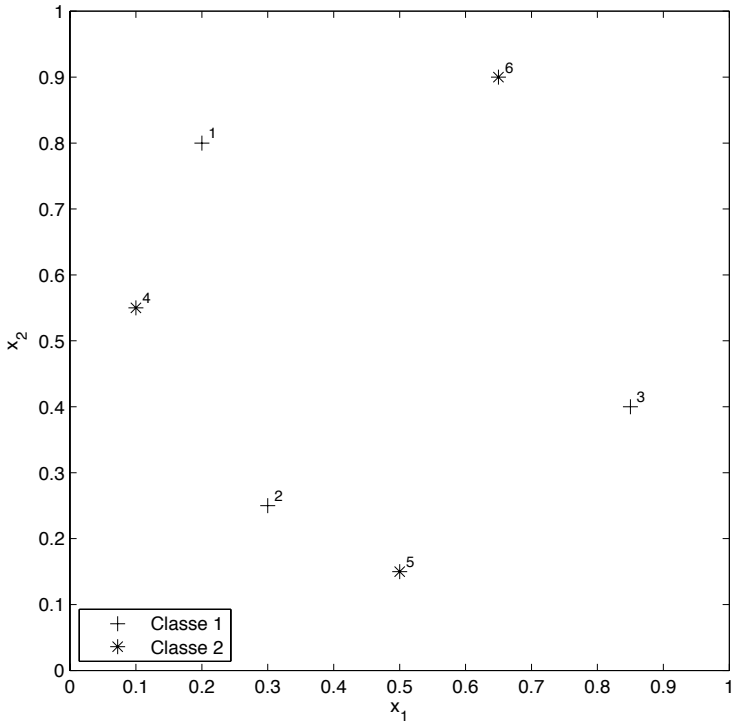
(a) (10pts) Développez les équations de $\Delta w_{j,i},\ j = 1, \ldots, K$, permettant d'effectuer un apprentissage des poids $\mathbf{w}_j$ et biais $w_{j,0}$ de la couche de sortie du réseau RBF à K neurones par une rétropropagation des erreurs.

(b) (15pts) Développez les équations de $\Delta m_{j,i}$ et Δs_j permettant d'effectuer un apprentissage des paramètres $\mathbf{m}_j$ et s_j de la couche cachée de neurones du réseau RBF, par une rétropropagation des erreurs.

3. Soit le jeu de données $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^6$ suivant, comportant deux classes ($r^t \in \{-1, 1\}$).

$$\mathbf{x^1} = \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix}, \quad r^1 = -1, \quad \mathbf{x^2} = \begin{bmatrix} 0.3 \\ 0.25 \end{bmatrix}, \quad r^2 = -1, \quad \mathbf{x^3} = \begin{bmatrix} 0.85 \\ 0.4 \end{bmatrix}, \quad r^3 = -1$$
$$\mathbf{x^4} = \begin{bmatrix} 0.1 \\ 0.55 \end{bmatrix}, \quad r^4 = 1, \quad \mathbf{x^5} = \begin{bmatrix} 0.5 \\ 0.15 \end{bmatrix}, \quad r^5 = 1, \quad \mathbf{x^6} = \begin{bmatrix} 0.65 \\ 0.9 \end{bmatrix}, \quad r^6 = 1$$

La figure suivante trace ces points en deux dimensions.

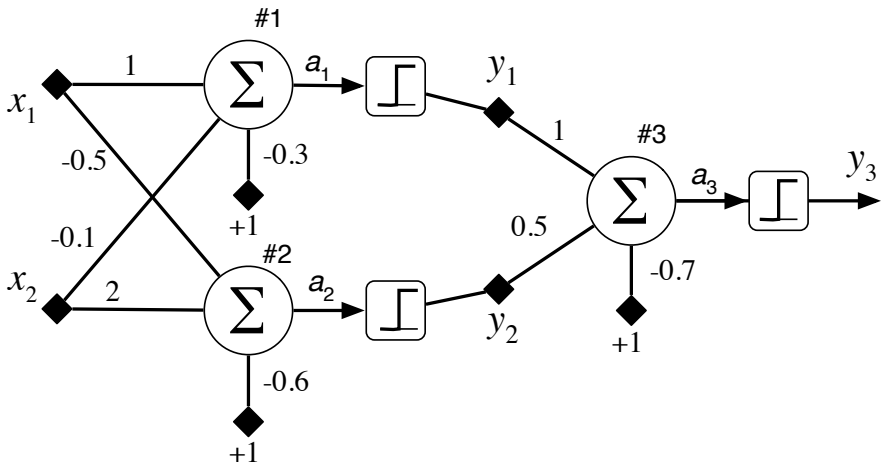


(a) (5pts) Supposons que l'on a entraîné un SVM avec ces données et que l'on a obtenu les valeurs de α^t et w_0 suivantes, en utilisant la matrice de Gram $G(\mathcal{X})$ donnée ci-bas, calculée avec un noyau de type gaussien.

