Master 2 IWOCS
Apprentissage
Technique de classification supervisée bayésienne naive
Application à la détection de maladies cardiaques

Rodolphe Charrier

Dec. 2019

Classification avec une approche bayésienne "naïve"

Dans cette application, on va partir d'un fichier décrivant les descripteurs utilisés pour diagnostiquer des maladies cardiaques. Il s'agira dans un premier temps à apprendre les corrélations existants entre ces descripteurs et l'occurrence d'une maladie cardaique selon une approche bayésienne naïve. L'ensemble des données est issu d'une étude menée à Cleveland sur 303 cas dont on connaît les diagnostics. Dans un second temps, on testera cet apprentissage sur un second jeu de données du même type provenant d'une étude menée en Hongrie. Ces données sont issues du site (`http://archive.ics.uci.edu/ml/datasets/Heart+Disease`).

Données

Les données relatives à cette application sont à télécharger sur le cours Eureka :

1. les données d'apprentissage sont dans le fichier *processed.cleveland.data*
2. les données de test sont dans le fichier *reprocessed.hungarian.data*
3. les explications sur ces données sont dans le fichier *heart-disease.names*. Seuls les 14 colonnes de la section 7 de ce document sont à considérer.

Voici les 14 descripteurs utilisés dans les fichiers de données :

```
Attribute Information:
   -- Only 14 used
      -- 1. #3  (age)
      -- 2. #4  (sex)
      -- 3. #9  (cp)
      -- 4. #10 (trestbps)
      -- 5. #12 (chol)
      -- 6. #16 (fbs)
      -- 7. #19 (restecg)
      -- 8. #32 (thalach)
      -- 9. #38 (exang)
      -- 10. #40 (oldpeak)
      -- 11. #41 (slope)
      -- 12. #44 (ca)
      -- 13. #51 (thal)
      -- 14. #58 (num)       (the predicted attribute)
```

où les chiffres après le "#" renvoient à un des 76 attributs des fichiers initiaux répertoriés en fin du fichier *heart-disease.names*.
Nous n'utiliserons pas tous ces paramètres ici. Il s'agira donc dans un premier temps de transformer ces données pour les adapter à l'approche bayésienne. Dans la liste ci-dessus, on ne gardera en fait que les attributs suivants :

```
Attribute Information:
   -- Only 14 used
      -- 1. #3  (age)
      -- 2. #4  (sex)
      -- 3. #9  (cp)
      -- 6. #16 (fbs)
      -- 7. #19 (restecg)
      -- 9. #38 (exang)
      -- 14. #58 (num)       (the predicted attribute)
```

Le dernier attribut (num) est l'attribut à prédire par notre algorithme bayésien. Dans ce fichier, il possède plusieurs valeurs possibles de 0 à 4 en fonction du niveau de pathologie : 0 signifie aucune pathologie, les valeurs strictement positives seront considérées comme associées à une pathologie. Il faudra donc modifier les valeurs de cette colonne pour n'avoir que des 0 (absence de maladie) ou des 1 (présence de maladie). Ce n'est pas la seule colonne qu'il faudra modifier parmi ces 6 attributs : pour adapter les données à l'algorithme bayésien naïf, l'idéal est d'avoir des colonnes seulement avec des 0 ou des 1 :

- Ainsi, pour l'attribut "age", nous séparerons l'échantillon en deux parties : les moins de 50 ans (valeur 0) et les plus de 50 ans (valeur 1)
- pour l'attribut "cp", qui correspond à une évaluation des symptômes pneumologiques, on a 4 valeurs/modalités de 1 à 4, chacune ayant une importance par rapport au diagnostic. Il faut donc les "séparer" en 4 indicateurs distincts. On créera donc à partir de cet attribut/colonne, quatre nouveaux attributs/colonnes, une colonne par modalité (une colonne pour la présence des valeurs 1, une pour les valeurs 2, ...)
- il faudra faire de même pour l'attribut "restecg" qui possède lui aussi trois modalités de 0 à 2 relatives à 3 types de symptômes observés sur les signaux d'électrocardiogrammes. Il faudra donc ici également produire 3 colonnes distinctes pour la présence ou l'absence de ces trois symptômes dans les données.

Au final, on devra obtenir 11 attributs d'observations, avec en plus l'attribut à prédire si la maladie est présente ou pas, chaque attribut ayant des 0 ou des 1. On se retrouve donc dans la situation vue en TP, où le dictionnaire de mots est ici réduit à ces 11 attributs associés aux symptômes, et le fait d'être un spam est remplacé par le fait que la maladie est diagnostiquée.

On doit donc, avant toute chose, constituer la matrice creuse contenant les données d'apprentissage issues du fichier fourni *processed.cleveland.data*. Celle-ci devrait avoir une taille $303x12$ si on intègre la colonne (num) des décisions. Une possibilité de script python pour lire ces données est la suivante qui permet de lire les fichiers csv :

```
totData = np.genfromtxt("processed.cleveland.data", dtype=float,delimiter=",")
```

Utilisation de l'algorithme bayésien naif

Phase d'apprentissage

1. Commencer par calculer ϕ_y où y est la variable indiquant s'il y a une maladie.
2. Calculer $\phi_{k|y=1}$ pour chaque descripteur k lorsqu'il y a maladie et stocker l'ensemble dans un vecteur
3. de même pour $\phi_{k|y=0}$

Les equations de calcul sont les mêmes que celles données dans le tp sur les spams.

Phase de test et d'évaluation

Poursuivre le script python pour l'évaluation de l'apprentissage à partir du fichier de tests *reprocessed.hungarian.data*. Charger les données de la même façon que pour le fichier des exemples d'apprentissage, avec un peu moins de données.
A partir des statistiques obtenues en phase d'apprentissage calculer sur chaque enregistrement de test :

1. $\log(\phi_{x|y=1}) + \log(p(y=1))$
2. puis $\log(\phi_{x|y=0}) + \log(p(y=0))$

3. et conclure en comparant les deux dans chaque cas.

Puis calculer le taux d'erreurs de classification en comparant vos résultats avec ceux de la dernière colonne du fichier de tests qui donne la présence ou absence de maladie. Il s'agit alors d'établir la matrice de confusion :

↓ Classé/Vrai classe →	positif	négatif
positif	VP	FP
négatif	FN	VN
total	P	N

et de calculer les taux de positifs bien classés : $\frac{VP}{P}$
et de négatifs mal classés : $\frac{FP}{N}$.
Afficher ces taux en sortie standard.

Ajustement de la taille de l'ensemble d'apprentissage

Dans un second script python, pour analyser l'effet de la taille de l'ensemble d'apprentissage sur le taux d'erreurs de classification, prendre 100 exemples de tests comme exemples supplémentaires d'apprentissage.
Exécuter la procédure d'apprentissage précédente et comparer les nouveaux taux d'erreurs de classification.