Examen Intra IFT3390/6390

Mardi 20 février 2007

Durée: 1h50

Professeur: Pascal Vincent

Prénom:

Nom:

Code permanent:

IFT 3390 ou 6390?

Notations

Les notations suivantes sont définies pour tout l'examen:
On suppose qu'on dispose d'un ensemble de données de n exemples: $D_n = \{z_1, ..., z_n\}$. Dans le cas supervisé (classification, régression), chaque exemple z_i est constitué d'une paire *observation, cible:* $z_i = (x_i, y_i)$, alors que dans le cas non-supervisé (estimation de densité), 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: $x_i \in \mathbb{R}^d$.

1 Estimation de densité de probabilité paramétrique Gaussienne (25 pts)

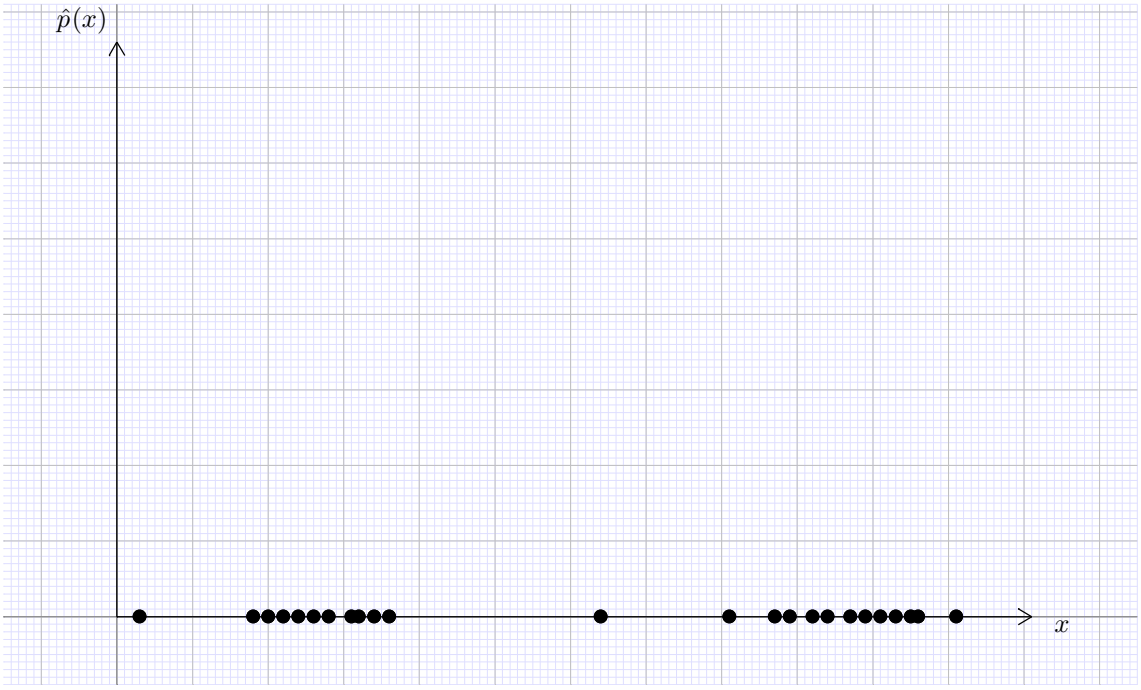
On rappelle la formule de densité de probabilité pour une Gaussienne multivariée de moyenne μ et de matrice de covariance C:

$$\mathcal{N}_{\mu,C}(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\det C}} e^{-\frac{1}{2}(x-\mu)^T C^{-1}(x-\mu)}$$

On suppose pour cette question que les données x_i (en dimension d) sont distribuées selon une loi Gaussienne dont on ne connaît aucun des paramètres, et qu'on veut les déterminer.

