Notes sur les k moyennes

Maximilien

June 2020

Notions MIASHS fonctions, normes, espaces vectoriels

1 Introduction

Le problème du clustering suppose que nous disposons d'un ensemble de données non labellisées mais pour lesquelles nous avons une forte suspicions qu'il existe une variable latente qui permettrait de les regrouper de manière logique. Dit autre, il s'agit de trouver une partition de nos données qui satisferaient certains critères. L'algorithme des k moyenne, en particulier, cherche à créer une partition de k *clusters* (ou parties) où chacun est décrit par un vecteur "central". Une donnée appartient au *cluster* dont le centre lui est le plus proche. De plus, ce centre est calculé comme la moyenne des donnée que le *cluster* regroupe.

L'algorithme des k moyennes offre un certain nombre d'avantages. Il est intuitif tout en satisfaisant certaines propriétés intéressantes.

En raison de cet aspect intuitif, ces notes commencerons par présenter l'algorithme avant de formaliser le problème plus rigoureusement. On notera dans la suite $\mathcal{D} = \{\boldsymbol{x_i}\}_{i \leq n}$ un jeu de données sur $\mathbb{R}^n$. La figure 1 représente le résultat d'une exécution des k moyennes. Chaque moyenne est le barycentre du *cluster* qu'elle décrit.

2 L'algorithme

L'algorithme des k moyennes consiste à répéter deux opérations jusqu'à convergence.

L'ensemble des points de notre jeu de données sont assignés à la moyenne la plus proche lors de la première étape. Puis, les moyennes sont mise à jour en moyennant les qui lui ont été assignés.

L'algorithme 1 décrit ce fonctionnement de manière plus rigoureuse.

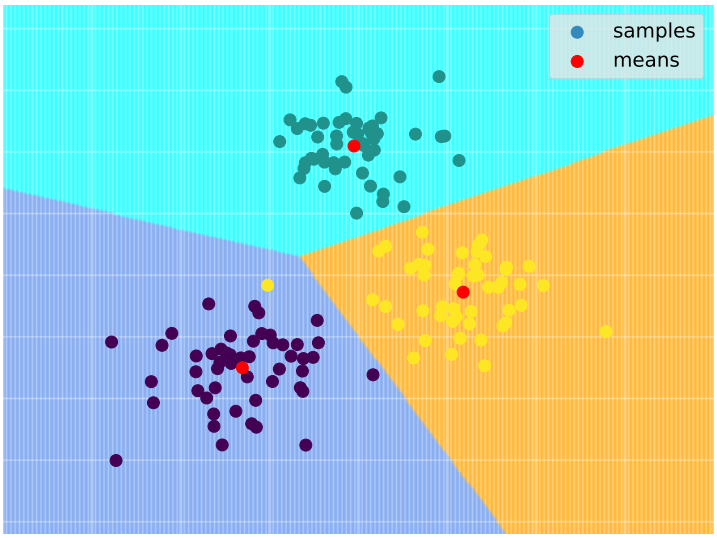

FIGURE 1 – Une exécution des $k = 3$ moyennes. Chacune d'entre elles est mise en avant en rouge. La couleurs de chaque point de notre jeu de données indique la "vraie" classe inconnue à laquelle il était initialement assigné.

Algorithm 1: k moyennes

input : $\mathcal{D}$ (non trié), k.
output: k moyennes.
means = `Array`[k];
assignment = `Array`[$|\mathcal{D}|$];
// Une stratégie d'initialisation est d'assigner;
// les k moyennes aux k premiers points de notre;
// jeu de données;
for $i = 1$ **to** k **do**
  means[i]=$\mathcal{D}[i]$;
end
while *not stop_condition* **do**
  // Première étape, assignation;
  // des points de notre jeu de données;
  // à la moyenne la plus proche;
  for $i = 1$ **to** $|\mathcal{D}|$ **do**
    assignment[i] = `closest_mean`($\mathcal{D}[i]$, means);
  end
  // Seconde étape, mise à jour des moyennes;
  for $i = 1$ **to** k **do**
    means[i] = `average`($\mathcal{D}$[assignment == i]);
  end
end
output=means;

On note cependant une étape d'initialisation. En effet, la valeur des moyennes au départ peut affecter la convergence de l'algorithme. Une stratégie qui s'est montrée efficace est d'assigner chaque "moyenne" à un point du jeu de données.

