Algorithme des k plus proches voisins

Évaluation et sélection de modèle

Marie Szafranski

03 octobre 2022

Crédits. Ce TP est l'adaptation d'un support de Gilles Gasso (INSA Rouen).

1 Préambule [45 minutes max]

Les données ainsi que des *notebooks* d'exemples sont disponibles sur le site de l'UE.

Données. Le jeu de données `pima_data.csv` recense des informations concernant 768 femmes d'au moins 21 ans d'origine amérindienne. Les 8 caractéristiques de la matrice d'observations `x` sont :

1. Le nombre de grossesses.
2. Le taux de glycémie 2 heures après l'absorption d'une solution sucrée (g de glucose par L de sang).
3. La pression sanguine diastolique ($mmHg$).
4. L'épaisseur du pli cutané au niveau du triceps (indicateur de masse graisseuse, mm).
5. La mesure de l'effet de l'insuline (muU/ml).
6. L'indice de masse corporelle (kg/m^2)
7. Le résultat d'une fonction calculant le risque de développer un diabète qui prend en compte l'hérédité.
8. L'âge.

Pour chacune de ces femmes, la classe représentée par la dernière colonne du fichier `pima_data.csv` concerne la présence (1) ou l'absence (-1) de diabète.

Chargement et examen des données. Utilisez le notebook fourni pour charger ces données et commenter les résultats.

`load_pima.ipynb`

Normalisation des données. `sklearn.preprocessing.StandardScaler`

Les caractéristiques permettant de décrire les données présentées ci-dessus utilisent des unités différentes. Il est courant de *normaliser* les données pour rendre la comparaison de ces différentes caractéristiques plus homogène. On notera `X` le tableau de données relatif aux observations et `Y` le vecteur relatif aux étiquettes. Pour chaque vecteur de variable de `X` :

— On doit soustraire à chaque élément de l'échantillon considéré la moyenne empirique d'un ensemble de référence sur la caractéristique,

— On doit diviser chaque élément de l'échantillon considéré par l'écart-type empirique d'un ensemble de référence sur la caractéristique.

Sur ce TP, on travaillera avec les données normalisées. Il faudra donc, quelle que soit la stratégie utilisée, normaliser chaque jeu de données en fonction d'un ensemble référence inclus dans l'ensemble des jeux de données construits.

Calcul de mesures d'erreur. Cette partie est purement illustrative et a pour objectif de vous faire manipuler les principales fonctions qui seront utilisées dans le TP. Elle ne correspond pas à la mise en place d'une procédure d'apprentissage rigoureuse.

P1. Séparez le jeu de données global `(X, Y)` en deux jeux `(X_1, Y_1)` et `(X_2, Y_2)` avec un ratio de `1/2`. L'ensemble de référence sera `(X_1, Y_1)`. Comment tenir compte de l'équilibre des classes dans cette répartition ?

`sklearn.model_selection.train_test_split`

P2. Normalisez les deux jeux de données selon les principes du paragraphe précédent.

`sklearn.preprocessing.StandardScaler`

P3. Appliquez l'algorithme des k plus proches voisins en fixant k à 5 puis à 15 : `sklearn.neighbors.KNeighborsClassifier`

(a) De façon à prédire `Y_1` à partir de `(X_1, Y_1)`, (évaluation en apprentissage)

(b) De façon à prédire `Y_2` à partir de `(X_1, Y_1)`. (évaluation en test)

Vous évaluerez à chaque fois l'erreur de classification, la précision, le rappel, et l'AUC-ROC. Que constatez vous ?

`sklearn.metrics.*`

2 Mise en place d'une procédure d'apprentissage

On parlera d'*entraînement du modèle* lorsqu'on appliquera l'algorithme sur un ensemble d'apprentissage pour des valeurs de k différentes. On parlera de *sélection de modèle* lorsque l'on choisira k à partir de l'erreur de validation obtenue sur un ensemble de validation, soit dans une procédure par découpage (stratégie 1), soit dans une procédure de validation-croisée (stratégie 2). Dans tous les cas, l'ensemble de test ne devra être utilisé *qu'une seule fois* pour évaluer l'erreur de test. On travaillera sur cette partie avec l'erreur de classification.

Dans un premier temps, séparez le jeu de données `(X, Y)` en deux ensembles `(X_av, Y_av)` et `(X_t, Y_t)` avec un ratio de `1/2`. Pensez à garder une copie de ces jeux données bruts.

Stratégie 1 : ensemble d'apprentissage, de validation et de test. On recherchera le nombre de voisins optimal dans l'ensemble de paramètres `p_k = {1, 5, 10, 15, 20, 25}`.

S1.1. Séparez l'ensemble `(X_av, Y_av)` en un ensemble d'apprentissage `(X_a, Y_a)` et un ensemble de validation `(X_v, Y_v)` avec un ratio de `2/3`.