1. Combien de paramètres scalaires (nombres réels) y a-t-il à déterminer pour ce modèle?
2. Quel principe général peut-on utiliser pour déterminer ces paramètres à partir de l'ensemble de données d'exemples D_n?
3. Exprimez, selon ce principe, l'équation à optimiser (précisez si on veut minimiser ou maximiser) qui permettrait de déterminer la valeur des paramètres recherchés.
4. Quel μ^* est solution de cette optimisation (donnez directement l'expression permettant son calcul. On ne demande pas de la redériver.).
5. Quel C^* est solution de cette optimisation (donnez directement l'expression permettant son calcul. On ne demande pas de la redériver.)
6. Quel est, selon ce modèle, la densité de probabilité pour un nouveau point de test $x \notin D_n$: $\hat{p}(x) = ...$?

2 Fenêtres de Parzen pour l'estimation de densité (25 pts)



On a représenté sur le graphique ci-dessus un petit ensemble de données pour un problème d'estimation de densité en 1 dimension.

1. Tracez, sur le graphique, de façon approximative, la densité estimée par un estimateur de densité de Parzen avec un noyau Gaussien d'écart type σ:

   a) En trait fin, pour un σ très (trop) petit.

   b) En trait pointillé pour un σ très (trop) grand.

   c) En trait épais pour un σ intermédiaire approprié.

2. Dans quel sens un σ peut-il être plus "approprié" qu'un autre? Quel type d'erreur espère-t-on minimiser avec un σ "approprié"?

3. Pouvez-vous penser à une technique pour trouver un σ approprié (pour Parzen), si on disposait d'un grand nombre d'exemples d'apprentissage? Sur quel critère pourrait-on se baser? Donnez une formule pour le calcul de votre critère, et expliquez brièvement la technique employée.

4. Tracez approximativement la courbe de densité qu'on obtiendrait avec une estimation de densité *paramétrique* Gaussienne (question 1 de l'examen). Utilisez au besoin un style de trait différent des précédents, et ajoutez une légende au graphique! L'estimation de densité obtenue avec cette méthode vous semble-t-elle à première vue appropriée pour l'ensemble de données illustré sur le graphique? Pourquoi?

5. **BONUS:** Pouvez-vous penser à une façon de décider, pour un ensemble d'apprentissage donné, si il vaudrait mieux utiliser un estimateur de Parzen ou un une estimation de densité paramétrique Gaussienne?

3 Classifieur de plus proche moyenne (25 pts)

On suppose qu'on a affaire à un problème de clasification avec m classes: $y_{i \in \{1,\ldots,m\}}$

Dans cette question, on s'intéresse à l'algorithme de classification de plus proche moyenne (aussi appelé classifieur à centroide) qui apprend pour chaque classe un unique point prototype: le *centroide* des points de la classe, c.a.d. le point "moyenne" des points de la classe dans l'ensemble d'apprentissage. En gros, l'algorithme "résume" chaque classe par son centroide calculé sur l'ensemble d'apprentissage. La décision pour un point de test consiste alors simplement à répondre la classe dont le centroide est le plus proche en distance Euclidienne.

Note: on ne demande pas d'écrire le code pour le calcul de la distance Euclidienne.

1. Définissez des variables correspondant aux *paramètres à apprendre* du classifieur. Dimensionnez-les en fonction des dimensions du problème (n, d, m). Précisez leur nom, leur type, etc... Vous pouvez utiliser des scalaires, vecteurs, matrices, ou tableaux comme bon vous semble, du moment que votre définition est claire et explicite.

2. A combien de paramètres scalaires (nombres réels) cela correspond-t-il?

3. Écrivez, sous forme de pseudo-code la méthode *train* de ce classifieur, qui apprend les *paramètres* à partir d'un ensemble d'apprentissage D_n. N'hésitez pas à utiliser des expressions de calcul vectoriel (additions de vecteurs, multiplication d'un vecteur par un scalaire, ...)

4. Exprimez la complexité algorithmique pour la méthode *train* (notation $O(...)$) en fonction des dimensions du problème (n, d, m, ...).

5. Écrivez, sous forme de pseudo-code la méthode *computeOutput* qui produit le vecteur des scores de chaque classe pour un point de test x. Et la fonction *decideClass* qui retourne la décision de classe (un entier) pour un point x.

