INF 413 – TP5-6

Alan Gardin – Ronan Garet

13/03/2017

# Notations utilisées

# Fonctionnement de l’algorithme

L’algorithme prend en entrée une liste de points à coordonnées et , le nombre de barycentres.

L’algorithme renvoie en sortie ou est une liste contenant l’ensemble des points du cluster associé au centre .

Début de l’algorithme :

Choisir centres distincts de parmi

Calculer l’ « éloignement » entre chaque couple de points, où correspond à l’éloignement entre les points et .

Tant que condition d’arrêt non-respectée :

Pour i allant de 0 à n-1 :

On cherche j tel que Lij minimal et pj ∈ [c0,…,ck-1]

Ajouter pi à C

Fin Pour

Pour i allant de 0 à k-1 :

Calculer Bi, le barycentre de Ci

Remplacer ci par Bi

Fin Pour

Fin Tant que

Retourner C

# Calcul de l’ « éloignement » :

Dans un 1er temps, nous avons choisi le calcul de distance qui nous a paru le plus naturel : la distance euclidienne. Soit la distance entre les points et , on a donc

Pour simplifier les calculs, on ne calculera pas la racine carrée et on appellera cette valeur « éloignement ». On a donc .

# Quantifier la qualité d’une solution

Pour un nombre de centres donnés, une solution est d’autant plus optimale que la distance moyenne entre chaque point et son centre est faible. C’est cette valeur que nous utiliserons pour quantifier la qualité d’une solution par rapport à une autre.

# Choix du nombre de sommets

Dans un premier temps, le nombre de sommets sera choisi de manière arbitraire.

Mais le choix du nombre de clusters est un point crucial. En effet, si le nombre de cluster est insuffisant par rapport à la distribution des données, l’algorithme renvoie des résultats « erronés ».

Pour pallier ce problème, nous avons implémenté l’algorithme G-means. Celui çi permet de « trouver » le bon nombre de cluster à adopter. Cet algorithme repose sur le pseudo-code suivant :

[code]

En pratique cependant, la reconnaissance d’une répartition gaussienne s’est avérée assez aléatoire. En conséquence, très souvent l’algorithme ne converge pas : étant donné une distribution gaussienne non reconnue, l’algorithme va couper cette distribution en deux et appliquer à nouveau des tests de répartition gaussienne.

[Schéma]

Il est facile de voir que la coupe d’une distribution gaussienne a très peu de chance d’être détectée comme gaussienne ensuite. La suite de découpe qui s’ensuit empêche la terminaison de l’algorithme.

# Calcul du barycentre

On prend comme barycentre d’un ensemble de n points la moyenne de leurs coordonnées.

# Condition d’arrêt

L’algorithme des k-means converge vers une solution localement optimale. L’idéal serait donc d’arrêter le programme quand les solutions trouvées ne varient plus.

Cependant, il est possible que cela n’arrive jamais (présence de cycles).

Dans un premier temps, on choisira comme condition d’arrêt un nombre d’itérations choisi de façon arbitraire.

Une manière plus efficace que nous implanterions dans une version ultérieure serait de garder en mémoire la meilleure solution trouvée (distance moyenne entre les points et leur centre minimale) ; si cette solution n’a pas changé après un certain nombre d’itérations (choisi de façon arbitraire ?) on arrête le programme.

# Choix initial des centres

Dans un premier temps, le choix initial des centres se fera de manière aléatoire. Aucun point n’étant plus important qu’un autre on choisira une probabilité uniforme.

Cependant, l’algorithme des k-means ne fournissant qu’une solution localement optimale, les choix initiaux des centres s’avère crucial et devra être amélioré. En effet, les choix initiaux dépendent fortement de l’utilisation qui sera faite de l’algorithme.

# Complexité de l’algorithme

* Le choix des centres initiaux est en .
* Le calcul de la distance est en  : pour chaque point, on calcule sa distance avec chaque barycentre, et le calcul de la distance entre points est en .
* Le calcul des barycentres est en  : Pour chaque cluster, on fait une addition des coordonnées de ses points qu’on divise par le nombre de points du cluster. Au final, on a un nombre de calculs en

Si on choisit comme condition d’arrêt un nombre donné d’itérations, on a une complexité de l’algorithme en .

Si on se donne comme condition d’arrêt la convergence de la solution

# Utilisations possibles de ce genre d’algorithmes

* Identification de groupes d’individus dans une foule (une dimension par caractéristique étudiée : âge, revenus, …).
* Identification de regroupements d’étoiles afin de déterminer des galaxies