Introduction à l'apprentissage machine (GIF-4101 / GIF-7005)
Département de génie électrique et de génie informatique
Automne 2018

UNIVERSITÉ LAVAL

Devoir 1

Instructions :

- Formation des équipes :
  - GIF-4101 : le devoir est réalisé en équipe de deux à trois étudiants
  - GIF-7005 : le devoir est réalisé individuellement
  - Les équipes doivent être formées dans monPortail avant le cours du 26 septembre
- Programmation :
  - Utilisez Python et scikit-learn autant que possible
  - Produisez vos solutions dans le fichiers fournis, en respectant les instructions
  - La performance attendue et le temps de calcul approximatif requis sont vérifiés dans le code, tout écart trop important par rapport à ces valeurs attendues entraînera *la note de zéro (0)* pour la sous-question correspondante
- Remise :
  - La remise du rapport et du code source se fait dans monPortail
  - Le remise doit être effectuée au plus tard le mercredi 3 octobre à 9h30
- Pondération :
  - Ce devoir compte pour 5% de la note finale

---

1. **Estimateurs statistiques (5pt)**

   Soit la loi exponentielle, pour laquelle la densité de probabilité est définie par l'équation suivante :

   $$p(x) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0 \end{cases} .$$

   L'espérance mathématique de cette loi est $\mathbb{E}[x] = \frac{1}{\lambda}$, alors que sa variance est $\mathrm{Var}(x) = \frac{1}{\lambda^2}$.

   (a) En suivant la démarche présentée en classe, calculez l'expression de l'estimateur de λ selon un maximum de vraisemblance.

   (b) Déterminez si cet estimateur de λ est biaisé dans le cas général.

   Indice : Vous pouvez considérer que $\mathbb{E}[1/x] \neq 1/\mathbb{E}[x]$ dans le cas général.

2. **Expérimentations avec scikit-learn (10pt)**

   Les Iris de Fisher forment un jeu de données classique en apprentissage automatique, qui a été utilisé la première fois en 1936 par R. A. Fisher pour illustrer sa méthode d'analyse discriminante linéaire. Chaque donnée du jeu comporte quatre mesures, soit la longueur et largeur de sépales, et la longueur et la largeur de pétales, et ce de trois variétés d'iris : Iris Setosa, Iris Versicolore (l'emblème floral du Québec) et Iris Virginia. Les données proviennent d'iris récoltés en Gaspésie.

Le jeu des Iris de Fisher disponible par la fonction `datasets.load_iris` de scikit-learn. Faites les manipulations suivantes avec scikit-learn en utilisant le fichier `d1q2.py` [1], joignez les résultats obtenus à votre rapport et fournissez le fichier source modifié de votre solution. Appuyez autant que possible vos discussions par des arguments quantitatifs et évitez le verbiage.

(a) Produisez des graphiques pour votre rapport représentant le jeu de données en 2D avec indicateurs de classe, pour quelques paires de mesures (longueur des sépales vs longueur des pétales, longueur vs largeur des sépales, etc.), afin de bien visualiser les données. Discutez brièvement de la distribution des données selon les classes.

(b) Expérimentez avec les classifieurs paramétriques suivants.

i. Classifieur bayésien de loi normale multivariée avec matrices de covariance $\mathbf{\Sigma}_i$ complètes et distinctes pour chaque classe.

ii. Classifieur bayésien de loi normale multivariée avec matrice de covariance $\mathbf{\Sigma}$ complète et partagée entre chaque classe.

iii. Classifieur bayésien de loi normale multivariée avec matrices de covariance $\mathbf{\Sigma}_i$ diagonales ($\sigma_{i,j} = 0, \forall i \neq j$) et distinctes entre les classes.

iv. Classifieur bayésien de loi normale multivariée avec matrice de covariance isotropique, soit $\mathbf{\Sigma} = \sigma^2\mathbf{I}$ avec valeurs égales sur la diagonale ($\sigma_j^2 = \sigma^2$) et nulles hors de la diagonale ($\sigma_{j,k} = 0, \forall j \neq k$), avec également un partage de la matrice de covariance entre chaque classe et des probabilités a priori égales pour chaque classe ($P(C_i) = P(C_j), \forall i,j$).

Pour chaque classifieur testé, donnez l'erreur empirique correspondant au taux d'erreur de classement sur le jeu de données au complet (erreur sur le jeu d'entraînement) avec chacune des paires de mesures possibles. Représentez également les résultats visuellement, en traçant les données (avec indicateurs de classe) et les régions de décision dans des figures 2D, pour quelques-unes de ces paires de mesures. À la lumière des résultats obtenus, déterminez le classifieur testé qui semble avoir le niveau de complexité le plus approprié pour ce jeu de données.