$$G(\mathcal{X}) = \begin{bmatrix} 1.00 & 0.54 & 0.31 & 0.87 & 0.36 & 0.65 \\ 0.54 & 1.00 & 0.52 & 0.77 & 0.90 & 0.34 \\ 0.31 & 0.52 & 1.00 & 0.31 & 0.69 & 0.56 \\ 0.87 & 0.77 & 0.31 & 1.00 & 0.53 & 0.43 \\ 0.36 & 0.90 & 0.69 & 0.53 & 1.00 & 0.31 \\ 0.65 & 0.34 & 0.56 & 0.43 & 0.31 & 1.00 \end{bmatrix}, \ \boldsymbol{\alpha} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}, \ w_0 = -0.0042$$

Effectuez le classement des données de $\mathcal{X}$ selon ces valeurs de α^t et w_0 et donnez le taux d'erreur de classement correspondant.

(b) (5pts) Supposons le réseau de type perceptron multi-couches suivant, avec une couche cachée à deux neurones et une couche de sortie à un neurone.



Les poids $\mathbf{w}$ et biais w_0 des trois neurones du réseau sont les suivants.

$$\mathbf{w}_1 = \begin{bmatrix} 1 \\ -0.1 \end{bmatrix}, \ w_{1,0} = -0.3, \ \mathbf{w}_2 = \begin{bmatrix} -0.5 \\ 2 \end{bmatrix}, \ w_{2,0} = -0.6, \ \mathbf{w}_3 = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \ w_{3,0} = -0.7$$

Les trois neurones du réseau utilisent une fonction de transfert binarisant les valeurs selon la valeur du signe de a_i.

$$\mathrm{f}(a) = \begin{cases} 1 & \text{si } a \geq 0 \\ 0 & \text{autrement} \end{cases}$$

Effectuez le classement de ces données de $\mathcal{X}$ selon cette paramétrisation de perceptron multi-couches et donnez le taux d'erreur de classement correspondant.

(c) (5pts) Le classement par des souches de décision est effectué avec l'équation suivante.

$$\mathrm{h}(\mathbf{x}|\theta, \upsilon, \gamma) = \mathrm{sgn}(\theta(x_\gamma - \upsilon)), \quad \theta \in \{-1, +1\}, \ \gamma \in \{1, \ldots, D\}, \ \upsilon \in \mathbb{R}$$

Soit l'ensemble des trois souches de décisions suivantes, avec les valeurs de paramètre β correspondantes, produits par l'exécution de l'algorithme AdaBoost sur le jeu de données $\mathcal{X}$.

$$\begin{array}{lllll} \text{Souche \#1}: & \theta_1 = +1, & \upsilon_1 = 0.4, & \gamma_1 = 1, & \beta_1 = 0.5 \\ \text{Souche \#2}: & \theta_2 = +1, & \upsilon_2 = 0.475, & \gamma_2 = 2, & \beta_2 = 0.333 \\ \text{Souche \#3}: & \theta_3 = -1, & \upsilon_3 = 0.15, & \gamma_3 = 1, & \beta_3 = 0.5 \end{array}$$

Effectuez le classement des données de $\mathcal{X}$ avec ces trois souches de décisions combinées selon l'algorithme AdaBoost et donnez le taux d'erreur de classement correspondant.

(d) (5pts) Supposons finalement que l'on veut utiliser une mixture d'experts pour classifier les données en utilisant les trois classifieurs précédents (SVM, perceptron multi-couches, AdaBoost avec souches de décision), en ayant une mesure de l'expertise pour chaque données du jeu $\mathcal{X}$, pour chacun des classifieurs ($\mathbf{w}(\mathbf{x}) = [w_{SVM}(\mathbf{x})\ w_{PMC}(\mathbf{x})\ w_{Ada}(\mathbf{x})]^T$).

