Remise à niveau en langage C et en IA : Clustering

M2 SIIC
Pierre Andry
Université de Cergy Pontoise
pierre.andry@ensea.fr

K-moyennes

L'algorithme des K-moyennes est un algorithme de classification automatique. A partir d'un nombre de partitions initiales, on cherche à améliorer itérativement le partitionnement. Chaque partition est représentée par un noyau, il y a donc k noyaux. Au cours des itérations, on cherche à minimiser le critère de distance des K noyaux aux clusters que l'on constitue. Voici l'algorithme :

1. Déterminer le nombre k, de centres, c'est à dire le nombre de partitions que l'on souhaite effectuer.
2. Choisir aléatoirement K noyaux parmis les échantllons.
3. tant que la convergence n'est pas atteinte (tant que des échantillons changent de groupes):
   (a) Déterminer, pour chaque element de l'ensemble à catégoriser, le centre le plus proche.
   (b) Redéfinir, pour chaque classe le nouveau noyau (l'échantillon au centre de la classe).

Application dans un espace géométrique

Nous allons appliquer l'algorithme des k-moyennes à un ensemble de points définis dans un espace géométrique en 2 dimensions. Nous programmerons l'algorithme en C avec une sortie des calculs sur fichier texte pour un affichage avec gnuplot.

1. Vous définirez dans le .h (par des directives #DEFINE) le nombre d'elements de l'échantillon (par exemple 200 et le nombre K de noyaux (par exemple K). Vous definirer aussi une directive FILE_ON qui permet d'afficher les resultats dans un fichier, ou à l'ecran si elle n'est pas definie (vous utiliserez fprintf sur stdout dans ce cas).
2. Vous écrirez la structure Data qui contiendra les coordonnées x et y d'un élément de l'échantillon, ainsi que d'un champ permettant d'indiquer quel est le noyau de cet échantillon (soit un entier, soit un pointeur sur l'élément noyau de la classe). L'échantillon sera un tableau de 200 Datas
3. creez un tableau de k pointeurs sur Datas (chaque pointeur pointera sur l'élément Data sélectionné)
4. Ecrivez la fonction qui initialise les 200 points par des valeurs aléatoires à des coordonnées comprises entre 0 et 100.
5. Ecrivez la fonction qui initialise (par tirage aléaoire) les K noyau (et affecte les k pointeurs)
6. Déterminez, pour chaque element de l'ensemble à catégoriser, le centre le plus proche (distance euclidienne). Cette fonction sera itérée jusqu'à convergence.
7. Redéfinissez, pour chaque classe le nouveau noyau (l'échantillon au centre de la classe) , c'est a dire l'élément Data le plus proche de la moyenne des coordonnées du tableau. Cette fonction sera itérée jusqu'à convergence.
8. A chaque itération, sauvegardez dans k fichiers différents les coordonnées des points qui appartiennent a chaque classe. Affichez vos résultats avec gnuplot.