(c) Comparez et discutez les résultats obtenus selon trois méthodologies expérimentales :

i. Erreur empirique rapportée sur le jeu entier (erreur sur le jeu d'entraînement).

ii. Partition aléatoire en un jeu d'entraînement (50 %), utilisé pour évaluer les paramètres des distributions, et de validation (50 %), utilisé pour calculer l'erreur en généralisation. Répétez les expériences 10 fois, à chaque fois avec des partitions aléatoires entraînement/validation distinctes. Rapportez l'erreur en généralisation moyenne.

iii. Évaluation des performances selon une méthodologie de validation à k plis, en utilisant $k = 3$ plis. Faites 10 répétitions de l'expérience, avec un partitionnement différent à chaque fois, et rapportez le taux d'erreur moyen.

Pour ces expérimentations, limitez-vous à un classifieur bayésien de loi normale multivariée avec matrices de covariance $\mathbf{\Sigma}_i$ complètes et distinctes pour chaque classe.

(d) Créez maintenant un nouveau jeu de données en utilisant la fonction `make_circles` avec l'argument `factor=0.3`. Testez les quatre classifieurs mentionnés en (b) sur ce jeu de données, en utilisant une partition aléatoire (mais identique pour tous les classifieurs) de 50% en entraînement et 50% en validation. Pour chaque classifieur testé, donnez l'erreur empirique correspondant au taux d'erreur de classement sur le jeu de validation. Représentez également les résultats visuellement, en traçant les données (avec indicateurs de classe) et les régions de

1. `http://vision.gel.ulaval.ca/~cgagne/enseignement/apprentissage/A2018/travaux/d1q2.py`

décision dans des figures 2D. Comment expliquez-vous la différence de performance entre les différentes approches ? Quel classifieur semble avoir le niveau de complexité le plus approprié pour ce jeu de données ?

3. **Classement avec option de rejet (5pt)**

   Soit un classifieur bayésien de loi normale multivariée avec matrices de covariance distinctes pour chaque classe et isotropiques, c'est-à-dire avec des valeurs égales sur toute la diagonale et nulles autrement, $\mathbf{\Sigma}_i = \sigma_i^2 \mathbf{I}, \forall i$.

   (a) Calculez l'équation pour l'estimation du paramètre σ_i par la méthode du maximum de vraisemblance, en fournissant les développements mathématiques complets dans votre rapport.

   (b) Supposons maintenant que l'on ajoute une option de rejet au classement. Le coût des erreurs est égal pour tous les types d'erreurs (coût de 1), sauf pour le rejet (coût de $\lambda \in [0,1]$). Donnez l'équation complète pour calculer la probabilité a posteriori $P(C_i|\mathbf{x})$ et la fonction pour la prise de décision minimisant le risque (minimisant le coût) avec l'option de rejet.

   (c) Faites une implémentation du modèle présenté au point précédent en utilisant l'interface scikit-learn, permettant ainsi de l'utiliser similairement aux autres algorithmes disponibles dans la librairie. Implémentez les méthodes `fit`, `predict`, `predict_proba` et `score` dans votre modèle. Pour la fonction `score`, utilisez le coût total de l'application de votre classifieur sur les données (somme du coût des rejets et du coût des mauvais classements). Utilisez le fichier `d1q3.py` [2] pour faire votre implémentation et joignez le code source résultant à votre remise.

   (d) Utilisez le jeu des Iris de Fisher pour tester l'algorithme que vous avez implémenté au point précédent. Pour ce faire, exécutez l'algorithme en variant le coût de rejet. Testez avec les coûts de rejet suivants : $\lambda = \{0{,}1; 0{,}3; 0{,}5; 1.0\}$. Pour chaque configuration, rapportez l'erreur empirique correspondant au taux d'erreur de classement sur le jeu de données au complet (erreur sur le jeu d'entraînement). Représentez également les résultats visuellement, en traçant les données (avec indicateurs de classe), les régions de décision dans des figures 2D, incluant les régions de rejet, pour quelques paires de variables. Comparez le risque des classifieurs avec rejet à celui qui ne rejette pas de données ($\lambda = 1$). Fournissez votre solution dans le fichier `d1q3.py`.

18/09/2018 (révision 19/09/2018, 22/09/2018, 23/09/2018)
CG & MAG

2. `http://vision.gel.ulaval.ca/~cgagne/enseignement/apprentissage/A2018/travaux/d1q3.py`