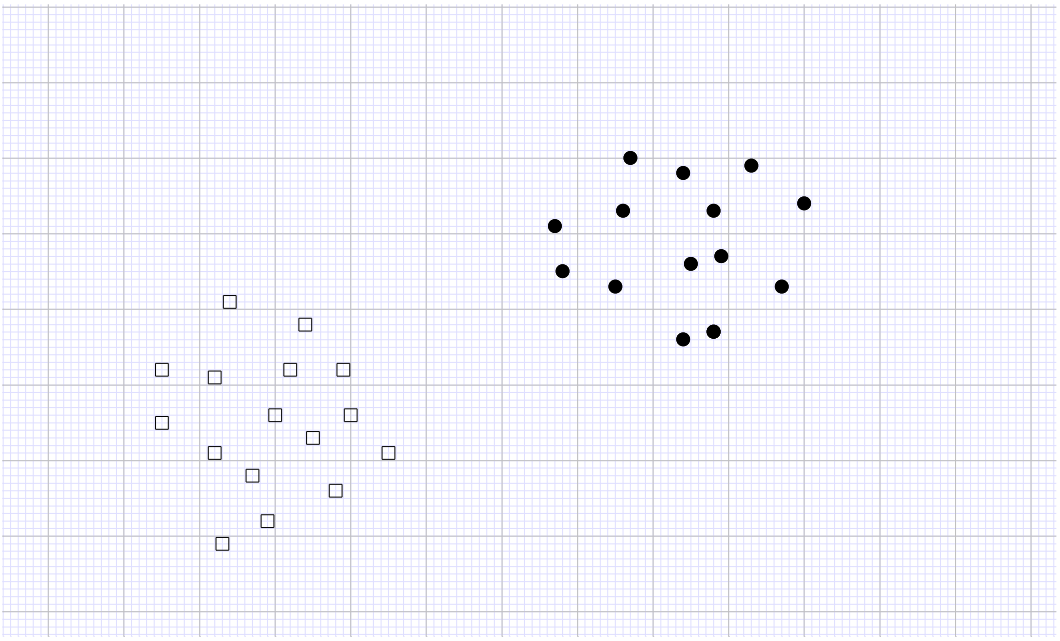
6. Exprimez la complexité algorithmique pour la méthode *computeOutput* (notation $O(...)$) en fonction des dimensions du problème.

7. Quelle serait en comparaison, la complexité algorithmique de la méthode *computOutput* d'un classifieur de k Plus Proches Voisins?

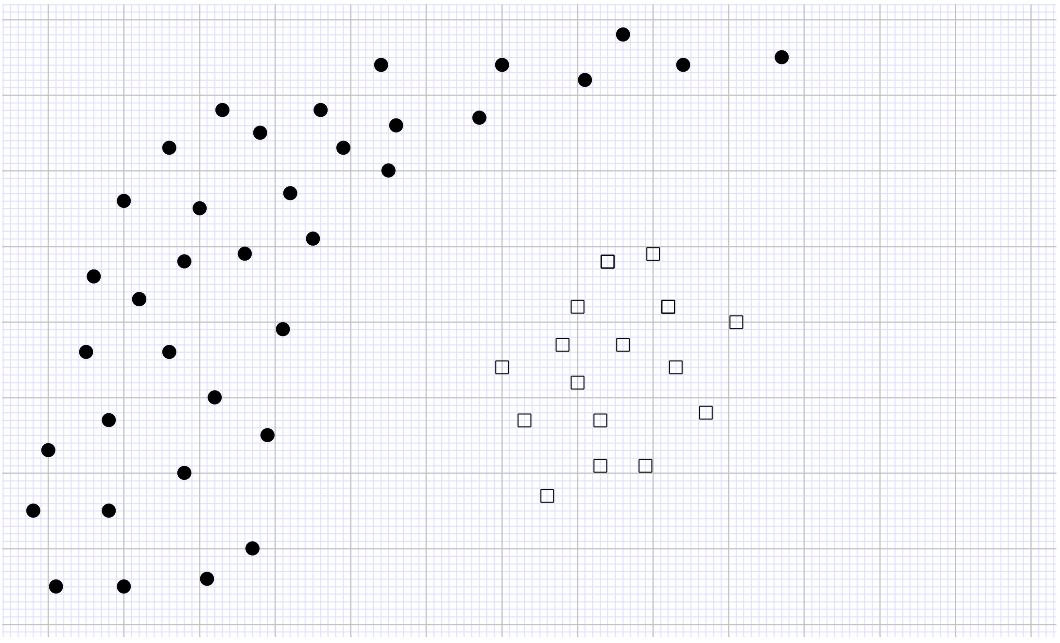
8. Dans le cas de deux classes, écrivez et simplifiez la formule de la frontière de décision obtenue avec un classifieur de plus proche moyenne. Quelle est la *forme* de la frontière de décision?

