FONDEMENTS DE L'APPRENTISSAGE MACHINE (IFT3395/6390)

Professeur: Pascal Vincent

Examen Final

Vendredi 25 avril 2008

Durée: 2h45

Prénom:

Nom:

Code permanent:

IFT 3395 ou 6390?

Veuillez répondre aux questions dans les zones de blanc laissées à cet effet.

Notations

Les notations suivantes sont définies pour tout l'examen, là où elles ont un sens:
On suppose qu'on dispose d'un ensemble de données de n exemples: $D_n = \{z^{(1)}, ..., z^{(n)}\}$. Dans le cas supervisé chaque exemple $z^{(i)}$ est constitué d'une paire *observation, cible:* $z^{(i)} = (x^{(i)}, t^{(i)})$, alors que dans le cas non-supervisé, on n'a pas de notion de cible explicite donc juste un vecteur d'observation: $z^{(i)} = x^{(i)}$.
On suppose que chaque observation est constituée de d traits caractéristiques: $x^{(i)} \in \mathbb{R}^d$: $x^{(i)} = (x_1^{(i)}, ..., x_d^{(i)})$

1 Apprentissage semi-supervisé et par renforcement

1.1 Expliquez brièvement ce qui caractérise l'apprentissage semi-supervisé. Dans quel cas aurait-on avantage à utiliser un algorithme d'apprentissage semi-supervisé plutôt qu'un algorithme d'apprentissage supervisé?

1.2 Expliquez brièvement votre compréhension de ce qui caractérise l'apprentissage par renforcement.

2 Réseaux de neurones en pratique...

Un collègue vous fournit un ensemble de données correspondant à un problème qu'il voudrait automatiser. Vous envisagez d'essayer un réseau de neurones du même genre que ceux vus en cours. Un examen de cet ensemble de données montre qu'il s'agit d'une table de $n = 1000$ exemples qui ressemblent à ceci:

alt	**tai**	**grp1**	**grp2**	**re**
225	0.03	A	1	BON
3800	-0.23	B	3	MAUVAIS
2750	-2.52	A	2	BON
327	1.27	C	1	MAUVAIS
3221	-5.2	C	2	BON
359	1.04	B	2	BON
827	0.22	A	3	MAUVAIS
...	...	...	...	...

Votre collègue vous dit que ce qui l'intéresse c'est de prédire la variable **re** en fonction des autres.

2.1

a) Est-ce un problème d'apprentissage supervisé, non-supervisé, semi-supervisé, ou par renforcement?

b) De quel type de problème précis s'agit-il (classification, régression, estimation de densité, partitionnement, réducion de dimensionalité, etc...)?

c) Pour votre réseau de neurones, combien choisiriez-vous de neurones de sortie?

d) Utiliseriez-vous une non-linéarité de sortie (si oui, laquelle)?

e) Quel type de coût (différentiable) choisiriez-vous pour l'apprentissage?

2.2 En l'absence d'informations plus détaillées sur la nature des entrées, **expliquez en détail quel prétraitement** vous effectueriez pour obtenir un ensemble d'entraînement D_n approprié pour ce réseau de neurones (et la plupart des algorithmes vus en cours d'ailleurs).

