Université de Poitiers
Machine Learning
M2 Info, M2 IOI
2022–2023

Compréhension des algorithmes

1 Programmation d'une méthode de discrimination : Kppv

Vous allez créer un programme permettant d'étudier l'un des algorithmes de discrimination étudiés durant le cours.

```
[clas]=kppv(apprent,classe_origine,k,x);
```

Ce programme applique la méthode de discrimination de kppv sur un ensemble d'individu élément de R^2.
Les paramètres :

- La variable `apprent` est une matrice qui doit contenir les différents individus de l'ensemble d'apprentissage rangés par colonne. Le nombre de ligne est 2 et le nombre de colonne est N_1.
- La variable `classe_origine` est un vecteur qui indique la classification de l'ensemble d'apprentissage. `classe_origine[i]` indique le numéro de la classe de l'individu `apprent[,i]`.
- La variable `k` indique le nombre de voisins utilisés dans l'algorithme.
- La variable `x` est une matrice qui doit contenir les différents individus à classer rangés par colonne. Le nombre de ligne est 2 et le nombre de colonne est N_2.

Le résultat :

- La variable `clas` est un vecteur qui indique le résultat de l'algorithme de la discrimination. `clas[i]` indique le numéro de la classe de l'individu `x[,i]`.

— *Réaliser la fonction.*
Vous allez maintenant tester votre algorithme.
— *Charger des données* `p1_test.mat`.
Sous Python, `Data = loadmat(name)`.
Ce fichier comprend trois variables :

- La matrice `test` qui contient l'ensemble d'apprentissage constitué de 3 classes de 50 individus chacunes $\in R^2$ suivant 3 distributions gaussiennes. Un "professeur" nous a indiqué que les 50 premiers individus sont issus de la classe 1, les 50 suivants de la classe 2, les 50 derniers de la classe 3.
- La matrice `x` qui contient un ensemble de 300 individus à classer.
- Le vecteur `clasapp` contient le résultat parfait que l'on doit obtenir par cette discrimination.

— *Appliquer la méthode de discrimination Kppv (pour $k = 1, 3, 5, 7, 13, 15$). Pour chacun des tests calculer le taux d'erreur. Vous pouvez visualiser le résultat des classifications par la fonction affiche_classe.*
— *Commenter les résultats.*

2 Comparaison des deux méthodes (paramétriques et non paramétriques)

Par la suite vous aller comparer les deux techniques vues en corus à savoir Bayes Naïf et Kppv en utilisant le package `Scikit-Learn`.

2.1 Influence de la taille de l'ensemble d'apprentissage : taille réduite

— *Charger des données* `p1_petit.xlsx`).
Ce fichier comprend deux onglets :

- [Ensemble d'apprentissage] qui contient l'ensemble d'apprentissage constitué de 3 classes de 20 individus chacune en dimension 2 suivant 3 distributions gaussiennes. Un "professeur" nous a indiqué que les 20 premiers individus sont issus de la classe 0, les 20 suivants de la classe 1, les 20 derniers de la classe 2.

- [Inconnu] qui contient un ensemble de 300 individus à classer. A la suite du tableau de mesures, une ligne est ajoutée (appelée Oracle) qui contient le résultat parfait que l'on doit obtenir par cette discrimination. Bien sûr dans le cadre d'application réelle, cette troisème ligne serait absente mais elle va vous permettre d'évaluer la qualité de votre algorithme.

— *Appliquer la méthode de discrimination paramétrique et Kppv (pour* $k = 1, 3, 5, 7, 13, 15$*). Pour chacun des tests calculer le taux d'erreur.*

— *Commenter les résultats.*

2.2 Influence de la taille de l'ensemble d'apprentissage : taille importante

— *Charger des données* `p1_grand.xlsx)`.

Ce fichier comprend deux onglets :

- [Ensemble d'apprentissage] qui contient l'ensemble d'apprentissage constitué de 3 classes de 450 individus chacune en dimension 2 suivant 3 distributions gaussiennes. Un "professeur" nous a indiqué que les 150 premiers individus sont issus de la classe 0, les 150 suivants de la classe 1, les 150 derniers de la classe 2.
- [Inconnu] qui contient un ensemble de 300 individus à classer. A la suite du tableau de mesures, une ligne est ajoutée (appelée Oracle) qui contient le résultat parfait que l'on doit obtenir par cette discrimination. Bien sûr dans le cadre d'application réelle, cette troisème ligne serait absente mais elle va vous permettre d'évaluer la qualité de votre algorithme.

— *Appliquer la méthode de discrimination paramétrique et Kppv (pour* $k = 1, 3, 5, 7, 13, 15$*). Pour chacun des tests calculer le taux d'erreur.*

— *Commenter les résultats.*

2.3 Absence de professeur

Dans ce cas, nous n'avions pas de spécialiste pour labelliser l'ensemble d'apprentissage. Nous avons donc appliqué un algorithme de clustering (içi Kmeans) pour extraire des groupes. Dans ce cas, l'ensemble d'apprentissage est imparfait, il présente des erreurs.

— *Charger des données* `p1_Kmean.xlsx)`.

