Apprentissage et reconnaissance (GIF-4101 / GIF-7005)
Département de génie électrique et de génie informatique
Automne 2011

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 (15 points sur 100)

Une matrice de décision $\mathbf{W}$, de taille $K \times L$, permet de combiner les décisions d'un ensemble de L classifieurs à deux classes, pour faire du classement de données à K classes. L'équation de décision basée sur cette matrice est la suivante :

$$\bar{\mathrm{h}}_i(\mathbf{x}) = \sum_{j=1}^{L} w_{i,j} \mathrm{h}_{j,i}(\mathbf{x}),$$

où :
– $\mathrm{h}_{j,i}(\mathbf{x})$ est le j-ème classifieur de base de l'ensemble ;
– $w_{i,j}$ est l'élément à la position (i,j) dans la matrice de décision $\mathbf{W}$;
– $\bar{\mathrm{h}}_i(\mathbf{x})$ est la décision combinée de l'ensemble pour la classe C_i.

(5) (a) Supposons que l'on veut résoudre un problème à $K = 3$ classes à l'aide d'un ensemble de classifieurs à deux classes combinés selon la méthode *un contre tous* (en anglais, *one against all*). Donnez le nombre de classifieurs à deux classes à utiliser ainsi que la matrice de décision $\mathbf{W}$ correspondant à cette configuration.

(5) (b) Supposons maintenant que l'on veut résoudre ce problème à $K = 3$ classes toujours à l'aide d'un ensemble de classifieurs à deux classes, mais cette fois en combinant les classifieurs selon la méthode de *séparation par paires* (en anglais, *pairwise separation*). Donnez le nombre de classifieurs à deux classes à utiliser ainsi que la matrice de décision $\mathbf{W}$ correspondant à cette configuration.

(5) (c) Finalement, supposons que l'on veut résoudre ce problème à $K = 3$ classes d'un ensemble redondant de $L = 7$ classifieurs, avec un matrice de décision basée sur un code à correction d'erreur (en anglais, *error code output correction*). Donnez la matrice de décision $\mathbf{W}$ correspondant à cette configuration. Déterminez également le nombre d'erreurs de classement des classifieurs de base que cette configuration de système peut tolérer sans se tromper.

Question 2 (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 avec 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 la valeur de l'étalement s_i à une valeur prédéterminée, et que l'on veut apprendre les valeurs w_i, w_0 et $\mathbf{m}_i$ par descente du gradient, en utilisant comme erreur le critère du perceptron,

$$E_{percp} = -\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \, \mathrm{h}(\mathbf{x}^t),$$

où :
- r^t est la valeur désirée pour le neurone de sortie du réseau, soit $r^t = 1$ si $\mathbf{x}^t$ appartient à la classe C_1 et $r^t = -1$ autrement ;
- $\mathcal{Y} = \{\mathbf{x}^t \in \mathcal{X} \,|\, r^t \, \mathrm{h}(\mathbf{x}^t) < 0\}$ est l'ensemble des données $\mathbf{x}^t$ du jeu $\mathcal{X}$ qui sont mal classée par le réseau.

(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 comme erreur le critère du perceptron.

(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 comme erreur le critère du perceptron.

Question 3 (25 points sur 100)

Soit le jeu de données $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^4$ présenté ici-bas.

$$\mathbf{x^1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad r^1 = -1, \qquad \mathbf{x^2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad r^2 = -1, \qquad \mathbf{x^3} = \begin{bmatrix} 0{,}8 \\ 1 \end{bmatrix}, \quad r^3 = 1, \qquad \mathbf{x^4} = \begin{bmatrix} 1{,}5 \\ 1{,}25 \end{bmatrix}, \quad r^4 = 1.$$

Supposons que l'on veut classer ces données avec un classifieur de type Séparateur à vastes marges (SVM) utilisant un noyau linéaire ($K(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}' \rangle = (\mathbf{x}')^T\mathbf{x}$), sans marge floue.

(5) (a) Tracez dans votre cahier de réponse les données du jeu $\mathcal{X}$, l'hyper-plan séparateur que l'on obtient avec un SVM sur ces données, les marges géométriques maximales correspondantes et encerclez les données agissant comme vecteurs de support.

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

(10) (c) Calculez la valeur des α^t correspondant au discriminant linéaire maximisant les marges géométriques calculé en (a).
Considérez que la valeur de w_0 calculée en (b) est toujours valide et n'oubliez pas que les valeurs des α^t doivent respecter la contrainte suivante : $\sum_t \alpha^t r^t = 0$.

Question 4 (40 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

(4) (a) Il a été indiqué dans le cours qu'il y a un lien entre le classement logistique et les méthodes paramètriques. Sous cette perspective, de quelle façon peut-on interpréter la signification de la valeur de la fonction $\mathrm{h}(\mathbf{x})$ d'une fonction de classement logistique traitant des données organisées selon deux classes ?

(4) (b) Avec un Séparateur à vastes marges (SVM) à marge douce, chaque données d'entraînement $\mathbf{x}^t$ a une variable *slack* ξ^t associée. Que signifie une valeur de $0 < \xi^t < 1$ pour une donnée $\mathbf{x}^t$ particulière relativement à son classement par le SVM ?

(4) (c) Avec une analyse en composantes principales (ACP) à noyau, quelle est la taille d'un vecteur propre extrait de la matrice de Gram ?

(4) (d) Indiquez en quoi un jeu de données de validation peut être intéressant pour faire un ajustement d'hyper-paramètres d'un classifieur.