$$\mathbf{w}(\mathbf{x}^1) = [0.8\ 0.5\ 0.6]^T, \quad \mathbf{w}(\mathbf{x}^2) = [0.9\ 0.5\ 0.7]^T, \quad \mathbf{w}(\mathbf{x}^3) = [0.5\ 0.3\ 0.65]^T$$
$$\mathbf{w}(\mathbf{x}^4) = [0.65\ 0.3\ 0.6]^T, \quad \mathbf{w}(\mathbf{x}^5) = [0.7\ 0.4\ 0.7]^T, \quad \mathbf{w}(\mathbf{x}^6) = [0.65\ 0.45\ 0.7]^T$$

Considérez que les décisions obtenues par les trois classifieurs ont été binarisées au préalable, c'est-à-dire que $\mathrm{h}_{SVM}(\mathbf{x}) \in \{-1, 1\}$, $\mathrm{h}_{PMC}(\mathbf{x}) \in \{-1, 1\}$ et $\mathrm{h}_{Ada}(\mathbf{x}) \in \{-1, 1\}$. Effectuez le classement de ces données de $\mathcal{X}$ selon cette mixture d'experts et donnez le taux d'erreur de classement correspondant.

4. Soit l'algorithme d'apprentissage AdaBoost, présenté dans le pseudo-code suivant.

> 1. Initialiser les probabilités de chaque données, $p_1^t = 1/N,\ t = 1, \ldots, N$
> 2. Pour chaque classifieur de base $j = 1, \ldots, L$:
> 2.1. Échantillonner jeu $\mathcal{X}_j$ à partir de $\mathcal{X}$ selon probabilités p_j^t
> 2.2. Entraîner classifieur h_j avec jeu $\mathcal{X}_j$
> 2.3. Calculer l'erreur du classifieur, $\epsilon_j = \sum_t p_j^t \ell_{0-1}(r^t, \mathrm{h}_j(\mathbf{x}^t))$
> 2.4. *Étape mystère*
> 2.5. Calculer $\beta_j = \frac{\epsilon_j}{1-\epsilon_j}$
> 2.6. Calculer les nouvelles probabilités p_{j+1}^t
>
> $$p_{j+1}^t = \frac{q_j^t}{\sum_s q_j^s}, \quad q_j^t = \begin{cases} \beta_j p_j^t & \text{si } \mathrm{h}_j(\mathbf{x}^t) = r^t \\ p_j^t & \text{autrement} \end{cases}, \quad t = 1, \ldots, N$$

(a) (4pts) Donnez l'énoncé de l'étape 2.4 (étape mystère) de l'algorithme AdaBoost et expliquez pourquoi cette étape est nécessaire.

(b) (3pts) Expliquez de quelle façon les probabilités p_j^t varient dans le temps avec AdaBoost et quelle est l'influence de ces probabilités sur la génération des classifieurs formant l'ensemble.

(c) (3pts) Expliquez le lien qu'il y a entre la maximisation des marges géométriques faite avec les SVM et la maximisation des marges de classement effectuée par AdaBoost.

5. Répondez aussi brièvement et clairement que possible aux questions suivantes.

(a) (2pts) Expliquez pourquoi l'entraînement d'un discriminant linéaire par descente du gradient en utilisant le critère d'erreur du perceptron est peu recommandé comparativement à d'autres approches existantes.

(b) (2pts) Expliquez brièvement les approches *un contre tous* et l'approche *séparation par paires* pour l'apprentissage de données comportant plusieurs classes, lorsqu'on veut utiliser des classifieurs à deux classes tels les discriminants linéaires, SVM ou AdaBoost.

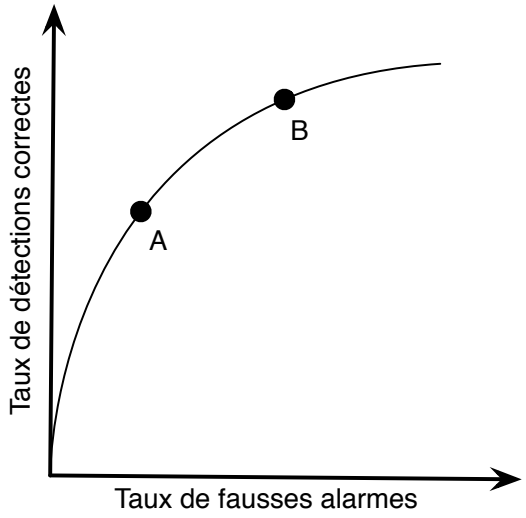
(c) (2pts) Dans l'utilisation de codes à correction d'erreur pour faire du classement avec des ensembles, expliquez brièvement pourquoi on veut maximiser la distance de Hamming (nombre de valeurs différentes) entre les codes de décision de chaque classe.