9. Placez les centroides, et tracez (approximativement) en **trait pointillé** la frontière de décision que produirait l'algorithme de plus proche moyenne sur les problèmes de classification 2D suivants:

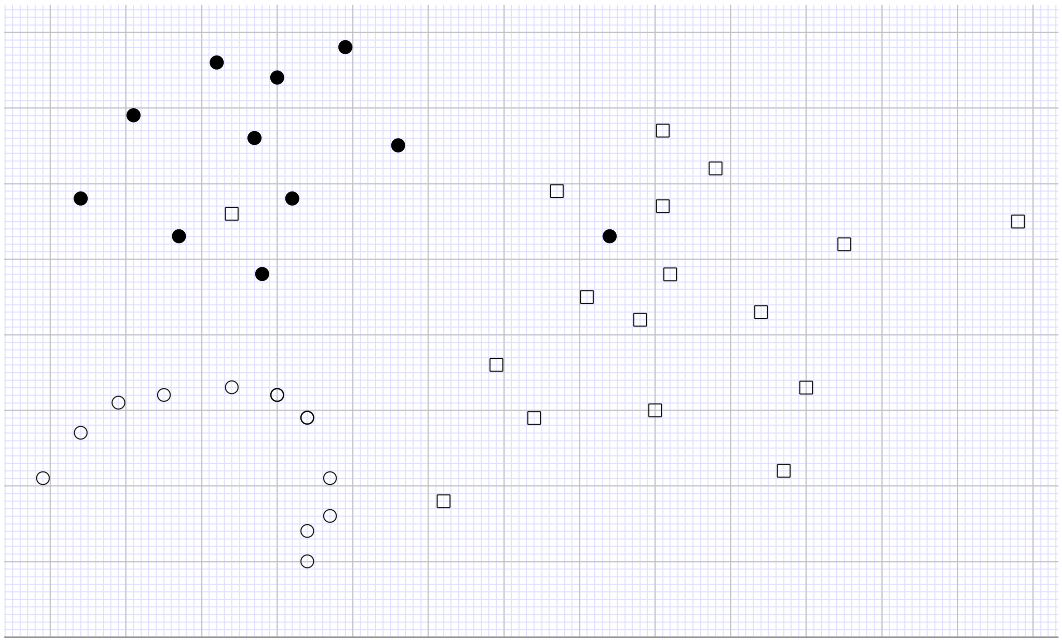
a) un problème à 2 classes:



b) un autre problèmes à 2 classes:



c) un problème à 3 classes:



10. Tracez (approximativement) en **trait plein** la frontière de décision que produirait l'algorithme du Plus Proche Voisin (1-PPV) pour ces mêmes problèmes 2D.

11. En vous basant sur *les* questions précédentes, nommez 1 avantage de l'algorithme 1-PPV par rapport à l'algorithme de plus proche moyenne, et 1 avantage de l'algorihme de plus proche moyenne par rapport à 1-PPV.