(4) (e) Dans un processus d'expérimentation, on identifie quatre éléments format l'expérimentation : les entrées, les sorties, les facteurs contrôlables et les facteurs incontrôlables. Dans une contexte de classement où une série d'expériences consiste en l'ajustement des hyper-paramètres du classifieur, donnez un exemple de facteurs contrôlables.

(4) (f) On dit souvent que l'utilisation d'une matrice de confusion pour évaluer les performances d'un classifieur permet de s'absoudre des probabilités *a priori* des données (balance entre les classes), comparativement à l'utilisation de l'erreur de classement. Expliquez en vos mots pourquoi ceci est possible.

(4) (g) Dans un perceptron multi-couche avec fonctions de transfert de type sigmoïde pour chaque neurones, il est nécessaire de normaliser les valeur de sortie désirée à des valeurs telles que $\tilde{r}_i^t \in \{0{,}05, 0{,}95\}$. Expliquez pourquoi cette normalisation est nécessaire.

(4) (h) Dans l'algorithme AdaBoost, il est indiqué que l'utilisation de *weak learner* permet d'obtenir de bonnes performances. Expliquez ce qu'est précisément un *weak learner* et spécifiez de quelle façon ce type de classifieur peut être intéressant pour l'algorithme AdaBoost.

(4) (i) Expliquez de quelle façon pourrait-on utiliser une sélection séquentielle vorace avant pour choisir les classifieurs formant un ensemble à partir d'un bassin de classifieurs, dans une approche de type surproduction et sélection.

(4) (j) Pourquoi une génération d'un ensemble de classifieurs par la maximisation de la corrélation négative permet souvent d'obtenir de bonnes performances de classement ?

Question 5 (15 points bonus)

Supposons que l'on veut appliquer l'algorithme Espérance-Maximisation (EM) à un jeu de données en une dimension, où chaque groupe $\mathcal{G}_i$ est décrit par une loi normale $\mathcal{N}(\mu_i,\sigma_i^2)$, soit :

$$p(x|\mathcal{G}_i,\mu_i,\sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right].$$

Donc, la paramétrisation du clustering par EM est donnée par $\Phi = \{\pi_i,\mu_i,\sigma_i\}_{i=1}^K$. En guise de rappel, la formule de l'espérance de vraisemblance de l'algorithme EM est la suivante :

$$\mathcal{Q}(\Phi|\Phi^l) = \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p(x^t|\mathcal{G}_i,\Phi^l).$$

(5 (bonus)) (a) Donnez le développement complet permettant de calculer les estimations π_i des probabilités *a priori* des groupes.

(5 (bonus)) (b) Donnez le développement complet permettant de calculer les estimations m_i des moyennes μ_i.

(5 (bonus)) (c) Donnez le développement complet permettant de calculer les estimations s_i^2 des variances σ_i^2.

Question 6 (10 points bonus)

Dans le cadre du projet final du cours, une équipe d'étudiants a suivi une méthode d'apprentissage leur permettant d'obtenir d'excellentes performances en classement sur la jeu de données de test MNIST à l'aide d'un perceptron multi-couche avec momentum. Voici la méthode proposée par l'équipe :

1. Partitionner le jeu d'entraînement $\mathcal{X}_{\text{train}}$ MNIST en trois jeux de données disjoints de taille comparable, soit $\mathcal{X}_{\text{train1}}$, $\mathcal{X}_{\text{train2}}$ et $\mathcal{X}_{\text{train3}}$;
2. Partitionner le jeu de test $\mathcal{X}_{\text{test}}$ MNIST en trois jeux de données disjoints de taille comparable, soit $\mathcal{X}_{\text{test1}}$, $\mathcal{X}_{\text{test2}}$ et $\mathcal{X}_{\text{test3}}$;
3. Tester plusieurs valeurs de taux d'entraînement η du perceptron multi-couche sur le jeu $\mathcal{X}_{\text{train1}}$, avec un topologie fixe (une couche cachée de 5 neurones) et un momentum nul ($\alpha = 0$), et sélectionner la valeur du taux d'apprentissage permettant de maximiser les performances sur le premier jeu de test, $\mathcal{X}_{\text{test1}}$;
4. Tester plusieurs valeurs de taux de momentum α du perceptron multi-couche sur le jeu $\mathcal{X}_{\text{train2}}$, avec un topologie fixe (une couche cachée de 5 neurones) et le taux d'apprentissage η trouvé à l'étape précédente, et sélectionner la valeur du taux de momentum permettant de maximiser les performances sur le deuxième jeu de test, $\mathcal{X}_{\text{test2}}$;
5. Tester différents nombres de neurones sur la couche cachée du perceptron multi-couche sur le jeu $\mathcal{X}_{\text{train3}}$, avec le taux d'apprentissage η trouvé à l'étape 3 et le taux de momentum α trouvé à l'étape 4, et sélectionner la valeur du nombre de neurones permettant de maximiser les performances sur le troisième jeu de test, $\mathcal{X}_{\text{test3}}$;

6. Faire un nouvel apprentissage avec les valeurs « optimale » du taux d'entraînement η, du taux de momentum α et du nombre de neurones sur la couche cachée trouvés aux trois étapes précédentes avec un nouveau perceptron multi-couche sur le jeu complet d'entraînement $\mathcal{X}_{\text{train}}$ et rapporter les performances de ce classifieur sur le jeu complet de test $\mathcal{X}_{\text{test}}$ comme étant les performances finale du classifieur en généralisation.

D'après vous, est-ce que cette méthodologie est valide afin d'évaluer la capacité de généralisation du classifieur ? Justifiez votre réponse.