Fondements de l'Apprentissage Machine (IFT 3395/6390)

Examen Intra

Automne 2013

Professeur : Pascal Vincent

Jeudi 31 octobre 2013
Durée : 2h30

- Seule documentation permise : 2 feuilles recto/verso (format letter 8" 1/2 x 11") pour votre résumé de cours.
- L'utilisation d'appareils électroniques n'est pas autorisée durant l'examen (à l'exception d'une montre pour connaître l'heure).
- Le total de l'examen est sur 100pts (dont 4 pts pour votre nom et prénom...). Veuillez répondre aux questions directement dans les zones de blanc laissées à cet effet. Répondez de manière concise, mais précise. **Bon examen !**

(4 pts) :
Prénom :
Nom :
Code permanent :
IFT3395 ou IFT6390 :

Programme d'études :
Laboratoire d'attache (s'il y a lieu) :

Notation

Pour toutes les questions, on suppose qu'on travaille avec un ensemble de données de départ comportant n exemples noté $D_n = \{z^{(1)}, \ldots, z^{(n)}\}$ avec, dans les cas supervisés, $z^{(k)} = (x^{(k)}, t^{(k)})$ où $x^{(k)} \in \mathbb{R}^d$ est l'entrée et $t^{(k)}$ est la cible correspondante. On vous demande de respecter scrupuleusement les notations de cet énoncé (c.a.d. ne vous contentez-pas de retranscrire des formules telles quelles mais adaptez les aux notations de l'énoncé si besoin).

1 Concepts graphiques (20 pts)

Soient les notions suivantes :
- point de l'espace d'entrée : x
- ensemble d'entraînement D_{train}
- ensemble de validation D_{valid}
- prédiction : C (numéro de classe prédite)
- paramètres : θ
- hyper-paramètres : λ
- paramètres "optimaux" appris (optimisés) par l'algorithme d'apprentissage sur l'ensemble d'entraînement : θ^*
- un risque empirique (ex : taux d'erreur de classification) calcué sur un ensemble D avec une fonction dont les paramètres valent θ : $\hat{R}(D, \theta)$

Les concepts graphiques suivants correspondent à des représentations graphiques de certaines fonctions. Indiquez à droite de chacun à "une fonction de quoi vers" quoi ils correspondent (c.a.d. à quoi correpsondent les axes du graphique).
Par ex. si un graphique représentait la valeur apprise ("optimale") des paramètres (θ^*) en fonction des hyper-paramètres (λ), il faudrait écrire : $\lambda \rightarrow \theta^*$.

1. Régions de décision :

2. Paysage de coût ou d'erreur :

3. Courbes d'apprentissage (courbe d'entraînement, et de validation) :

- Sur lequel de ces graphiques va-t-on souvent recourir à une descente de gradent ?

- À quoi sert la descente de gradient dans ce contexte ?

- Complétez la phrase : le gradient est un ..... qui a les mêmes dimensions que ..... et qui pointe dans la direction de .....

- Écrivez l'équation (de mise à jour) qu'on itère jusqu'à convergence lors d'une descente de gradient (veillez à bien utiliser les notations définies au début de la question).

- On se sert-on des courbes d'apprentissage pour décider de quoi ? Expliquez comment on procède précisément ?

2 Régions et frontières de décision

2.1 Algorithme de dessin de régions de décision (8 pts)

On suppose qu'on a un problème de classification 2D à m classes, dont l'ensemble d'emtraînement est $D_n = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(n)}, t^{(n)})\}$ avec $x^{(i)} \in \mathbb{R}^2$ et $t^{(i)} \in \{1, \ldots, m\}$.
On suppose également qu'on l'a utilisé comme ensemble d'entraînement pour un certain algorithme d'apprentissage A. Cet algorithme a appris une fonction $g : \mathbb{R}^2 \to \mathbb{R}^m$ qui associe à un point x un vecteur de m scores correspondant aux scores pour chacune des m classes.

