IFT3395/6390 Fondements de l'apprentissage machine

Professeur : Pascal Vincent

Devoir 1

— Ce devoir est à faire en équipe de 2 personnes. Assurez-vous d'avoir noté le nom de tous les coéquipiers en tête du rapport et en tête en commentaires de chacun des fichiers que vous remettrez.
— On demande, la remise d'un rapport en format électronique (.pdf). Tous les fichiers de code source que vous aurez créé ou adapté devront également être remis. La partie pratique est à faire en python (en utilisant les librairies numpy et matplotlib), et vous pouvez bien entendu fortement vous inspirer de ce qui a été fait pendant les labos.
— Vous pouvez remettre votre code python sous la forme d'un notebook ipython .ipynb. Pour produire un rapport avec des formules mathématiques vous pouvez utiliser le logiciel de votre choix : LaTeX ; LyX ; Word ; voire même écrire directement les parties théoriques dans le notebook (en entrant les équations en format MathJaX pour qu'elles s'affichent). Dans tous les cas on vous demande d'exporter votre rapport en .pdf que vous remettrez.
— La remise doit se fait via StudiUM. Une seule remise par équipe (un seul des coéquipiers effectue la remise). Assurez-vous d'avoir noté le nom de tous les coéquipiers en tête du rapport. Si vous avez beaucoup de fichiers à remettre vous pouvez aussi (c'est peut-être plus pratique) en faire une archive (.zip ou .tar.gz) et téléverser le fichier d'archive.

1 Petit exercice de probabilités

Une étude réalisée aux États-Unis il y a quelques années auprès de médecins, pour mesurer leur "intuition probabiliste" comportait la question suivante :

La probabilité d'un cancer du sein chez les femmes dans la quarantaine participant à un test de routine (mammographie) est de 1%. Si une femme est atteinte d'un cancer du sein, il y a 80% de chances que le test soit positif. Chez une femme qui n'est pas atteinte de cancer, il y a une probabilité de 9,6% que le test soit positif.

Une femme dans la quarantaine ayant passé ce test de routine reçoit un résultat de test positif. Quelle est la probabilité qu'elle soit effectivement atteinte d'un cancer du sein ?

A) plus de 90%
B) entre 70% et 90%
C) entre 50% et 70%
D) entre 30% et 50%
E) entre 10% et 30%
F) moins de 10%

95% des médecins interrogés ont répondu B). Qu'en pensez-vous ?

Formalisez la question et calculez la probabilité exacte.

Indication : utilisez la règle de Bayes...

2 Estimation de densité paramétrique Gaussienne, v.s. estimation de densité par fenêtres de Parzen

Dans cette question on considère un ensemble de données $D = \{x^{(1)}, \ldots, x^{(n)}\}$ avec $x \in \mathbb{R}^d$.

1. Supposons que l'on ait entraîné les paramètres d'une densité Gaussienne **isotropique** sur D (pour maximiser la vraisemblance) pour en estimer la densité de probabilité.
   (a) Nommez ces paramètres et indiquez-en les dimensions.
   (b) Si on apprend les paramètres en utilisant le principe de maximum de vraisemblance, exprimez en fonction des points de D la formule qui nous donnera la valeur des paramètres optimaux (indiquez seulement la formule qui calcule le résultat, on ne vous demande pas de la redériver)
   (c) Quelle est la complexité algorithmique de cet apprentissage (entrainement) c.a.d. du calcul de ces paramètres ?
   (d) Pour un point de test x, écrivez la fonction qui donnera la densité de probabilité prédite au point x :
   $\hat{p}_{gauss-isotrop}(x) = ?$

(e) Quelle est la complexité algorithmique pour le calcul de cette prédiction à chaque nouveau point x ?

2. Considérons à présent qu'on utilise plutôt pour estimer la densité de probabilité des fenêtres de Parzen avec un noyau Gaussien isotropique de largeur (écart-type) σ, et qu'on ait entraîné ces fenêtres de Parzen sur D.