S1.2. Normalisez les jeux de données `(X_a, Y_a)`, `(X_v Y_v)` et `(X_t, Y_t)` selon le principe décrit dans la partie 1. Quel doit être l'ensemble de référence ?

S1.3. Appliquez l'algorithme des k plus proches voisins, pour chaque valeur de `p_k`, de sorte à pouvoir évaluer l'erreur de classification en apprentissage et en validation.

S1.4. Tracez sur un même graphique l'erreur de classification obtenue en apprentissage et en validation en fonction de `p_k` et commentez les résultats.

S1.5. Sélectionner le meilleur modèle et évaluez l'erreur de test.

Remarque. *On peut réitérer la procédure précédente avec `B=10` tirages différents entre les jeux de données d'apprentissage, de validation et de test, afin d'avoir des résultats plus robustes.*

Stratégie 2 : validation-croisée pour la sélection de k. On recherchera le nombre de voisins optimal dans l'ensemble de paramètres `p_k = {1, 5, 10, 15, 20, 25}`.

S2.1. Normalisez les jeux de données `(X_av, Y_av)` et `(X_t, Y_t)` selon le principe décrit dans la partie 1. Quel doit être l'ensemble de référence ?

S2.2. À partir de `(X_av, Y_av)`, regardez comment construire `K = 5` sous-ensembles au sein de votre classifieur. Comment tenir compte de l'équilibre des classes dans cette répartition ? `sklearn.model_selection.cross_val_score`

S2.3. Pour chaque valeur de `p_k`, appliquez l'algorithme des k plus proches voisins dans une procédure de validation croisée.

S2.4. Tracez sur un même graphique l'erreur de classification moyenne sur les `K = 5` blocs obtenue en apprentissage et en validation en fonction de `p_k` et commentez les résultats.

S2.5. Déterminez le meilleur modèle et évaluez l'erreur de test.

3 Exercice noté

Jeu de données. Analysez le jeu de données disponible ici :

`https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set`

Adaptation des métriques d'évaluation.

1. Vous constaterez que ce problème contient 3 classes à prédire. Certaines métriques d'évaluation sont définies pour des problèmes de classification binaire. Une réflexion et une recherche sur l'extension de ces métriques à des problèmes de plus de 2 classes seront sans doute appropriées.
2. Il pourrait-être aussi opportun de réfléchir à comment intégrer une notion de coût sur les classes prédites : il est sans doute plus problématique de prédire qu'une personne à un faible risque d'avoir des complications pendant sa grossesse alors qu'en réalité elle a un fort risque. (facultatif)

Algorithme des C-moyennes. Nous comparerons dans ce TP l'algorithme des k-ppv avec celui des C-moyennes [1], dans une optique supervisée : on supposera pour évaluer l'algorithme des C-moyennes que le nombre de classe et l'appartenance des individus aux classes sont connus (bien qu'ils ne soient pas utilisés comme information dans l'algorithme.

Rendus. Sur le dépôt `mlds_1` du site `http://exam.ensiie.fr`, vous devrez rendre une archive, au format `.zip` ou `.tar.gz` et inférieure à 3 Mo, contenant :

R.1 Un notebook `python` (ou `R`) ayant permis l'analyse.

R.2 Un rapport ***au format pdf*** de ***3 pages maximum***.

Le rapport devra contenir :

— Une **introduction** décrivant les objectifs et présentant une analyse statistique descriptive *succincte et commentée* des données. (1 page maximum)

— Une comparaison dans un cadre supervisé des algorithmes k-ppv et C-moyennes sur ces données. (2 pages maximum)

  — Une ou deux phrases *concises mais précises* précisant la stratégie mise en œuvre. Il ne s'agit pas de recopier l'énoncé.

  — Vous ferez une présentation *claire et commentée* des résultats obtenus. Sur chaque graphique, il doit y avoir un titre, les axes et les courbes doivent êtres identifiés. Les graphiques doivent êtres lisibles s'ils sont imprimés en noir et blanc. Chaque figure doit contenir une légende explicite.

— Une **conclusion** permettant d'établir les *avantages* et les *limites* de chaque méthode. (1 page maximum)

Remarque. Attention :

— Aucun rendu par mail ne sera pris en compte.

— La note des archives rendues dans d'autre format, par exemple `.rar`, sera pénalisée.

— La note des archives de taille supérieure à 3 Mo sera pénalisée.

Remarque. Pour les étudiants ayant pris en compte l'intégration d'une notion de coût dans la métrique d'évaluation, une page supplémentaire peut être intégrée au rapport.

1. Vous trouverez une description de cet algorithme dans le livre de C.-A. Azencott disponible sur : `http://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf`.