1. Donnez l'expression de la fonction de décision qui à un point x associerait le numéro de classe prédit :

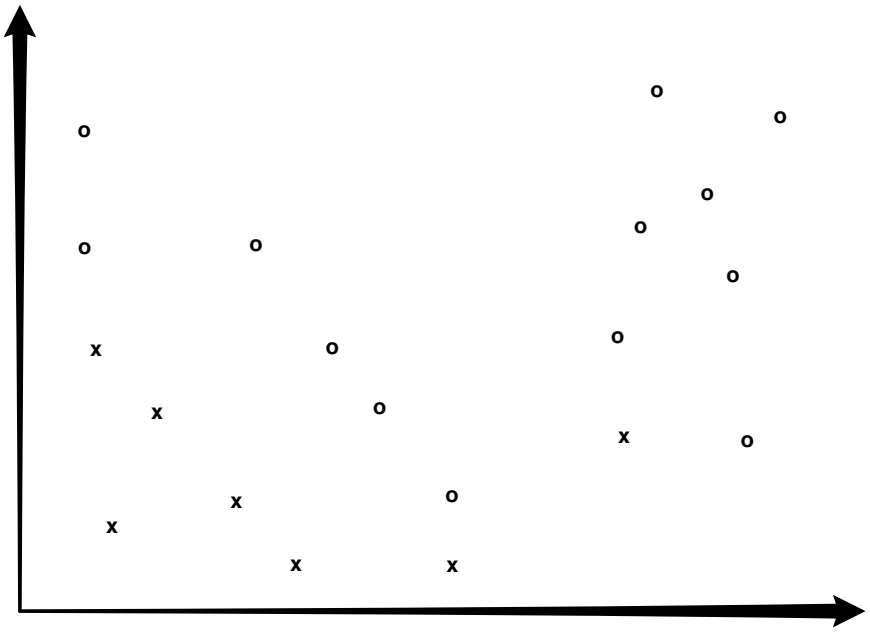
2. Écrivez le pseudo-code pour permettre d'afficher les régions de décision correspondantes (dans une partie de l'espace délimitée entre x_{1min} et x_{1max} pour la première coordonnée, et x_{2min} et x_{2max} pour la seconde coordonnée). La région de décision d'une classe devrait apparaître avec une couleur correpsondant à cette classes, telles que prédéfinies dans le tableau `couleurs_de_classe = [ ''bleu'', ''rouge'', ''vert'', ...]`. On suppose que vous disposez d'une librairie d'affichage graphique scientifique qui gère l'affichage (axes de coordonnées etc...). Pour afficher un point d'une certaine couleur aux coordonnées (x_1, x_2) il vous suffit d'appeler la fonction `point(`x_1`,`x_2`,`*`couleur`*`)`. Pour dessiner les régions de décision il suffit de dessiner les points de la bonne couleur sur une grille suffisamment fine (disons 200×200) qui couvre la région désirée ($[x_{1min}, x_{1max}] \times [x_{2min}, x_{2max}]$). Écrivez ci-dessous le pseudo-code :

   `def dessine_regions_de_decision_2D(` g`,` x_{1min}`,` x_{1max}`,` x_{2min}`,` x_{1max}`,` `couleurs_de_classe):`