   (a) Supposons que l'utilisateur ait fixé σ. En quoi consiste la phase « entrainement/apprentissage » pour ces fenêtres de Parzen ?

   (b) Pour un point de test x, écrivez en *une seule formule détaillée* (c.a.d. avec des exponentielles), la fonction qui donnera la densité de probabilité prédite au point x :
   $\hat{p}_{Parzen}(x) =?$

   (c) Quelle est la complexité algorithmique pour le calcul de cette prédiction à chaque nouveau point x ?

3. Capacité

   (a) Laquelle de ces 2 approches (paramétrique Gaussienne v.s. Parzen à noyau Gaussien) a la plus forte *capacité* ? Expliquez.

   (b) Avec laquelle de ces approches, et dans quel *cas précis*, a-t-on toutes les chances d'être en sur-apprentissage ?

   (c) Le σ dans les fenêtres de Parzen est généralement traité comme un hyper-paramètre, alors que pour une densité paramétrique Gaussienne il est généralement traité comme un paramètre. Pourquoi ?

4. Considérons maintenant une estimation de densité paramétrique avec une densité Gaussienne **diagonale**.

   (a) Exprimez l'équation d'une densité Gaussienne diagonale dans $\mathbb{R}^d$. Et précisez ce que sont ses paramètres et leurs dimension.

   (b) Démontrez que les *composantes* d'un vecteur aléatoire qui suit une distribution Gaussienne diagonale sont des variables aléatoires **indépendantes**.

   (c) En utilisant comme coût $-\log p(x)$ écrivez l'équation qui correspondrait à la minimisation du risque empirique sur l'ensemble d'entraînement D (pour apprendre les paramètres)

   (d) Résolvez cette équation de manière analytique pour obtenir les paramètres optimaux.

5. Face à un problème de classification avec $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, on peut construire un classifieur de Bayes en se basant sur l'une ou l'autre de ces 3 estimations de densité.

   (a) Écrivez dans vos propres mots comment vous apprendriez un classifieur de Bayes (phase d'entraienement)

   (b) Pour un point de test x, écrivez, la fonction qui donnera le vecteur de probabilité prédite pour chaque classe au point x :
   $g(x) = (\ldots, \ldots, \ldots, \ldots)$

3 Partie pratique : estimation de densité

1. Implémentez un estimateur de densité paramétrique Gaussien diagonal. Il devra pouvoir fonctionner pour des données de dimension d arbitraire. Comme vu dans les labos, il devrait avoir une méthode **train** pour apprendre les paramètres et une méthode **compute_predictions** qui calcule les *log* de densité.
2. Implémentez un estimateur de densité de Parzen à noyau Gaussien isotropique. Il devra pouvoir fonctionner pour des données de dimension d arbitraire. De même il devrait avoir une méthode **train** pour apprendre les paramètres et une méthode **compute_predictions** qui calcule les *log* de densité.
3. Densités 1D : Parmi l'ensemble de données Iris, choisissez le sous-ensemble correspondant à une des classes (de votre choix), et un des trait caractéristique, de sorte qu'on sera en dimension $d = 1$ et produisez un unique graphique (à l'aide de la fonction plot) comportant :

   (a) les points du sous-ensemble de données (affichés sur l'axe des x)

   (b) une courbe de la densité estimée par votre estimateur paramétrique Gaussien

   (c) une courbe de la densité estimée par l'estimateur de Parzen avec un hyper-paramètre σ (écart type) trop petit

   (d) une courbe de la densité estimée par l'estimateur de Parzen avec un hyper-paramètre σ un peu trop grand

   (e) une courbe de la densité estimée par l'estimateur de Parzen avec un hyper-paramètre σ que vous jugerez plus approprié.
   Utilisez une couleur différente pour chaque courbe, et munissez votre graphique d'une légende claire.

