ALGORITHMS IN ACTION - Clustering

Based on Lectures by URI ZWICK and HAIM KAPLAN

June 13, 2016

1 k-centers

Given a set of n points A of some metric space X, find a set C of k points in X, such that we minimize $\max_{x \in A} d(x, C)$.

One can think of it as covering A with k cycles of the same radius while trying to minimize that radius.
we will use an approximation algorithm:

Algorithm 1 k centers approximation

pick an arbitrary point $x1$ as the first center.
For $j = 2, ..., k$ pick x_j as the point farthest away from the set $\{x_1, ..., x_{j-1}\}$.

Denote r as the algorithm's radius and OPT as the optimal radius.

Theorem 1.1 $\frac{r}{2} \leq OPT$

Proof 1.1 *Let $x \in A$ be the point that achieves $d(x, C) = r$, were $C = \{x_1, ..., x_k\}$ the centers. By definition: $\forall i \quad d(x, x_i) \geq r$. Because x wasn't chosen as a center (and the fact that he is far from all of the centers) we get: $\forall i \neq j \quad d(x_i, x_j) \geq r$. Therefore $x, x_1, ..., x_k$ form a $k+1$ clique of point with distance greater than r. If we map those points into the optimal solution then surely 2 points y, z will be mapped to the same center c. Note that if $d(c, y) < \frac{r}{2}$,$d(c, z) < \frac{r}{2}$ then $d(y, z) < r$, a contradiction. This derives $OPT \geq \frac{r}{2}$.*

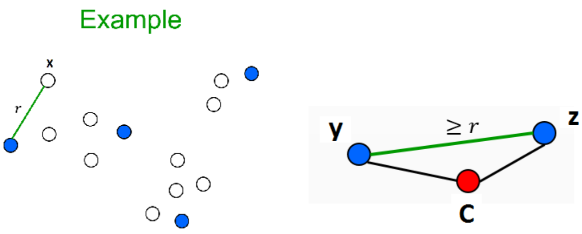

Figure 1: left - The k centers and x
right - "impossible" triangle

2 k-medians

Given a set of n points A of some metric space X, find a set C of k points in X, such that we minimize $\sum_{x \in A} d(x, C)$.

One can notice that the answer to the 1-median problem in $\mathbb{R}$ is exactly the median of the input points!
Here is a local search algorithm for the k-medians problem:

Algorithm 2 k centers approximation

Start with an arbitrary set of k centers.
Swap a center with some point which is not a center if the sum of the distances decreases.

Denote the optimal centers as $o_1, ..., o_k$ and the local search algorithm centers as $x_1, ..., x_k$.

Theorem 2.1 *Assume that* $\forall i$ o_i *is mapped to* x_i *(the mapping of an optimal center to it's closest local search center forms a matching), then* $L \leq 3OPT$.

Proof 2.1 $\forall 1 \leq i, j \leq k$ *define* $A_{i,j}$ *as the points which are closest to* o_i *and* x_j *(with the respective mappings). Also define* $\forall 1 \leq i \leq k$ $B_i = \bigcup_{j=1}^{k} A_{i,j}$ *and* $C_i = \bigcup_{j=1}^{k} A_{j,i}$. *Consider the swaps defined by this matching. By our local search definition we know* $COST(L - x_1 + o_1) - COST(L) \geq 0$. *Now we will present classification of* A *into the new centers (division of* A *into* k

group corresponding to the centers, if a point is in a center's group then we "think" of it as that center is the closest to that point - even if it's not true [it will give us an upper bound on the cost]):
o_1's will be B_1.
$\forall 2 \le i \le k$ we classify to x_i the following set: $(C_i \cup A_{i,1}) \setminus A_{1,i}$.
Note that $\forall 2 \le i \le k$ and $\forall x \in A_{i,1}$ it holds that:

$$d(x,x_i)-d(x,x_1) \le d(x,o_i)+d(o_i,x_i)-d(x,x_1) \le d(x,o_i)+d(o_i,x_1)-d(x,x_1) \le 2d(x,o_i)$$

Therefore,

$$COST(L-x_1+o_1)-COST(L) \le COST_{OPT}(B_1)-COST_L(B_1)+2COST_{OPT}(C_i)$$

Summing it for $i = 1,..,k$ and we indeed get
$0 \le OPT - L + 2OPT = 3OPT - L$, and we won!

