Computation of centroids for fair k-means

These slides describe the algorithm (*CentroidSelection*) to be used in the variant of Lloyd's algorithm for fair k-means clustering, proposed by

M. Ghadiri, S. Samadi, S.S. Vempala. Socially Fair k-Means Clustering. Proc. of ACM FAccT 2021: p.438-448

For details, refer to Section 2 of the paper, and, in particular, to Algorithm 2 (called "Line Search") on page 441 of the paper.

Let $U = A \cup B$ be a set of points in $\mathbb{R}^d$, where the subsets A and B represent two demographic groups, and suppose that U is partitioned into k clusters $U_1, U_2, \ldots, U_k$.

Given one such partition, we compute a set $\{c_1, c_2, \ldots, c_k\}$ of k centroids which minimize the following objective function:

$$\Phi(A, B, C) = \max \left\{ \frac{1}{|A|} \sum_{i=1}^{k} \sum_{a \in A \cap U_i} \|a - c_i\|^2, \frac{1}{|B|} \sum_{i=1}^{k} \sum_{b \in B \cap U_i} \|b - c_i\|^2, \right\}$$

For $1 \leq i \leq k$ define:

$$\alpha_i = \frac{|A \cap U_i|}{|A|} \qquad \beta_i = \frac{|B \cap U_i|}{|B|}$$

$$\mu_i^A = \frac{1}{|A \cap U_i|} \sum_{a \in A \cap U_i} a \qquad \mu_i^B = \frac{1}{|B \cap U_i|} \sum_{b \in B \cap U_i} b$$

$$\ell_i = \|\mu_i^A - \mu_i^B\| \qquad (\textit{Euclidean norm}).$$

Observation: μ_i^A and μ_i^B are the standard centroids of $A \cap U_i$ and $B \cap U_i$, respectively, and ℓ_i the Euclidean distance between them. It can be shown that the best centroid c_i for U_i is along the segment connecting μ_i^A and μ_i^B, of length ℓ_i.

NOTATIONS:

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k) \qquad \beta = (\beta_1, \beta_2, \ldots, \beta_k)$$

$$M^A = (\mu_1^A, \mu_2^A, \ldots, \mu_k^A) \qquad M^B = (\mu_1^B, \mu_2^B, \ldots, \mu_k^B)$$

$$\ell = (\ell_1, \ell_2, \ldots, \ell_k)$$

Define the two quantities:

$$\begin{aligned}\Delta(A, M^A) &= \sum_{i=1}^{k} \sum_{a \in A \cap U_i} \|a - \mu_i^A\|^2 \\ \Delta(B, M^B) &= \sum_{i=1}^{k} \sum_{b \in B \cap U_i} \|b - \mu_i^B\|^2\end{aligned}$$

Finally, for a vector $x = (x_1, x_2, \ldots, x_k)$ of k real numbers, define the functions

$$\begin{aligned}f_A(x) &= \frac{\Delta(A, M^A)}{|A|} + \sum_{i=1}^{k} \alpha_i x_i^2 \\ f_B(x) &= \frac{\Delta(B, M^B)}{|B|} + \sum_{i=1}^{k} \beta_i (\ell_i - x_i)^2.\end{aligned}$$

Functions $f_A(x)$ and $f_B(x)$ represent the contributions of A and B to $\Phi(A, B, C)$ when each centroid c_i is chosen at distance x_i from μ_i^A, along the segment connecting μ_i^A and μ_i^B.

The algorithm described below computes a vector x which reduces the discrepancy between $f_A(x)$ and $f_B(x)$ as much as possible, and use this vector to compute the centroids c_i's.

Algorithm CentroidSelection

$\texttt{Compute vectors}$ $\alpha, \beta, M^A, M^B, \ell$; // *as defined above*

$\texttt{fixed}_A \leftarrow \Delta(A, M^A)/|A|$;

$\texttt{fixed}_B \leftarrow \Delta(B, M^B)/|B|$;

$(x_1, x_2, \ldots, x_k) \leftarrow \texttt{computeVectorX}(\texttt{fixed}_A, \texttt{fixed}_B, \alpha, \beta, \ell, k)$;

for $1 \leq i \leq k$ **do**

  $c_i \leftarrow \frac{(\ell_i - x_i)\mu_i^A + x_i \mu_i^B}{\ell_i}$

return $(c_1, c_2, \ldots, c_k)$

Function `computeVectorX`, whose code will be provided to you both in Java and Python, is described in the next slide

Let T be an integer parameter.

Function computeVectorX($\text{fixed}_A, \text{fixed}_B, \alpha, \beta, \ell, k$)

$\gamma \leftarrow 0.5$;

for $1 \leq t \leq T$ **do**

    $f_A(x) \leftarrow \texttt{fixed}_A$;

    $f_B(x) \leftarrow \texttt{fixed}_B$;

    for $1 \leq i \leq k$ **do**

        $x_i \leftarrow \frac{(1-\gamma)\beta_i \ell_i}{\gamma \alpha_i + (1-\gamma)\beta_i}$

        $f_A(x) \leftarrow f_A(x) + \alpha_i (x_i)^2$;

        $f_B(x) \leftarrow f_B(x) + \beta_i (\ell_i - x_i)^2$;

    if $f_A(x) = f_B(x)$ **then exit the for-loop;**

    else

        if $f_A(x) > f_B(x)$ **then** $\gamma \leftarrow \gamma + (1/2)^{t+1}$;

        else $\gamma \leftarrow \gamma - (1/2)^{t+1}$;