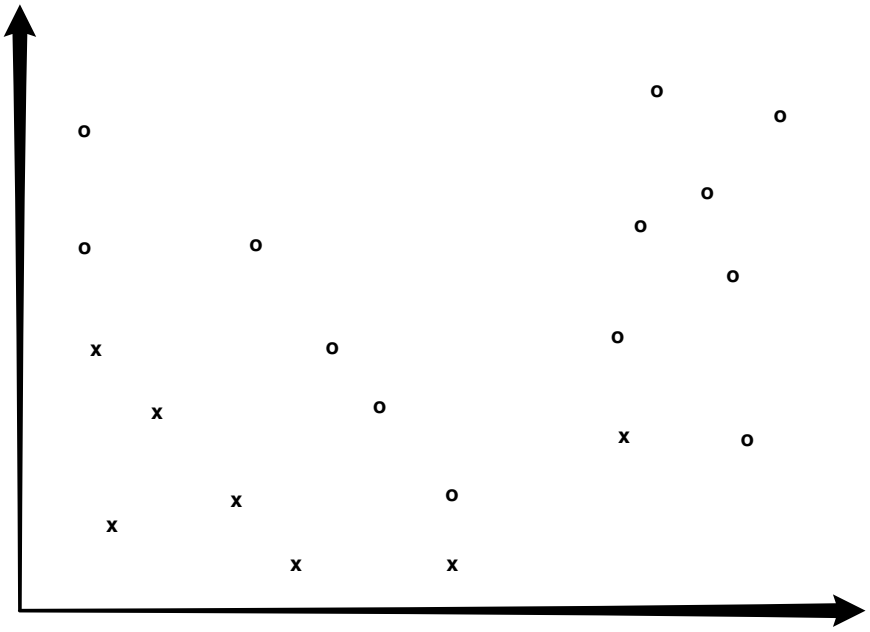
2.2 Région et frontière de décision pour différents algorithmes (15 pts)

On a reproduit ci-dessous plusieurs dessins représentant le même ensemble de données 2D de classification binaire. On va considérer plusieurs algorithmes d'apprentissage auxquels on aurait fournit cet ensemble comme ensemble d'entraînement. Dans chaque cas **hachurez ou coloriez** légèrement la **région de décision** de la **classe des ronds**, et tracez la *frontière de décision* en trait plein. (Si il y a une région où le classifieur ne peut pas se prononcer sur la classe, indiquez-là par un " ?").

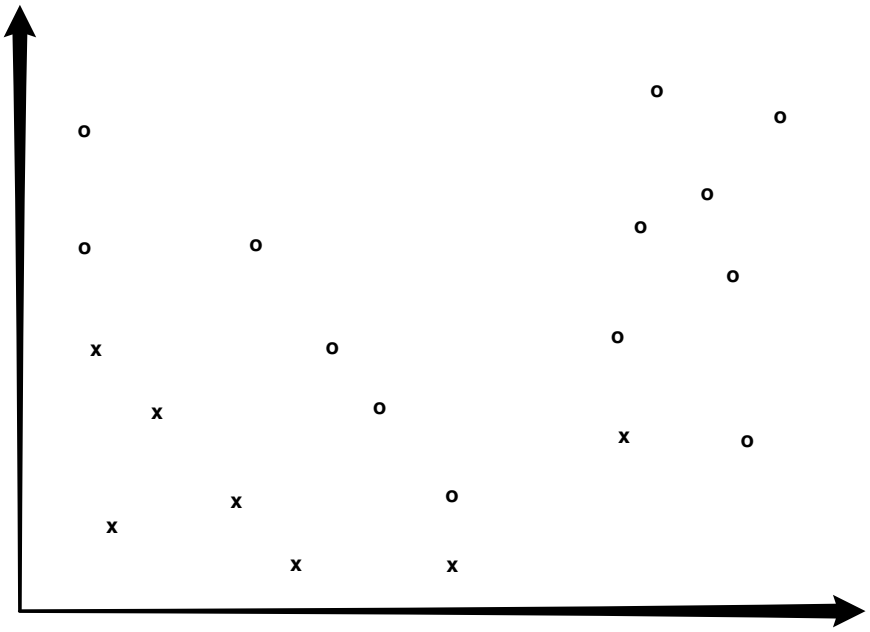
k-plus proche voisin (k-nearest neighbor) avec k=1