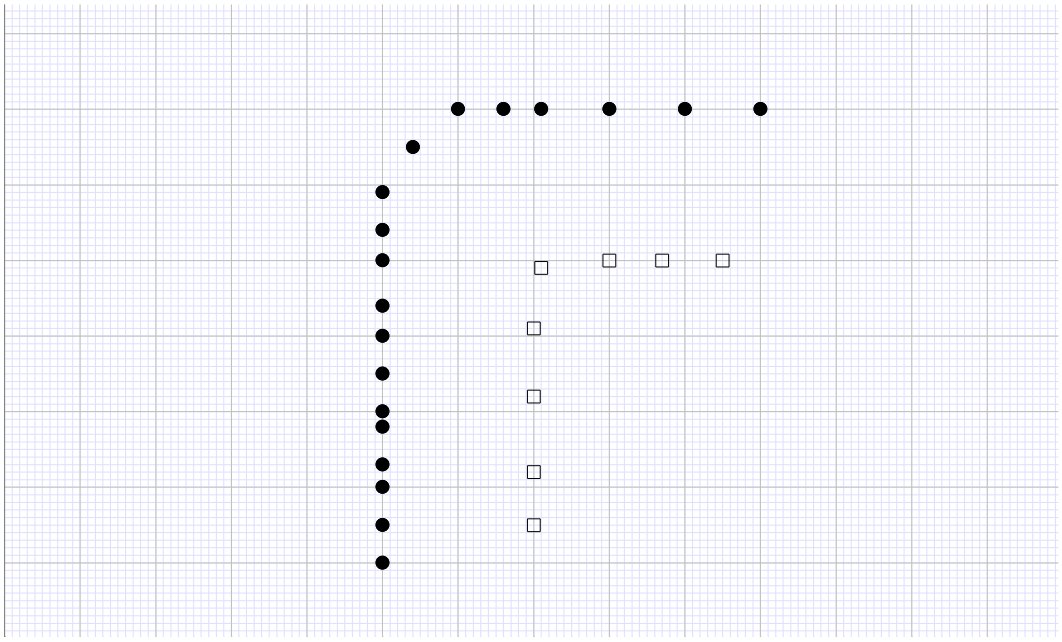
12. **BONUS:** Mis à part les algoritmes à base de voisinage (comme k-PPV ou Parzen), de quel autre algorithme de classification le classifieur de plus proche moyenne se rapproche-t-il? Qu'est-ce que cet autre type de classifieur a de différent (de plus)?

4 Classifieur linéaire (25 pts)

On considère pour cette question un problème de classification binaire (2 classes) dont les classes sont représentées par un label $y \in \{-1, +1\}$.

On veut entraîner un classifieur linéaire dont les paramètres sont un vecteur de poids $w = (w_{[1]}, ..., w_{[d]})^T \in \mathbb{R}^d$ et un biais $b \in \mathbb{R}$.

1. Écrivez la fonction discriminante $g(x)$ et la fonction de décision $f(x)$ pour un tel classifieur linéaire.
2. Exprimez la fonction de perte qui comptabiliserait les erreurs de classification, et formulez le calcul du taux d'erreur de classification sur un ensemble D'.
3. Expliquez, dans vos propres mots, le principe de minimisation du risque empirique.
4. Exprimez l'équation de la frontière de décision. Tracez (approximativement) pour l'ensemble de données 2D ci-dessous, une frontière de décision, réalisable par un classifieur linéaire, et qui ferait le moins possible d'erreurs de classification sur cet ensemble d'apprentissage. Ces données sont-elles linéairement séparables?



5. Plutôt que l'erreur de classification, on va plutôt utiliser la fonction de perte $L(g(x), y) = e^{-yg(x)}$. Exprimez le risque empirique $\hat{R}(D_n)$ correspondant à cette fonction de perte, et formulez le problème d'optimisation qui nous permettrait de trouver la valeur des paramètres w et b du modèle minimisant ce risque empirique.
6. **BONUS:** Pour résoudre ce problème d'optimisation, on va utiliser la technique de descente de gradient simple (dans sa forme "batch"). Calculez le gradient pour chacun des paramètres $w_{[k]}$ et b et indiquez la formule de mise-à-jour des paramètres à chaque pas de gradient.
7. On voudrait estimer la performance de généralisation (en terme de *taux d'erreur de classification*) de ce classifieur sur le problème ayant donné lieu à l'ensemble de données D_n. Comment peut-on procéder si n est suffisamment grand? Précisez la technique sous forme d'un bref pseudo-code.
8. Si on dispose de peu d'exemples (n petit), quelle autre technique peut-on utiliser pour estimer la performance de généralisation? Précisez la technique sous forme de pseudo-code.
9. **BONUS:** Quel prétraitement pourrait-on appliquer aux données pour obtenir, avec ce même algorithme, une frontière de décision non-linéaire, possiblement capable de séparer les points de deux classes dans l'exemple 2D ci-dessus?