4. Densités 2D : Ajoutez maintenant un second trait caractéristique d'iris, afin d'avoir des entrées en dimension $d = 2$ et produisez 4 graphiques, chacun affichant les points du sous-ensemble de données (avec la fonction plot), et les lignes de contours de la densité estimée (à l'aide de la fonction contour) suivante :

   (a) par votre estimateur paramétrique Gaussien diagonal

   (b) par l'estimateur de Parzen avec un hyper-paramètre σ (écart type) trop petit

   (c) par l'estimateur de Parzen avec un hyper-paramètre σ un peu trop grand

   (d) par l'estimateur de Parzen avec un hyper-paramètre σ que vous jugerez plus approprié.

4 Partie pratique : classifieur de Bayes

1. Mélangez les exemples de Iris (utilisez numpy.random.shuffle après avoir initialisé le générateur aléatoire comme suit numpy.random.seed(123)). Puis divisez l'ensemble de tous les exemples en 2 : un ensemble d'entraînement, et un ensemble de validation. Préparez deux versions de chacun de ces ensembles : une version complète comportant les $d = 4$ traits caractéristiques. Et une version avec seulement les $d = 2$ premiers traits caractéristiques qu'on utilisera pour fins de visualisation.
2. **Classifieur de Bayes basé sur des densités paramétriques Gaussiennes diagonales**

   (a) Implémentez l'algorithme de classifieur de Bayes basé dur des densités paramétriques Gaussiennes diagonales

   (b) Visualisation en dimension $d = 2$. Considérant seulement les deux premiers traits caractéristiques d'iris, entrainez votre classifieur de Bayes sur votre ensemble d'entrainement ; affichez un graphique avec les régions de décision obtenues (ainsi que les points des ensembles d'entrainement et de validation).

   (c) Calcul des erreurs en dimension $d = 2$: calculez et affichez le taux d'erreur de votre classifieur (entrainé sur les 2 premier traits caractéristiques), à la fois sur l'ensemble d'entrainement et de validation.

(d) Calcul des erreurs en dimension $d = 4$: Entrainez votre classifieur en utilisant tous les traits caractéristiques. Puis calculez et affichez le taux d'erreur de votre classifieur (entrainé sur tous les traits caractéristiques), à la fois sur l'ensemble d'entrainement et de validation.

3. **Classifieur de Bayes basé sur des densités de Parzen avec noyau Gaussien** isotropique

   (a) Implémentez l'algorithme de classifieur de Bayes basé dur des densités de Parzen avec noyau Gaussien isotropique

   (b) Visualisation en dimension $d = 2$. Considérant seulement les deux premiers traits caractéristiques d'iris, entrainez votre classifieur de Bayes sur votre ensemble d'entrainement ; affichez un graphique avec les régions (surface) de décision obtenues (ainsi que les points des ensembles d'entrainement et de validation). Produisez 3 tels graphiques de régions de décision : un avec un σ trop petit, trop grand, et approprié

   (c) Courbes d'apprentissage avec $d = 2$. Calculez le taux d'erreur de classification, à la fois sur l'ensemble d'entrainement, et sur l'ensemble de validation, en fonction des valeurs de l'hyper-paramètre σ (calculez-les pour une centaine de valeurs différentes de cet hyper-paramètre afin de pouvoir afficher la courbe). Indiquez la meilleure valeur de l'hyper-paramètre σ que vous avez trouvé.

   (d) Courbes d'apprentissage avec $d = 4$. On va maintenant utiliser tous les traits caractéristiques. Calculez le taux d'erreur de classification, à la fois sur l'ensemble d'entrainement, et sur l'ensemble de validation, en fonction des valeurs de l'hyper-paramètre σ (calculez-les pour une centaine de valeurs différentes de cet hyper-paramètre afin de pouvoir afficher la courbe). Indiquez la meilleure valeur de l'hyper-paramètre σ que vous avez trouvé.

4. D'après ces expériences, pour le problème de classification d'Iris (et pour cette division entrainement/validation particulière) indiquez quel est le meilleur choix d'algorithme entre classifieur de Bayes avec Gaussiennes diagonales, et fenêtres de Parzen, et des autres hyper-paramètres : dimension de l'entrée (2 ou 4), et σ (s'il y a lieu). Précisez les taux d'erreurs de classification qu'ils permettent d'atteindre.