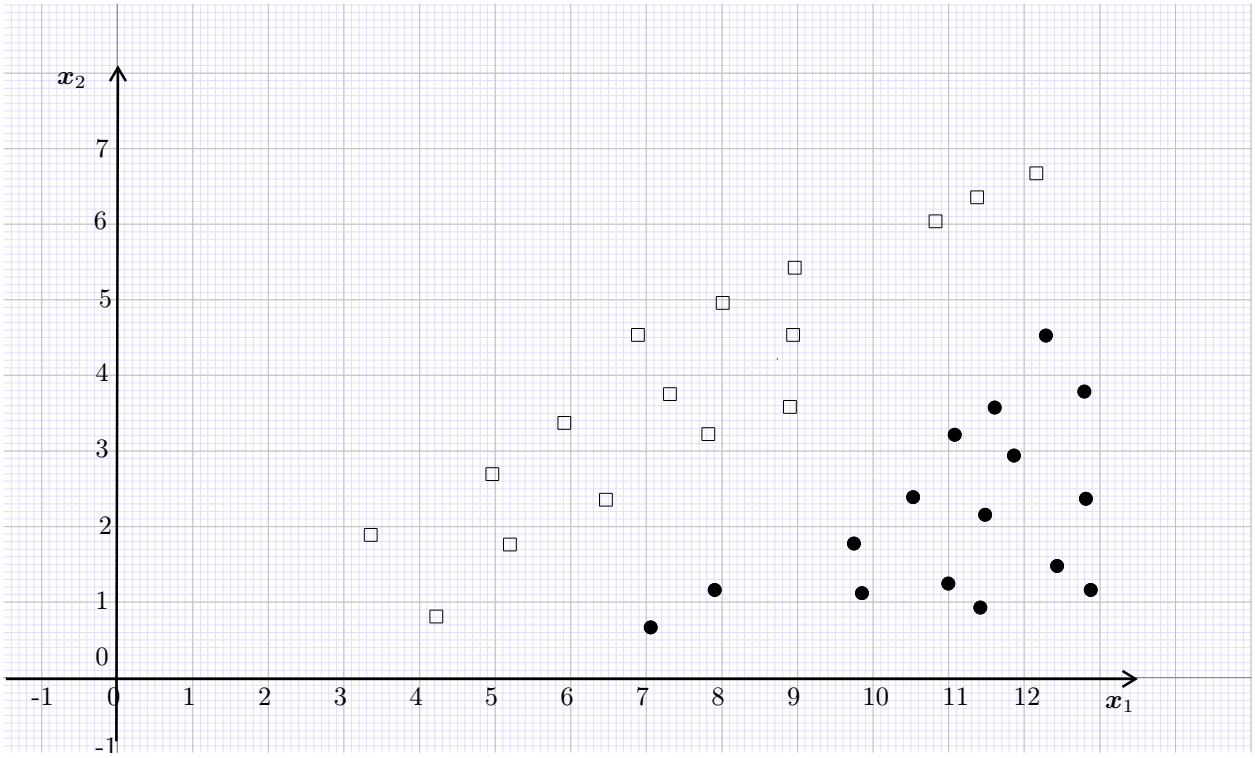
2.3 Écrivez à quoi ressembleraient les 3 premières lignes de l'ensemble d'entraînement D_n obtenu après votre prétraitement (donnez des valeurs approximatives pour les composantes réelles). Quelle est la dimensionalité de l'entrée dans cet ensemble prétraité?

2.4 Dans un modèle de réseau de neurones typique on dispose de plusieurs leviers (hyper-paramètres) et techniques pour *contrôler la capacité* du modèle et ainsi gérer le risque de sur-apprentissage. Quels sont ces leviers et techniques? Nommez-les, expliquez précisément ce qu'ils représentent et, pour chacun, indiquez le sens de l'effet du levier, c.a.d. si le fait de l'**augmenter** va *augmenter la capacité* du modèle (et le risque de sur-apprentissage) ou au contraire *diminuer la capacité* (et augmenter le risque de sous-apprentissage).

2.5 Expliquez, par un pseudo-code informel de haut niveau, la procédure qu'on va typiquement utiliser pour choisir une valeur appropriée de ces leviers, pour un problème donné sous la forme d'un unique ensemble de données D_n.

3 Arbres de décision

Pour un problème de classification binaire (2 classes) en 2 dimension, on donne l'ensemble de données d'entraînement suivant:



On considère un classifieur de type arbre de décision *binaire* classique où chaque noeud n'effectue que la comparaison d'une seule variable d'observation à un seuil choisi (ex. CART). L'arbre est entraîné avec le critère suivant: on choisit des subdivisions qui minimisent la proportion d'erreurs de classification, et ceci jusqu'à obtenir 0 erreurs sur l'ensemble d'entraînement.

a) Sur le graphique ci-dessus, tracez une à une, en les numérotant, les subdivisions de l'espace que réaliserait un tel classifieur. Hachurez la *région de décision* correspondant à la classe des ronds noirs.

