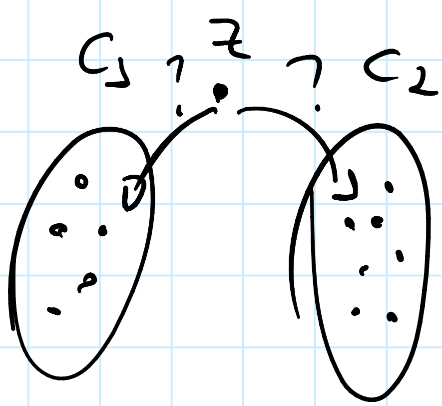
Algorithm - Cost of Moving

Saturday, May 27, 2017 10:14 AM

$$\max_{\{C_j\}} \sum_{j=1}^{k} \underbrace{\frac{1}{n_j} \sum_{x,y \in C_j} k(x,y)}_{\frac{1}{n_j} Q_j}$$



Suppose $z \in C_1$:

$$Q = \frac{1}{n_1} \sum_{\substack{x,y \in C_1 \\ \ni z}} k(x,y) + \frac{1}{n_2} \sum_{\substack{x,y \in C_2 \\ x,y \neq z}} k(x,y)$$

Move z to C_2:

$$Q' = \frac{1}{n_1 - 1} \sum_{\substack{x,y \in C_1 \\ x,y \neq z}} k(x,y) + \frac{1}{n_2 + 1} \sum_{\substack{x,y \in C_2 \\ \ni z}} k(x,y)$$

$$= \frac{1}{n_1 - 1} \sum_{x,y \in C_1 \ni z} k(x,y) - \frac{1}{n_1 - 1} \Big(2 \sum_{y \in C_1 \ni z} k(z,y) \Big)$$

$$+ \frac{1}{n_2 + 1} \Big(\sum_{\substack{x,y \in C_2 \\ x,y \neq z}} k(x,y) + 2k(z,z) + 2 \sum_{\substack{y \in C_2 \\ y \neq z}} k(z,y) \Big)$$

$\rightarrow$ original costs

$$n_2+1 \Big(\sum_{\substack{x,y\in C_2\\ x,y\neq z}} \qquad \sum_{\substack{y\in C_2\\ y\neq z}} \Big)$$

original costs

$$Q' = \frac{1}{n_1-1}\sum_{x,y\in C_1} k(x,y) + \frac{1}{n_2+1}\sum_{x,y\in C_2} k(x,y)$$
$$- \frac{1}{n_1-1}\, 2\sum_{y\in C_1} k(z,y) + \frac{2}{n_2+1}\Big(k(z,z) + \sum_{y\in C_2} k(z,y) \Big)$$

$$Q'-Q = \left(\frac{1}{n_1-1} - \frac{1}{n_1}\right)\sum_{x,y\in C_1} K(x,y) + \left(\frac{1}{n_2+1} - \frac{1}{n_2}\right)\sum_{x,y\in C_2} K(x,y)$$
$$- \frac{2}{n_1-1}\sum_{y\in C_1} k(z,y) + \frac{2}{n_2+1}\sum_{y\in C_2} k(z,y)$$
$$+ \frac{2}{n_2+1} k(z,z)$$

$$= \frac{1}{n_1(n_1-1)}\sum_{x,y\in C_1} k(x,y) - \frac{1}{n_2(n_2+1)}\sum_{x,y\in C_2} k(x,y)$$
$$+ \frac{2}{n_2+1} K(z,z)$$
$$+ \frac{2}{n_2+1}\sum_{y\in C_2} k(z,y) - \frac{2}{n_1-1}\sum_{y\in C_1} k(z,y)$$

$$Q'-Q = \frac{1}{n_1(n_1-1)} Q_1 - \frac{1}{n_2(n_2+1)} Q_2$$
$$+ \frac{2}{n_2+1}\Big(\sum_{y\in C_2} k(z,y) + k(z,z) \Big)$$
$$- \frac{2}{n_1-1}\Big(\sum_{y\in C_1} k(z,y) \Big)$$

