Premier rapport

November 25, 2012

1 Définition du problème de partionnement

Soit $G_0 = (\mathcal{V}_0, \mathcal{E}_0, A_0)$ un graphe, où $\mathcal{V}_0$, $\mathcal{E}_0$ et A_0 sont respectivement les ensembles de sommets, d'arêtes et enfin des poids associés à chaque arête. Le partionnement du graphe G_0 est un procédé qui vise à partitionner l'ensemble$\mathcal{V}_0$ de ses points en K parties distinctes $(\pi_k)_{1\leq k\leq K}$, de sorte à optimiser une fonction objectif tout en respectant des contraintes données.

Une partition satisfait :

$$\begin{aligned} \pi_k &\subseteq \mathcal{V}_k \\ \bigcup_{k=1}^{K} \pi_k &= \mathcal{V}_k \\ \pi_k &\cap \pi_{k'} = \emptyset \qquad \forall k \neq k^{'} \end{aligned}$$

Pour chaque sous ensemble π_k, nous notons $\mu_k$son centre :

$$\mu_k = \frac{1}{|\pi_k|} \sum_{i \epsilon \pi_k} X_i$$

Le problème des k-moyennes consiste à determiner une partition minimisant la somme des carrées des distances entre chaque point et le centre de sous-ensemble auquel il est affecté. C'est a à dire :

$$min \sum_{i=1}^{n} \parallel X_i - \mu_i \parallel^2$$

Dans notre cas, on travaille avec des objectifs séparables, donc on peut réécrire la fonction objectif comme :

$$\underset{\pi}{arg\,min} \quad \sum_{i=1}^{n} \sum_{X_j \epsilon \pi_i} \quad \parallel X_j - \mu_i \parallel^2$$

2 L'algorithme multi-niveaux

2.1 Réduction du graphe

La première étape de l'algorithme consiste à réduire la taille du graphe de départ en fusionnant des sommets en "supersommets" via différentes méthodes de combinaison.

On utilise ici l'algorithme suivant pour construire G_i à partir de G_{i+1} :

Algorithm 1: Initialisation avant la réduction

pour chaque $x \in G_i$ **faire**
  $marque\,(x) = faux$
fin

Algorithm 2: Phase de réduction

Appeler $algorithm1()$
pour chaque $x \in G_i$ **faire**
  si $\forall y \in voisins\,(x)\,, marque\,(y) = vrai$ **alors**

$$\begin{aligned} marque(x) &= vrai \\ G_i &= G_i \setminus x \end{aligned}$$

  sinon
    trouver y qui maximise $\frac{e(x,y)}{w(x)} + \frac{e(x,y)}{w(y)}$, avec $w(i)$ le poids associé au sommet i
    fusionner x et y

$$\begin{aligned} marque(x) &= vrai \\ marque(y) &= vrai \\ G_i &= G_i \setminus x \\ G_i &= G_i \setminus y \end{aligned}$$

  fin
fin

2.2 Phase de partitionnement initial

On arrête la réduction du graphe lorsque celui-ci compte moins que $20k$ sommets, où k est le nombre de parties désiré. On peut alors obtenir un partitionnement initial de G_0 en partitionnant G_m, le plus petit graphe obtenu après la réduction.

Pour ce faire on utilise l'algorithme spectral de Yu et Shi généralisé avec des pondérations arbitraires.

2.3 Phase de raffinement

Ce premier partitionnement ayant été obtenu, on peut le raffiner en parcourant les graphes G_{m-1} ,..., G_0 de la manière suivante:si un "supersommet" de G_i est dans une partition c, alors tous les sommets de G_{i-1}qui le composent sont également dans c, ce qui mène à un partitionnement initial pour le graphe G_{i-1}.

Il faut ensuite raffiner ce partitionnement pour G_{i-1}. Pour ce faire, on définit un partitionnement initial $\pi_c^{(0)}$ une matrice noyau K selon la fonction objectif du partitionnement et on utilise l'algorithme suivant :

Algorithm 3: Initialisation avant le raffinement

pour chaque *chaque ligne i de K* **faire**

$$d(i, \boldsymbol{m}_c) = K_{ii} - \frac{2\sum_{j\in\pi_c(t)} \cdot poids(j) \cdot K_{ij}}{\sum_{j\in\pi_c(t)} \cdot poids(j)} + \frac{\sum_{j,l\in\pi_c(t)} \cdot poids(j) \cdot poids(l) \cdot K_{jl}}{(\sum_{j\in\pi_c(t)} \cdot poids(j))^2}$$

fin

Algorithm 4: Phase de raffinement

Appeler $algorithm3()$

Trouver $c^*(i) = argmin_c d(i, \boldsymbol{m}_c)$ (on s'occupe des égalités de façon arbitraire)

tant que $((\neg$ *convergence*$)$ *et* $($*seuil d'itérations pas dépassé*$))$ **faire**

Mettre à jour les partitions :

$$\begin{aligned} \pi_c^{(t+1)} &= \{i : c^*(i) = c\} \\ t &= t+1 \end{aligned}$$

fin

2.4 Algorithme de principe

Notre algorithme général peut se décomposer en trois phases:

- Le première phase "d'agrégagtion" qui va utiliser le premier algorithme précédent (1 & 2) afin de reduire et d'obtenir un graphe avec K "supersommets".
- Une phase de "partitionnement" qui va nous façonner notre premiere et grossière partition décrite au point 2.2 .
- Puis vient la phase de "raffinement" qui, par le biais de l'algorithme (3 & 4), permet de raffiner la grossière partition afin de tirer le meilleur regroupement possible.

Algorithm 5: Algorithme général de partitionnement$(G_0 = (\mathcal{V}_0, \mathcal{E}_0, A_0), \{\pi_c\}_{c=1}^{k}, t_{max}))$

Input: k : nombre de partition que l'on désire , G_0 : Graphe initial , t_{max}: nombre maximum d'itération
Output: $\{\pi_c\}_{c=1}^{k}$: partitionnement final de notre graphe
$compteur = 0$
tant que $|\mathcal{V}_{compteur}| < 20k$ **faire**

$$\begin{aligned} G_{compteur+1} &= algorithm2(G_{compteur}) \\ compteur &= compteur + 1 \end{aligned}$$

fin
$\{\pi_c\}_{c=1}^{k}$ =Partitionnement primaire par l'algorithme de Yu and Shi.
pour *i de compteurjusqu'à 1 avec pas* $= -1$ **faire**

$$\begin{array}{ccc} Appliquer & \{\pi_c\}_{c=1}^{k} & \text{à } G_{compteur-1} \\ \{\pi_c\}_{c=1}^{k} & = & algorithm4(K, k, \omega, t_{max}, \{\pi_c\}_{c=1}^{k}) \end{array}$$

fin
