Projet
Apprentissage non supervisé

à rendre au plus tard le 01/12/2017

1 Objectif

L'objectif de ce projet est d'étudier une méthode de clustering qui recherche des classes ordonnées dans le temps : la méthode de programmation dynamique de Fisher [1][2]. Elle est utilisée pour la segmentation de signaux temporels, ou pour la détection de changement dans une séquence de données. Dans les systèmes de reconnaissance de la parole par exemple, on exploite ce type de méthode pour partitionner des signaux audio en plages temporelles associées à des locuteurs différents qui sont ensuite identifiés. De manière plus générale, l'organisation d'une séquence de données en segments homogènes est un processus qui aide à mieux les organiser.

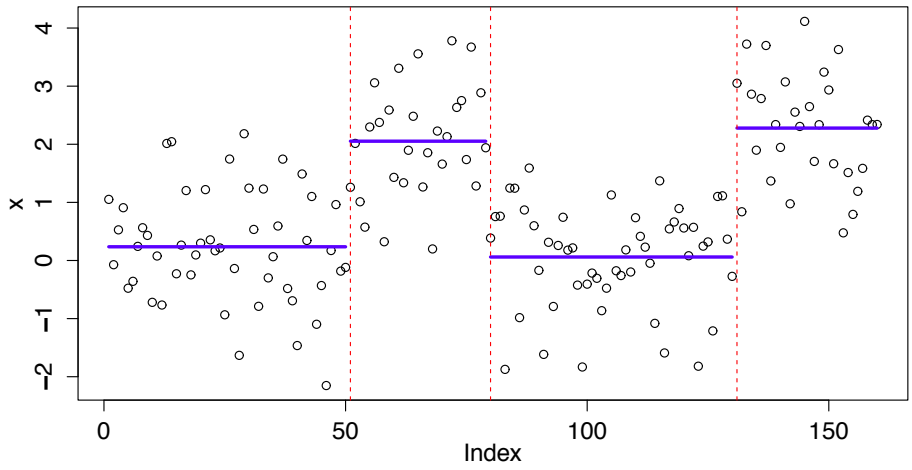

FIGURE 1 – Partitionnement de données temporelles en classes ordonnées dans le temps (segmentation)

2 Algorithme de programmation dynamique de Fisher

L'algorithme de programmation dynamique de Fisher cherche à partitionner une séquence de données $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ décrites par p variables en K classes qui apparaissent l'une après l'autre. Compte-tenu de la contrainte d'ordre, ce problème diffère de celui (classique) visant à trouver une partition minimisant par exemple l'inertie intra-classe. Dans cette nouvelle situation, l'objectif est de déterminer K intervalles temporels $[1; t_1[$, $[t_1; t_2[$, $\cdots$, $[t_{K-2}; t_{K-1}[$, $[t_{K-1}; n]$, ou, de manière équivalente, $K-1$ instants de changement $(t_1, \ldots, t_{K-1})$ minimisant le critère (de type inertie intra-classe) suivant :

$$\left\{ \begin{array}{lcll} C(t_1, \ldots, t_{K-1}) & = & D(1, t_1 - 1) + \left(\sum_{k=2}^{K-1} D(t_{k-1}, t_k - 1) \right) + D(t_{K-1}, n) & \text{si } K \geq 3 \\ C(t_1) & = & D(1, t_1 - 1) + D(t_1, n) & \text{si } K = 2 \end{array} \right. \tag{1}$$

avec

$$D(a, b) = \sum_{i=a}^{b} \| \boldsymbol{x}_i - \boldsymbol{\mu}_{ab} \|^2 \qquad \text{et} \qquad \boldsymbol{\mu}_{ab} = \frac{\sum_{i=a}^{b} \boldsymbol{x}_i}{b - a + 1}. \tag{2}$$

On notera ici que D est une matrice (triangulaire supérieure) de taille (n, n) appelée souvent **matrice des diamètres des classes**.