$$\frac{2}{n_1-1}\left(\sum_{y\in C_1} k(z,y)\right)$$

$$= \frac{1}{n_1(n_1-1)} Q_1 - \frac{1}{n_2(n_2+1)} Q_2 + \frac{2}{n_2+1} Q_2^{+}(z) - \frac{2}{n_1-1} Q_1(z)$$

$$= \frac{1}{n_1(n_1-1)} Q_1 - \frac{2}{n_1-1} Q_1(z) \rightarrow A_1$$

$$- \left(\frac{1}{n_2(n_2+1)} Q_2 - \frac{2}{n_2+1} Q_2^{+}(z)\right) \rightarrow B_2$$

move if $A_1 > B_2$

$$Z = \overbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ 0 & 1 & 0 & \cdots \\ & & & \end{pmatrix}}^{n}_{n\times k} \quad \text{label matrix}$$

$$K = \begin{pmatrix} k_{11} & k_{12} & \cdots \\ k_{21} & k_{22} & \cdots \\ k_{31} & k_{32} & \cdots \end{pmatrix}_{m\times m} \quad \text{Kernel matrix}$$

$$Q_1(z) = \sum_{y\in C_1} k(z,y) \quad \text{assuming } z \rightarrow \text{index } i$$

$$= \sum_{l=1}^{m} K_{il} Z_{l1} = (K Z_{\cdot 1})_i$$

If we remove z from Q_1 the new cost is

$$Q_1' = \sum_{x,y \in C_1} k(x,y) - 2 \sum_{y \in C_1} k(z,y)$$

$$= Q_1 - 2\,Q_1(z)$$

If we add z to Q_2 the new cost is

$$Q_2' = \sum_{x,y \in C_2} k(x,y) + 2 \sum_{y \in C_2} k(z,y)$$

$$= Q_2 + 2\,Q_2^+(z)$$

Algorithm

1) Start with Z_0. Compute

$$Q = [Q_1, \ldots, Q_k]$$
$$m = [m_1, \ldots, m_k]$$

2) For each point z:

   Find j such that $z \in C_j$.

   Compute A_j

   $j^* = \operatorname{argmin}_{l \neq j} B_l$

   if $A_j - B_{j^*} > 0$:

   move z to C_{j^*}

if $A_j - B_{j^*} > 0$:
move z to C_{j^*}
update Q, m.

3) Repeat until convergence.

<u>Practical Aspects</u>:

To compute $Q_j(z)$ pick the jth column of Z: $Z_{\bullet j} = Z[:, j]$. Let z have index i. Then $Q_j(z) = K_{i\bullet} \cdot Z_{\bullet j}$

To compute $Q_\ell^+(z)$ pick the ℓth column of Z and add z to the partition:
$Z_{\bullet \ell} = Z[:, \ell]$, $(Z_{\bullet \ell})_i = 1$. Now
$Q_\ell^+(z) = K_{i\bullet} \cdot Z_{\bullet \ell}$.

We store $Q_\ell^+(z)$ in a list. Then when moving the point to a new partition:

$Z_{ij} = 0$; $Z_{ij^*} = 1$
$m_j \leftarrow m_j - 1$; $m_{j^*} \leftarrow m_{j^*} + 1$
$Q_j \leftarrow Q_j - 2Q_j(z)$; $Q_{j^*} \leftarrow Q_{j^*} + 2Q_{j^*}^+(z)$

—— n ——

Algorithm

Algorithm

1) Compute kernel Matrix G
2) Initialize $z = z_0$
3) For each point z:
   3.1) j such that $z \in C_j$
   3.2) Q_j, $Q_j(z)$
   3.2) $A_j = \frac{1}{n_j(n_j-1)} Q_j - \frac{2}{n_j - 1} Q_j(z)$
   3.3) For each partition $l \neq j$
   $A_l = \frac{1}{n_l(n_l+1)} Q_l - \frac{2}{n_l+1}\left(Q_l(z) + G(z,z)\right)$
   3.4) $j^* = \arg\max_l (A_l - A_j)$
4) If $A_j - A_{j^*} > 0$:
   move z to C_{j^*}
   else:
   keep z in C_j
5) Repeat until convergence.