3 Formalisme

Soit un espace vectoriel $\mathbb{R}^d$ et $\|\cdot\|_2$ la norme ℓ_2. L'espace vectoriel $\{\mathbb{R}^d, \|\cdot\|_2\}$ est donc normé. Soit un ensemble de vecteurs qu'on appellera points $\mathcal{D} = \{\boldsymbol{x_i}\}_{i \leq n}$. Le problème adressé par l'algorithme des k moyennes est celui de la recherche d'une "bonne" partition des données. Ainsi, soit $k \in \mathbb{N}^+$ le cardinal de notre partition et notons z_i l'indice de la partie à laquelle l'élément $\boldsymbol{x_i}$ est rattaché. Le barycentre d'une partie d'indice j est défini de la manière suivante :

$$\boldsymbol{\mu_j} = \frac{1}{\sum_i \mathbb{1}[z_i = j]} \sum_i \boldsymbol{x_i} \mathbb{1}[z_i = j]$$

Où on supposera que chaque partie contient au moins un élément. La fonction $\mathbb{1}[z_i = j]$ renvoie 1 si l'égalité est satisfaite, 0 sinon. En nous appuyant sur le barycentre, nous pouvons évaluer la qualité d'une partition :

$$\phi(z_1, ..., z_n, x_1, ..., x_n) = \sum_{i=1}^{n} \|\boldsymbol{x_i} - \boldsymbol{\mu_{z_i}}\|_2^2$$

Enfin, nous pouvons définir le problème qu'essaye de résoudre l'algorithme des k moyennes de la manière suivante :

$$P^\star = \underset{z_1,...,z_n \in \mathcal{P}_k(\mathcal{D})}{\operatorname{argmin}} \phi(z_1, ..., z_n, x_1, ..., x_n) \tag{1}$$

où $\mathcal{P}_k$ est l'ensemble des k-partitions (i.e. partitions de cardinal k). La notation $z_1, ..., z_n \in \mathcal{P}_k$ est un abus de langage et décrit une partition du point de vue de notre jeu de données $\mathcal{D}$. La résolution de ce problème est connue pour être NP complète à partir de $k = 2$. L'algorithme 1 n'est ainsi qu'une recherche locale d'une solution et dépend de l'initialisation. Il est possible de montrer que l'algorithme fait décroître ϕ de manière monotone à chaque itération. Il suffit de constater que chacune des deux étapes de l'algorithme minimise ϕ : si les points sont déplacés vers le *cluster* le plus proche, alors elles sont plus proches de leur *cluster* et ϕ est réduit, puis si le centre des *clusters* est recalculé par rapport à la moyenne, alors la distance moyenne ϕ est réduite également.

Le nombre total de combinaison étant fini, on peut garantir que l'algorithme se terminera nécessairement. Cependant, le nombre d'itérations effectives est en réalité très faible.

Notons que l'algorithme des k moyennes se rapproche de l'algorithme EM appliqué au cas du mélange gaussien où la matrice de covariance de chaque Gaussienne est la matrice identité.

4 Qualité d'un *clustering*

Définir ce qu'est un bon regroupement est une question fondamentalement complexe. C'est celle qui nous amène a décider ce qu'est une espère ou ce qui revêt de la variabilité intra-espèce.

Cependant, nous pouvons définir des "mesures" de qualité permettant de comparer plusieurs types de regroupement. Intuitivement, on aimerait que les éléments qui appartiennent à un même groupe se ressemblent le plus possibles et que des éléments appartenant à des groupes différents soient plus distants. Ces idées sont particulièrement intéressantes par exemple pour choisir le k des k moyennes.

Pour cela, introduisons la mesure de distance intra-cluster ou *homogénéité* :

$$H_j(z_1, ..., z_n, x_1, ..., x_n) = \frac{\sum_i \|\boldsymbol{\mu_j} - \boldsymbol{x_i}\|_2 \, \mathbb{1}[z_i = j]}{\sum_i \mathbb{1}[z_i = j]},$$

ainsi que la notion de distance inter-cluster ou *séparation* :

$$S_{kl}(z_1, ..., z_n, x_1, ..., x_n) = \|\boldsymbol{\mu_k} - \boldsymbol{\mu_l}\|_2 .$$

On souhaite maximiser la première et minimiser la seconde. De manière un peu plus formalisé, on peut mesurer cet effet par le ratio suivant :

$$D_k(z_1, ..., z_n, x_1, ..., x_n) = \max_{j \neq k} \frac{H_k(\ldots) + H_j(\ldots)}{S_{kj}(\ldots)}$$

Ce qui nous mène à la mesure globale :

$$D(\ldots) = \frac{1}{k} \sum_{l=1}^{k} D_l(\ldots)$$