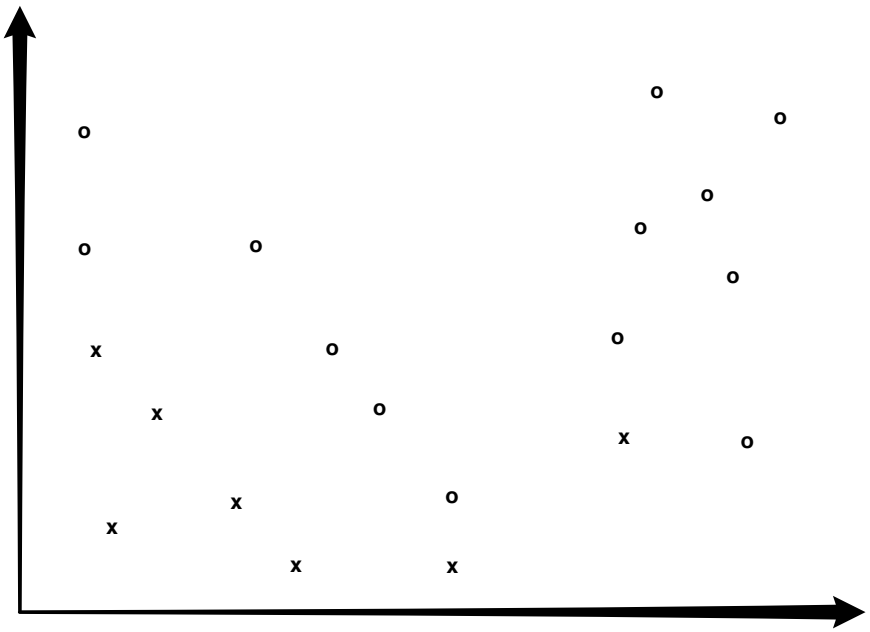
Classifieur à base d'histogramme, avec chaque dimension subdivisée en 3 parties égales



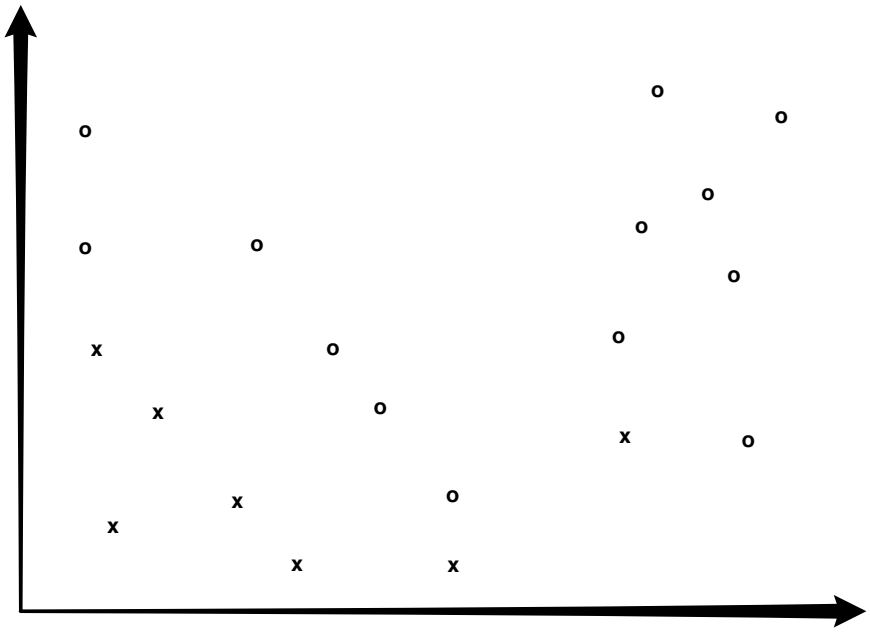
k-plus proche voisin (k-nearest neighbor) avec k=20



classifieur de plus proche moyenne (centroide) (dessin approximatif OK)



Machine à vecteur de support (SVM) linéaire à marge souple tolérant une erreur (faites comme si le point d'erreur n'était pas là).



