Travail dirigé n^o 5

Reconnaissance de formes : sélection d'attributs et approches neuronales

Instructions

- Les travaux pratiques sont effectués par équipes de deux.
- Le compte rendu doit comporter une réponse concise mais complète à chacune des questions, accompagnée au besoin des courbes, figures et images appropriées ;
- Le compte rendu doit être rédigé à l'aide des fonctionnalités de publication de `matlab` (menu "File / Publish" de l'éditeur `matlab`), en format html ou pdf. L'ensemble des fichiers doit être placé dans une unique archive `zip`.
- Le compte rendu doit être remis au plus tard à minuit le jour de la séance, en utilisant l'outil approprié disponible sur le site web du cours.
- Le travail doit être remis par un seul des membres du groupe. Si tel n'est pas le cas, la version la plus récente du travail remis sera prise en compte.

1 Introduction

L'objet de cette séance de travaux pratiques est de mettre en œuvre certaines des techniques de reconnaissance de formes présentées en cours et d'évaluer leur comportement. Lors de la dernière séance, nous avons utilisé une approche statistique ; cette fois-ci, nous allons nous intéresser plus particulièrement à l'approche neuronale. Notre but est encore la classification automatique de spécimens d'iris de l'ensemble de Fisher. Nous allons par ailleurs tenter diverses approches de transformation de l'espace d'attributs propres à faciliter (possiblement) la classification.

Les données d'entraînement et de classification sont disponibles dans une archive sur le site web du cours, qui contient aussi une fonction de soutien à l'analyse statistique des données d'entraînement et un script de mise en œuvre d'un classificateur basé sur un réseau de neurones à rétropropagation.

Téléchargement et contenu de l'archive

L'archive `TD5.zip` est disponible sur le site web du cours. Récupérez la et extrayez son contenu dans votre répertoire de travail. Vous aurez ainsi accès à la fois au fichier de données et aux fonctions

`matlab` décrits ci-dessous.

Fichiers de données

Les données relatives aux trois conditions expérimentales sont contenues dans le fichier suivant `iris.mat`, identique à celui utilisé lors du TD4.

Fonctions `matlab`

Afin d'effectuer la reconnaissance de forme par réseau de neurones, nous allons essentiellement utiliser des fonctions directement disponibles dans le logiciel `Matlab` ; nous vous fournissons cependant le script `Matlab` `rn.m`. Dans sa forme actuelle, ce script permet de créer et d'entraîner un réseau de neurones propre à la classification des spécimens représentés sur un espace de quatre attributs réels. Le lancement de ce script procède à la classification et à l'affichage des résultats.

2 Sélection des attributs

Le premier but du travail est d'explorer les méthodes de réduction de la dimensionalité de l'espace des attributs vues en cours. Ces techniques réalisent la réduction à l'aide d'une transformation linéaire de chaque spécimen $\boldsymbol{x} \in \mathbb{R}^n$ vers le spécimen de taille réduite $\boldsymbol{y} = \mathbf{T}\boldsymbol{x} \in \mathbb{R}^q$, $q < n$.

Pour l'ensemble de ce travail, on suppose que la distribution des spécimens conditionnelle (respective) à chaque classe est une loi gaussienne multivariée de moyenne $\boldsymbol{m}_k$ et de matrice de covariance $\mathbf{R}_k$.

2.1 Décomposition du minimum d'entropie

En s'appuyant sur l'hypothèse que les objets des diverses classes observent la même matrice de covariance $\mathbf{R}$, cette technique suggère une transformation linéaire qui minimise la dispersion intra-classe des spécimens. L'approche consiste à utiliser comme lignes de $\mathbf{T}$ les m vecteurs propres de $\mathbf{R}$ correspondant aux q plus petites valeurs propres.

Dans le cas des données d'iris, la matrice de covariance pour chacune des classes n'est pas la même. On peut quand même tenter l'usage de cette méthode en approchant $\mathbf{R}$ par la matrice $\mathbf{R}_k$ pour l'une des trois classes.

2.2 Discriminant linéaire de Fisher

Cette approche cherche à maximiser le rapport entre la dispersion intra-classe et la dispersion inter-classe. On définit

$$\boldsymbol{m} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{m}_k \tag{1}$$

$$\mathbf{B} = \sum_{k=1}^{K} (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^{\mathrm{t}} \tag{2}$$

$$\mathbf{R} = \sum_{k=1}^{K} \mathbf{R}_k. \tag{3}$$

Le rapport est maximal pour la transformation linéaire dont les lignes correspondent aux vecteurs propres de $\mathbf{R}^{-1}\mathbf{B}$ associés aux q plus grandes valeurs propres.

2.3 Décomposition en composantes principales

Dans ce cas, l'approche maximise la dispersion globale des spécimens projetés. On l'obtient en prenant pour lignes de $\mathbf{T}$ les vecteurs propres de $\mathbf{R}_x$ qui correspondent aux q plus grandes valeurs propres.

Prenez garde : $\mathbf{R}_x$ est la matrice de covariance de *tous* les éléments d'entraînement, abstraction faite de leur appartenance à l'une ou l'autre classe. Vous la calculez donc en tenant compte de toutes les données d'entraînement mélangées en un seul ensemble.

2.4 Travail à effectuer

Implantez ces trois techniques de réduction de dimensionalité de manière à réduire le nombre d'attributs des spécimens d'iris à deux. Comparez la qualité des résultats en représentant sur un graphique les spécimens 2D de l'ensemble d'entraînement, en utilisant des couleurs différentes pour distinguer les points de chaque classe.

3 Classification automatique de spécimens d'iris

Le deuxième but du travail dirigé est de mettre en œuvre la classification automatique de spécimens d'iris par un réseau de neurones. Nous étudierons en outre la sensibilité du classificateur à la réduction de la dimensionalité des données.

Pour ce faire, nous utiliserons le script `rn.m` distribué avec les données. Dans la configuration téléchargée, ce script réalise la classification des spécimens à quatre attributs de `iris.mat` : il admet donc quatre entrées. Comme toute sortie du réseau de neurones considéré ici est une valeur entre zéro et un, la classification entre trois classes nécessite la configuration de deux sorties et d'une convention d'encodage de ces sorties. Le script `rn.m` réalise la convention selon laquelle la sortie $(0, 0)$ correspond à la classe *setosa* (0), la sortie $(0, 1)$, à la classe *versicolor* et la sortie $(1, 0)$, à la classe *virginica*. Il reste la possibilité d'une sortie $(1, 1)$, qui correspondrait à un échec de classification (-1). Le script s'occupe de traduire la sortie de la classification par le réseau en index de classe 0, 1, 2 ou -1 et à produire un graphe comparant les classes réelles à celles déterminées par le réseau.

3.1 Travail à effectuer

Vous devrez modifier des copies du script de manière à classifier les spécimens dont la dimensionalité à été réduite selon chacune des méthodes considérées ci-haut, l'une après l'autre. Comparez et commentez brièvement les résultats de classification dans chacun des quatre cas : spécimens originaux (quatre attributs), spécimens projetés par minimum d'entropie, spécimens projetés par discriminant linéaire de Fisher et spécimens projetés sur les composantes principales.

4 Synthèse

Discutez brièvement de chacune des trois approches de réduction des attributs en vous basant notamment sur :

1. la dispersion intra- et inter-classe observée sur les représentations graphiques des données réduites ;
2. la qualité de classification des données réduites.

Ces éléments varient-ils de manière significative selon la méthode utilisée ?