(d) (2pts) Développez l'équation démontrant que la variance globale d'un ensemble est inférieure à la variance des classifieurs individuels la composant, lorsqu'on fait l'hypothèse que les décisions des classifieurs individuels sont statistiquement indépendantes.

(e) (2pts) Lorsque le nombre de données d'entraînement est très grand, $N \to \infty$, on peut dire que les étiquettes de classe des plus proches voisins à une donnée $\mathbf{x}$ ont une probabilité $P(C_i|\mathbf{x})$ d'appartenir à la classe C_i, où $P(C_i|\mathbf{x})$ est la probabilité *a posteriori* de la véritable densité de probabilité des données. Donnez la probabilité de classer une donnée $\mathbf{x}$ comme appartenant à la classe C_i en fonction de $P(C_i|\mathbf{x})$ lorsqu'on utilise l'algorithme des k-plus proches voisins, en utilisant $k = 3$ voisins et que $N \to \infty$.

(f) (2pts) Combien d'entraînements distincts de classifieur sont fait lorsque l'on veut en évaluer les performances avec la validation croisée 5×2 ?

(g) (2pts) Soit la courbe ROC (*receiver operator characteristics*) présentée ci-bas, décrivant les performance pour la détection d'intrusions de différentes paramétrisation de classifieurs à deux classes.



Expliquez brièvement l'effet occasionné sur les performances du système (intrusions manquées, fausses détections) par le passage de la paramétrisation correspondant point opératoire A sur la courbe ROC à la paramétrisation correspondant au point opératoire B.

(h) (2pts) Le test statistique de l'analyse de variance (ANOVA) peut être utilisé pour déterminer si plusieurs algorithmes de classement ont des performances statistiquement similaires ou non. Ce test fait deux estimations distinctes de σ, soit la variance sur les taux de classement obtenue sur tous les plis et tous les algorithmes testés. Indiquez en quoi diffèrent conceptuellement ces deux estimateurs de σ utilisés pour l'ANOVA.

(i) (2pts) Voici l'équation pour mettre à jour les poids $\mathbf{w}$ de discriminants logistiques.

$$\Delta w_j = \eta \sum_{t=1}^{N} (r^t - y^t) x_j^t$$

Indiquez si cette algorithme correspond à un apprentissage de type *batch* ou de type en-ligne, en justifiant brièvement votre réponse.

(j) (2pts) L'algorithme des K-means en-ligne vise à minimiser l'erreur de reconstruction de K groupes. Cette erreur est présentée dans l'équation suivante.

$$E(\{\mathbf{m}_i\}_{i=1}^{K}|\mathbf{x}^t) = \frac{1}{2}\sum_{i=1}^{K} b_i^t \|\mathbf{x}^t - \mathbf{m}_i\|^2$$

Développez l'équation de $\Delta m_{i,j}$ permettant de mettre à jour les valeurs des positions des centres $\mathbf{m}_i$ de chacun des K groupes par descente du gradient.

(k) (2pts) Dans un processus général de traitement de séquences avec états discrets, on exprime les probabilités de transition de l'état actuel vers l'état S_i au temps t par la suivante.

$$P(s_{t+1} = S_j | s_t = S_i, s_{t-1} = S_k, \ldots, s_1 = S_l)$$

Donnez l'expression simplifiée correspondant à cette probabilité de transition pour un processus de Markov discret.

(l) (2pts) Expliquez en quoi consiste précisemment le problème du décodage avec les modèles de Markov cachés.

(m) (2pts) L'algorithme Baum-Welch permet de faire un apprentissage du modèle λ d'un modèle de Markov caché à partir de plusieurs séquences d'observation. Il a été mentionné que l'algorithme Baum-Welch est une forme spécifique de l'algorithme EM. De ce point de vue, indiquez à quoi correspondent l'étape E et l'étape M avec l'algorithme Baum-Welch.

(n) (2pts) Expliquez brièvement ce que représente la probabilité $\alpha_t(i)$ dans la procédure avant, utilisée pour évaluer la probabilité $P(O|\lambda)$, avec un modèle de Markov caché défini par le modèle λ, d'avoir une séquence d'observations $O = \{o_1, o_2, \ldots, o_T\}$.

(o) (2pts) Expliquez la tâche générale (haut niveau) que l'on veut effectuer avec l'apprentissage par renforcement.

JOYEUSES FETES

18 décembre 2009
CG