3 Nombres de paramètres (13 pts)

Les algorithmes d'apprentissage, apprennent (calculent, trouvent, optimisent ou mémorisent) durant une phase d'*entraînement*, un certain nombre de *paramètres*, qui seront par la suite utilisés pour faire une prédiction sur de nouveaux points de test. Ces paramètres sont constitués d'un certain nombre de *scalaires* (nombres réels). Ex : si les paramètres sont une matrice $m \times n$ et un vecteur de taille m, alors le nombre total de paramètres scalaires appris est $mn + m$. Pour chacun des cas et algorithmes suivants écrivez le **nombre total de paramètres scalaires** appris ou mémorisés (on exclut de ce compte les hyper-paramètres) :

1. Problème de régression avec $x \in \mathbb{R}^d$; algorithme de régression linéaire (affine) :
2. Problème de classification binaire (2 classes) avec $x \in \mathbb{R}^d$; algorithme du perceptron :
3. Problème de classification binaire (2 classes) avec $x \in \mathbb{R}^d$; algorithme de régression logistique :
4. Problème de régression avec $x \in \mathbb{R}$ (1 dimension) ; algorithme de régression polynomiale de degré 2 :
5. Problème de régression avec $x \in \mathbb{R}^2$ (2 dimension) ; algorithme de régression polynomiale de degré 2 :
6. Problème de régression avec $x \in \mathbb{R}^d$. Histogramme avec chaque dimension subdivisée en k intervalles :
7. Problème de classification à m classes ($m \geq 3$) avec $x \in \mathbb{R}^d$. Histogramme avec chaque dimension subdivisée en k intervalles :
8. Problème d'estimation de densité avec $x \in \mathbb{R}^d$. Une *Gaussienne isotropique* :
9. Problème d'estimation de densité avec $x \in \mathbb{R}^d$. Une *Gaussienne* avec covariance *diagonale* :

10. Problème d'estimation de densité avec $x \in \mathbb{R}^d$. Une *Gaussienne* avec covariance pleine :
11. Problème de classification à m classes ($m \geq 3$) avec $x \in \mathbb{R}^d$; classifieur de Bayes utilisant des *Gaussiennes diagonales* (sans autre contrainte ou partage de paramètres. Attention, n'oubliez rien !) :
12. Problème d'estimation de densité avec $x \in \mathbb{R}^d$; estimateur de Parzen avec noyau Gaussien isotropique de variance fixée à l'avance, entraîné sur un ensemble comportant n exemples (pensez à ce qu'il est nécessaire de conserver en mémoire afin d'effectuer une prédiction sur un point de test) :
13. Problème de classification à m classes avec $x \in \mathbb{R}^d$; classifieur de fenêtres de Parzen avec noyau Gaussien isotropique de variance fixée à l'avance, entraîné sur un ensemble comportant n exemples (pensez à ce qu'il est nécessaire de conserver en mémoire afin d'effectuer une prédiction sur un point de test) :

4 Régression avec prédicteur constant (15 pts)

On suppose qu'on a affaire à un problème de régression avec des cibles scalaires $t \in \mathbb{R}$. On va "apprendre" le prédicteur le pus simple possible : un prédicteur "constant" qui prédit toujours la même valeur c, quel que soit x :

$$f(x) = c$$

Son seul paramètre est donc $\theta = c$.

1. Quelle est la fonction de perte (coût) le plus souvent utilisée pour la régression ?

2. En utilisant le principe de minimisation du risque empirique, avec cette fonciton de perte, exprimez (avec une équation détaillée) le problème d'optimisation qui permettra de trouver la valeur optimale c^* du paramètre :

   $\theta^* = c^* =$

3. Résolvez ce problème d'optimisation (écrivez toutes les étapes)

5 Classifieurs linéaires

5.1 Fonction discriminante linéaire (10 pts)

On considère un problème de classification à 2 classes en dimension d où D_n sert d'ensemble d'entraînement. $D_n = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(n)}, t^{(n)})\}$ avec $x^{(i)} \in \mathbb{R}^d$ et $t^{(i)} \in \{-1, 1\}$.