b) Dessinez ci-dessous l'*arbre de décision binaire* correspondant, en indiquant sur chaque arc la condition qui doit être vérifiée pour suivre cet arc. Indiquez à l'intérieur de chaque noeud, sous forme d'une paire fractions, la proportion d'exemples de chacune des deux classes qu'on a au niveau de ce noeud. La première fraction de chaque paire devra correspondre à la proportion des carrés, et la deuxième à la proportion des ronds noirs.

c) Exprimez ci-dessous, sous forme d'une règle logique (avec des ET et des OU), la règle représentée par cet arbre qui permet de décider si un point de test $x = (x_1, x_2)$ est de la classe des ronds noirs.

d) Indiquez au moins un point fort des algorithmes de type arbre de décision.

e) Quel est le *principal* point faible des algorithmes de type arbre de décision? Connaissez-vous une technique qui, utilisée en conjonction avec les arbres de décision, permet de contrebalancer ou mitiger ce point faible?

4 Algorithme des K-moyennes (K-means)

4.1 Pour quel type de tâche (de problème) emploie-t-on l'algorithme des K-moyennes? Expliquez dans vos mots à quoi cela peut servir. S'agit-il d'appretissage supervisé, non-supervisé, semi-supervisé ou par renforcement?

4.2 Écrivez l'algorithme des K-moyenes sous forme de pseudo-code, à partir des notations définies plus haut pour tout l'exeman:

ALGORITHME K-MOYENNES (paramètres: D_n, K)

5 Modèles graphiques dirigés

5.1 Dessinez ci-dessous le modèle graphique dirigé (réseau Bayesien) correspondant à la décomposition de la probabilité jointe suivante:

$$P(A,B,C,D,E) \;=\; P(E|A,D,C)\,P(D|C)\;P(C)P(A|C,B)\,P(B)$$

5.2 On veut écrire un programme qui pourra générer des exemples de quintuplets (a,b,c,d,e) tirés aléatoirement de la distribution jointe $P(A,\,B,\,C,\,D,\,E)$ ci-dessus. On suppose que vous avez déjà écrit les fonctions permettant de tirer des exemples selon chacune des distributions situées à droite du $=$. On pourra faire appel par ex. au tirage d'un exemple e de $P(E|A,\,D,\,C)$, sachant $A=a,\ D=d$, et $C=c$, et on notera cette opération $e\sim P(E|A=a,D=d,C=c)$.

En utilisant cette notation, écrivez la procédure pour générer des exemples $(a,\,b,\,c,\,d,\,e)$ de la distribution jointe $P(A,B,C,D,E)$:

6 Réseaux de neurones de type Radial Basis Function

On considère un réseau de neurones, paramétré par un ensemble de paramètres θ, comme une fonction $f_\theta(x)$. Pour une entrée $x \in \mathbb{R}^d$, il produit une prédiction de sortie $y = f_\theta(x)$.

Pour les réseaux de type Perceptron Multicouche (MLP) vus en cours, un neurone N_k de la première couche cachée reçoit une entrée x et a un vecteur de poids synaptiques $w^{(k)} \in \mathbb{R}^d$ et un biais $b^{(k)} \in \mathbb{R}$. Il calcule sa sortie h_k avec la formule $h_k = \text{sigmoid}(< w^{(k)}, x > + b^{(k)})$, où $< w^{(k)}, x >$ dénote le produit scalaire usuel.

On s'intéresse pour cette question à un type de réseaux de neurones différent, nommé RBF (Radial Basis Function). Ces réseaux à une couche cachée sont très similaires aux MLP. La différene est qu'un neurone RBF N_k de la couche cachée, ayant un vecteur de poids w_k calcule sa sortie h_k ainsi:

$$\begin{aligned} h_k &= \exp(-\beta \|x - w^{(k)}\|^2) \\ &= \exp\left(-\beta \sum_{j=1}^{d} (x_j - w_j^{(k)})^2\right) \end{aligned}$$

où exp désigne l'exponentielle et β est un hyper-paramètre (le même pour tous les neurones de la première couche cachée. Remarquez aussi qu'il n'y a **pas de biais**. Une unique couche cachée de m neurones RBF ayant des sorties $(h_1, ..., h_m) = h$ est typiquement suivie d'une couche de sortie linéaire avec des poids $(a_1, ..., a_m) = a$ pour donner une sortie $y = f_\theta(x) = < a, h > = \sum_{k=1}^{m} a_k h_k$.

