The kmeans++ algorithm

- Lloyd's randomized kmeans algorithm initialized the iterations with k points selected uniformly at random among $x_1, \ldots, x_m$. These points are used as the initial means.
- kmeans++ is also randomized. The only difference between kmeans++ and Lloyd's kmeans is in the procedure of selecting the initial means.

The following notation is useful for describing the kmeans++ algorithm. If U is a set of points, the distance between a point x and U is defined as follows:

$$\mathrm{dist}(x, U) = \min_{u \in U} |x - u|$$

The kmeans++ algorithm for the initial selection of means

The algorithm selects the initial k points $u_1, \ldots, u_k$ from among $x_1, \ldots, x_m$.

Input: $x_1, \ldots, x_m$, and an integer value $k \leq m$.

Output: a set U containing k points.

1. Select u_1 uniformly at random from $x_1, \ldots, x_m$. Put u_1 in U.

2 Iterate $k-1$ times, with $j = 2, \ldots, k$:

2.1 For $i = 1, \ldots, m$ compute
$$d_i = \mathrm{dist}(x_i, U)$$

2.2 Select u_j at random from $x_1, \ldots, x_m$, where the probability of selecting x_i is proportional to d_i^2.

The weighted kmeans++ algorithm for the initial selection of means

Input: $x_1, \ldots, x_m$, and associated weights $w_1, \ldots, w_m$. An integer value $k \leq m$.

Output: a set U containing k points.

The only difference between kmeans++ and weighted kmeans++ is in Step 2.2.

2.2 Select u_j at random from $x_1, \ldots, x_m$, where the probability of selecting x_i is proportional to $w_i d_i^2$.

The promise of kmeans++

Recall that k-means and k-means++ are designed to minimize the following error:

$$E(c, u_1, \ldots, u_j) = \sum_{j=1}^{k} \sum_{c(i)=j} |x_i - u_j|^2$$

Taking $u_j = \sum_{c(i)=j} x_i/m_j$ we can express the error above as a function of c as $E(c)$. Let c^* be the optimal (smallest) error, so that for each clustering c we have:

$$E(c^*) \leq E(c)$$

How close is the error of the clustering computed by by k-means to the optimum?

1. For Lloyd's algorithm no exact bound is known.

2. For k-means++ : $E(c) < E(c^*)\ O(\log k)$ in expectation.

3. There are other algorithms that give : $E(c) < E(c^*)\ O(1)$.

The algorithms mentioned in 3. do not perform well in practice. For 2. the approximation is achieved after the initial selection, before the k-means iterations.