Ce fichier comprend deux onglets :

- [Ensemble d'apprentissage] qui contient l'ensemble d'apprentissage constitué de 3 classes, avec en tout 150 individus en dimension 2 suivant 3 distributions gaussiennes. A la suite du tableau de mesures, une ligne est ajoutée (appelée Classe) qui contient le résultat du clustering par l'algorithme Kmeans.
- [Inconnu] qui contient un ensemble de 300 individus à classer. A la suite du tableau de mesures, une ligne est ajoutée (appelée Oracle) qui contient le résultat parfait que l'on doit obtenir par cette discrimination. Bien sûr dans le cadre d'application réelle, cette troisème ligne serait absente mais elle va vous permettre d'évaluer la qualité de votre algorithme.

— *Appliquer la méthode de discrimination paramétrique et Kppv (pour* $k = 1, 3, 5, 7, 13, 15$*). Pour chacun des tests calculer le taux d'erreur.*

— *Commenter les résultats.*

2.4 Distribution inconnue

Dans ce cas, les mesures associés aux individus observés ne suivent plus une distribution Gaussienne.

— *Charger des données* `p1_NonGaussien.xlsx)`.

Ce fichier comprend deux onglets :

- [Ensemble d'apprentissage] qui contient l'ensemble d'apprentissage constitué de 3 classes, avec en tout 210 individus en dimension 2 suivant 3 distributions inconnues. A la suite du tableau de mesures, une ligne est ajoutée (appelée Classe) qui contient le résultat du clustering par l'algorithme Kmeans.
- [Inconnu] qui contient un ensemble de 300 individus à classer. A la suite du tableau de mesures, une ligne est ajoutée (appelée Oracle) qui contient le résultat parfait que l'on doit obtenir par cette discrimination. Bien sûr dans le cadre d'application réelle, cette troisème ligne serait absente mais elle va vous permettre d'évaluer la qualité de votre algorithme.

— *Appliquer la méthode de discrimination paramétrique et Kppv (pour* $k = 1, 3, 5, 7, 13, 15$*). Pour chacun des tests calculer le taux d'erreur.*

— *Commenter les résultats.*

Université de Poitiers
Machine Learning
M2 Info, M2 IOI
2021–2022

Classification d'images

L'objectif de ce mini-projet est de réaliser un système discrimination d'images réelles. La base de données d'images est une base de données classique de test (base Wang) et est composée de 10 types d'images : Jungle, Plage, Monuments, Bus, Dinosaures, Eléphants, Fleurs, Chevaux, Montagne et Plats.

Bien sûr à l'intérieur de chaque classe (100 images par classe), les images présent une forte variabilité. Je vous propose de mettre en place une application de classification. Dans le cadre du système de classification, notre système de base de données d'images fonctionnera de la façon suivante :

— l'utilisateur proposera au système une nouvelle image "inconnue" ;
— le système affichera alors la classe d'appartenance de l'image.

Afin de mettre en place cette classification, nous devons mesurer des paramètres sur chaque image afin de caractériser ces différentes images.

Vous allez développer deux approches, l'une reposant sur des descripteurs "Image", l'autre suivant une stratégie Deep donc à base de descripteurs CNN.

3 Approche basée Descripteurs

3.1 Calcul des descripteurs

Cette étape n'est pas forcement complexe mais nécessite du temps de développement et des connaissances en traitement d'images. Dans le cadre de ce projet, nous vous fournissons directement les mesures de caractéristiques préalablement faites sur les images de la base Wang dans un fichier Excel.

Nous proposons d'utiliser 5 ensembles de mesures :

— JCD,
— PHOG,
— CEDD,
— FCTH,
— Fuzzy Color Histogram.

Pour chacun de ces vecteurs de mesures devant caractériser chaque image, nous fournissons l'article fondateur expliquant le principe de ces mesures. Elles vont mesurées des paramètres de couleur, de géométrie, de texture ...

Nous faisons ces 5 types de mesures pour chacune des images de la base, ainsi nous obtenons 5 tableaux de mesures ayant un nombre de colonnes variable selon le descripteur et comprenant 1000 lignes pour les 1000 images. Les mesures sont stockées dans le fichier `wang.xlsx`.

Ce fichier contient un onglet par type de mesure. Attention à l'ordre de rangement des images.

La base est donc composée de 10 types d'images : Jungle, Plage, Monuments, Bus, Dinosaures, Eléphants, Fleurs, Chevaux, Montagne et Plats. Comme vous pouvez le constater en visualisant les images, les classes sont regroupées à travers leur numéro (par exemple de 200 à 299, ce sont les images "Monuments").

— Importer les différents tableaux de mesures ; c
— Créer un vecteur de label indiquant la classe de chaque image sous forme d'un nombre : la classe de l'image de nom `i.jpg` est contenue dans `label[i]`.

3.2 Mise en place d'un système de classification

Nous proposons de mettre en place un système de classification qui doit déterminer à partir d'un ensemble d'apprentissage la catégorie de l'image inconnue.

Vous allez extraire aléatoirement quelques images de la base de données ainsi que les mesures associées. Ces images tests vont constituer vos images "inconnues", le reste du tableau constitue l'ensemble d'apprentissage.

— Mettre en place un système de classification qui pour la présentation d'une image inconnue et de son vecteur de mesures associé propose une classe (Jungle, Plage, Monuments, Bus, Dinosaures, Eléphants, Fleurs, Chevaux, Montagne et Plats).
— Tester la procédure de classification avec les 5 types de mesures et différentes images inconnues.
— Analyser les résultats (matrice de confusion, taux d'erreur).
— Comparer les deux techniques (Bayes Naïf, Kppv) en utilisant le package `Scikit-Learn`.