Contrairement au critère d'inertie intra-classe qui a été vu en cours (dans lequel aucune contrainte d'ordre n'est imposée sur les classes), le critère C peut être **minimisé de manière exacte (optimum global)** grâce à l'algorithme de programmation dynamique de Fisher qui est basé sur le principe selon lequel « toute sous-partition d'une partition optimale est nécessairement optimale ». L'algorithme est décrit ci-dessous.

Algorithm 1: Algorithme de programmation dynamique de Fisher

Entrées: séquence $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, nombre de segments souhaité ($K \geq 2$)

Étape 0

Initialiser les matrices D, M_1 et M_2, ainsi que le vecteur $\boldsymbol{t}$ des instants de changement et le vecteur des classes (sous R, on pourra utiliser des matrices et des vecteurs nuls)

Étape 1 : calcul de la matrice triangulaire supérieure des diamètres

pour $a = 1, \ldots, n,\ b = a, \ldots, n$ **faire**
  Calculer $D[a, b]$ selon la formule (2)
fin

Étape 2 : calcul récursif des critères optimaux

pour $i = 1, \ldots, n$ **faire**
  $M_1[i, 1] \leftarrow D[1, i]$
fin
pour $k = 2, \ldots, K$, $i = k, \ldots, n$ **faire**
  $M_1[i, k] \leftarrow \min\Big\{M_1[t-1, k-1] + D[t, i] \quad \text{avec} \quad t = k, \ldots, i\Big\}$
  $M_2[i, k] \leftarrow \operatorname{argmin}\Big\{M_1[t-1, k-1] + D[t, i] \quad \text{avec} \quad t = k, \ldots, i\Big\}$
fin

Étape 3 : calcul récursif des instants de changement optimaux

$k \leftarrow K - 1$
$m \leftarrow n$
tant que $k \geq 1$ **faire**
  $t_k \leftarrow M_2(m, k+1)$
  $m \leftarrow t_k - 1$
  $k \leftarrow k - 1$
fin

Étape 5 : labels des classes formés à partir des instants de changement

pour $i = 1, \ldots, t_1 - 1$ **faire**
  $cluster[i] \leftarrow 1$
fin
pour $k = 2, \ldots, K-1$, $i = t_{k-1}, \ldots, t_k - 1$ **faire**
  $cluster[i] \leftarrow k$
fin
pour $i = t_{K-1}, \ldots, n$ **faire**
  $cluster[i] \leftarrow K$
fin

Sorties: labels *cluster* et instants de changement $\boldsymbol{t} = (t_1, \ldots, t_{K-1})$

3 Travail à effectuer

1) Implémenter sous R l'algorithme de programmation dynamique de Fisher. Créer pour cela deux fonctions : une fonction `diam` calculant le diamètre $D(a, b)$ suivant la formule (2) et une fonction `clustfisher` pour l'algorithme lui-même. Pour le calcul de l'argument minimum (argmin) d'un vecteur numérique, vous pourrez utiliser la fonction `which.min` de R.

2) Application à des données simulées : en utilisant l'algorithme de Fisher, segmenter la séquence de données simulées `sequencesimu` (à télécharger) et régler le nombre de classes (segments) à l'aide de la méthode du coude. Comparer cette partition avec celles de l'algorithme des kmeans et de la méthode CAH-Ward.

3) Application à des données réelles : segmenter en 4 classes la séquence $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{140})$ des données `aiguillage` (on rappelle que dans ce cas, chaque observation $\boldsymbol{x}_i$ appartient à $\mathbb{R}^{552}$. Comparer la partition obtenue par l'algorithme de Fisher avec la vraie partition ainsi qu'avec les partitions issues de l'algorithme des kmeans et de la méthode CAH-Ward.

4 Documents à rendre

Un compte-rendu n'excédant pas 7 pages (.doc ou .pdf) et le code R associé.

Références

[1] Fisher W. D. (1958). On grouping for maximum homogeneity. Journal of the American Statistical Association, 53(284) :789–798.
[2] Lechevallier Y. (1990). Recherche d'une partition optimale sous contrainte d'ordre total. Research Report RR1247, INRIA. Projet CLOREC.