6.1 Quel est l'ensemble θ des paramètres (excluant les hyper-paramètres) d'un tel réseau RBF?

$\theta = \{$ $\}$

A combien de nombre réels ajustables cela correspond-t-il?

6.2 Le coût pour un exemple x pour lequel le réseau prédit $f_\theta(x)$ alors que la vraie cible est t est donné par une fonction de coût différentiable $L(f_\theta(x), t)$. On cherche les valeurs des paramètres qui vont minimiser le coût empirique moyen sur un ensemble d'apprentissage $D_n = \{(x^{(1)}, t^{(1)}), ..., (x^{(n)}, t^{(n)})\}$. Exprimez ce problème de minimisation, puis nommez la technique qu'on va typiquement utiliser pour trouver une solution, et détaillez-là brièvement sous la forme d'un pseudo-code de haut niveau.

6.3 On s'intéresse au gradient, c.a.d la dérivée partielle du coût L par rapport aux paramètres, **on suppose qu'on a déjà calculé** $\frac{\partial L}{\partial y}$, et on va rétropropager le gradient. Exprimez et calculez (en fonction de a_k, h_k, et $\frac{\partial L}{\partial y}$):

$$\frac{\partial L}{\partial a_k} =$$

$$\frac{\partial L}{\partial h_k} =$$

puis, en fonction (entre autres) de $\frac{\partial L}{\partial h_k}$
Rappel de la formule pour dériver une exponentielle: $\exp(u)' = u' \exp(u)$, *ou encore:* $\frac{\partial \exp(u)}{\partial \theta} = \frac{\partial u}{\partial \theta} \exp(u)$

$$\frac{\partial L}{\partial w_j^{(k)}} =$$

7 Mélange de Gaussiennes

On considère, en dimension d, un mélange de k Gaussiennes avec des matrices de covariance **diagonales**.

7.1 A quoi sert un mélange de Gaussienne: pour quel type de problème d'apprentissage s'en sert-on? Dans quels cas un mélange de Gaussiennes est-il plus approprié qu'une unique Gaussienne?

7.2 Chacune des Gaussiennes **diagonales** de ce mélange a des paramètres: nommez-les et indiquez leur dimension. A combien de nombre réels ajustables cela correspond-t-il?

7.3 En plus des paramètres de chaque Gaussienne, quels autres paramètres y a-t-il dans un tel mélange? En tout combien y a-t-il de nombre réels ajustables dans les paramètres d'un tel mélange de Gaussiennes?

7.4 Écrivez la formule permettant de calculer la densité donnée par le mélange de Gaussiennes en un point $x \in \mathbb{R}^d$ (avec les notations que vous avez utilisé ci-haut pour représenter les paramètres).

7.5 On suppose qu'on dispose de deux procédures informatiques fournies par une librairie logicielle:

- une fonction `tirageGaussienne` reçoit en paramètre les paramètres d'une Gaussienne et retourne un point de $\mathbb{R}^d$ tiré aléatoirement selon cette distribution Gaussienne.
- une fonction `tirageDiscret` reçoit en paramètre un vecteur de probabilités sommant à 1, effectue un tirage selon ces probabilités discrètes et retourne l'indice entier correspondant. Ainsi par ex., si on lui passe en paramètre le vecteur $(0.30, 0.50, 0.20)$, la procédure retournera la valeur 1 dans 30% des appels, la valeur 2 dans 50% des appels et la valeur 3 dans 20% des appels.

Écrivez, sous forme de pseudo-code, la procédure permettant de générer un points $x \in \mathbb{R}^d$ selon la distribution de mélange de Gaussiennes paramétrée telle que vous l'avez définie plus haut.