1. Donnez l'équation d'une fonction discriminante linéaire :

2. Quels sont ses paramètres (précisez aussi leurs dimensions) : $\theta = \{$

3. Donnez l'équation de la **fonction de décision** (en faisant appel à la fonction discriminante linéaire)

4. Exprimez le **taux d'erreur de classification** qu'obtient une telle fonction sur un ensemble d'entraînement D_n.

5. Quelle est l'équation précise de la frontière de décision (exprimée en fonction des paramètres)

6. Comment nomme-t-on cette forme géométrique (la frontière de décision que vous avez définie à la question précédente)

7. Nommez tous les algorithmes d'apprentissage que vous connaissez, qui permettent d'apprendre un classifieur linéaire :

5.2 Séparabilité linéaire (10 pts)

a) À quoi s'applique la notion de séparabilité linéaire : de quoi dit-on qu'il est ou non linéairement séparable ?

b) Parmi les définitions suivantes, lesquelles permettent de définir la notion "linéairement séparable" (entourez la ou les bonnes réponses) :

1. L'algorithme de classification linéaire considéré est capable d'atteindre 0 erreurs sur tout ensemble de donnée de classification binaire.
2. L'algorithme du perceptron stochastique (on-line) s'arrête au bout d'un nombre fini d'itérations.
3. On peut positionner un hyper-plan dans l'espace tel que tous les points d'une classe (parmi l'ensemble de donnée) sont d'un bord de l'hyper-plan et tous les points de l'autre classe sont de l'autre bord.
4. Tout classifieur linéaire fera 0 erreurs sur l'ensemble de donnée
5. Il existe une fonction discriminante linéaire dont le signe indique la classe de tout point généré par la distribution inconnue ayant généré les données
6. Il existe une fonction discriminante linéaire dont le signe indique la classe de tout point de l'ensemble de donnée
7. Soit $D_n = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(n)}, t^{(n)})\}$ avec $x^{(i)} \in \mathbb{R}^d$ et $t^{(i)} \in \{-1, 1\}$
   $\exists w \in \mathbb{R}^d, b \in \mathbb{R}$ tels que $\forall (x, t) \in D_n$, $\text{sign}(w^T x + b) = t$
8. Soit $D_n = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(n)}, t^{(n)})\}$ avec $x^{(i)} \in \mathbb{R}^d$ et $t^{(i)} \in \{-1, 1\}$
   $\exists w \in \mathbb{R}^d, b \in \mathbb{R}$ tels que $\forall (x, t) \in D_n$, $t(w^T x + b) < 0$
9. Soit $D_n = \{(x^{(1)}, t^{(1)}), \ldots, (x^{(n)}, t^{(n)})\}$ avec $x^{(i)} \in \mathbb{R}^d$ et $t^{(i)} \in \{-1, 1\}$
   $\exists w \in \mathbb{R}^d, b \in \mathbb{R}$ tels que $\left(\sum_{i=1}^{n} I_{\{(w^T x^{(i)} + b) t^{(i)} > 0\}}\right) = n$

c) Dessinez ci-dessous des exemples comportant une dizaine de points. Dans chaque cas, dessinez clairement les axes. Pour les cas linéairement séparables, dessinez également en plein la frontière de décision.

Dessinez ci-dessous un exemple 2D linéairement séparable	Dessinez ci-dessous un exemple 2D **non** linéairement séparable
Dessinez ci-dessous un exemple 1D linéairement séparable	Dessinez ci-dessous un exemple 1D **non** linéairement séparable (avec frontière de décision)

5.3 Séparation non-linéaire à l'aide d'un classifieur linéaire (5 pts)

On vous demande de penser à un exemple 1D ($x \in \mathbb{R}$) non-linéairement séparable (avec une dizaine de points), et à une transformation (non-linéaire) $\phi : \mathbb{R} \to \mathbb{R}^2$, **dont vous donnerez explicitement la formule**, qui rendrait votre exemple linéairement séparable en 2D.

Dessinez ci-dessous votre exemple 1D et sa transformation 2D.

Votre exemple 1D non linéairement séparable	Votre transformation	L'exemple transformé en 2D (à présent linéairement séparable)
	$\phi(x) = \begin{pmatrix} \cdots \\ \cdots \end{pmatrix}$	

- Tracez la frontière de décision obtenue en 2D
- Hachurez ou coloriez légèrement une des régions de décision dans le graphique 2D.
- Hachurez ou coloriez légèrement la région de décision correspondante sur la vue 1D.