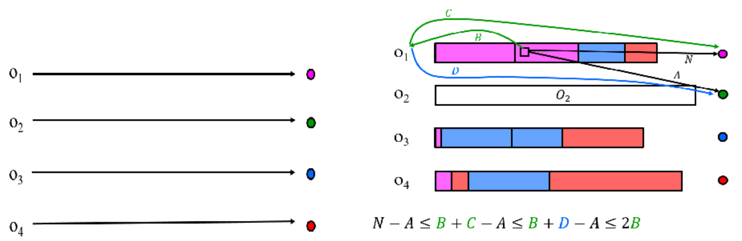

Figure 2: left - The matching between optimal centers to local search centers right - difference between distances from centers (case we swap o_2 with x_2)

3 k-means

Given a set of n points A of some metric space X, find a set C of k points in X, such that we minimize $\sum_{x \in A} d^2(x, C)$.

One can notice that the answer to the 1-median problem in $\mathbb{R}$ is exactly the average of the input points!
Here is a local search algorithm for the k-medians problem:

Algorithm 3 k means approximation

Start with an arbitrary set of k centers.
  Assign each point to its closest center
  Recalculate centers - the new centers are the means of the clusters

Note that if we look on 3-means in $\mathbb{R}$ then we can't guarantee any approximation factor. Figure 3 shows an example: taking 3 lines of distances x, z, y. if $x < y \ll z$ then one can make the local search result $\frac{y^2}{2}$ while the optimal solution is $\frac{x^2}{2}$.

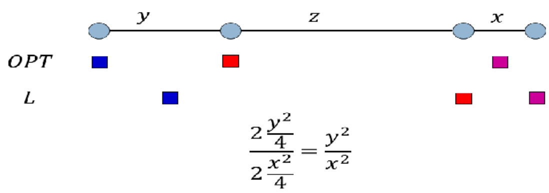

Figure 3: local solution vs. optimal solution

Running time: Note that no 2 partition can happen in 2 different iteration, this derives an upper bound on the running time of $O(k^n)$.

3.1 Voronoi diagram

The Voronoi diagram of a set of points $p_1, p_2, \ldots, p_n$ is a partition of the plane to n cells, cell i contains all points closest to p_i.

3.2 Voronoi partition

A Voronoi Partition of a set of points $p_1, p_2, \ldots, p_n$ is a partition of the points which is consistent with the voronoi diagram of the centers (of each part).

Running time: Note that no 2 voronoi partitions can appear twice, therefore the running time is bounded by the number of voronoi partition to the points.

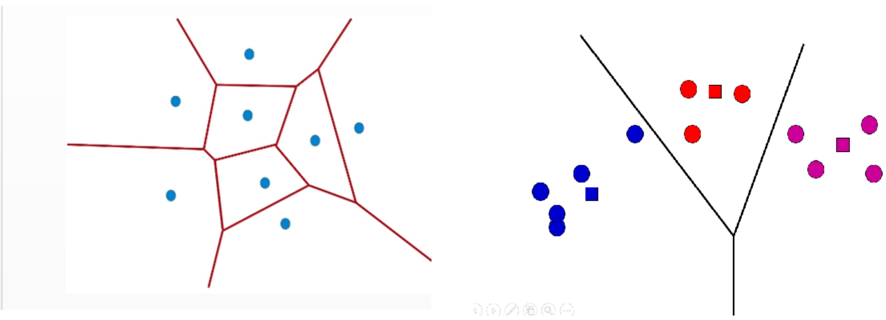

Figure 4: left - voronoi diagram.
right - voronoi partion