Mining Massive Dataset
CSE6017

Dr. Ramesh Ragala

School of Computer Science and Engineering
VIT Chennai

February 10, 2021

1. K-Means++ Clustering

- **Overview on K-means clustering:**
  - k-means algorithm is an old but popular algorithm due to its simplicity and observed speed
  - Given an integer k and a set of n data points in R^d, the goal is to choose k-centers so as to minimize ϕ, the total squared distance between each point and its closest center. $\rightarrow$ Solving this problem exactly is NP-hard.
  - But 25 years ago, **Lloyd** proposed a **local search solution** to this clustering problem that is still very widely used today.
  - From this point, it is a well known geometric clustering algorithm based on work by Lloyd in 1982
  - It uses local search approach to partition the points into k-clusters
  - It seeks to minimize the average squared distance between points in the same cluster
  - A set of k-initial cluster centers is chosen arbitrarily (typically chosen uniformly at random from the data points)
  - Each point is then assigned to the center closest to it, and the centers are recomputed as centers of mass of their assigned points.
  - This process is repeated until the process stabilizes or convergences

- **Finite number of Iterations in K-means clustering:**
  - It can be shown that no partition occurs twice during the course of the algorithm $\rightarrow$ so, the algorithm is guaranteed to terminate.
  - Another reason $\rightarrow$ Since, there are only k^n possible clusterings, the process will always terminate.
  - **In practice the number of iterations is generally much less than the number of points** $\rightarrow$ Duda et al. (Text book)
  - It is the speed and simplicity of the k-means method that make it appealing, not its accuracy.
  - Indeed, there are many natural examples for which the algorithm generates arbitrarily **bad clusterings**.
  - This does not rely on **an adversarial placement of the starting centers**, and in particular, it can hold with high probability even if the **centers are chosen uniformly at random** from the data points.

- **K-means Algorithm:**
- The k-means algorithm is a simple and fast algorithm for this problem, although it offers no approximation guarantees at all.
  - **1.** Arbitrarily choose an initial k centers $\mathcal{C} = \{ \mathrm{c}_1, \mathrm{c}_2,.., \mathrm{c}_k \}$
  - **2.** For each i $\in \{$ 1, ... ,k $\}$, set the cluster C_i to be the set of points in χ that are closer to c_i than they are to c_j for all j $\neq$ i.
  - **3.** For each i $\in \{$ 1, ... ,k $\}$, set c_i to be the center of mass of all points in $\mathcal{C}_i$: $\mathrm{c}_i = \frac{1}{|\mathcal{C}_i|} \sum_{x \in \mathrm{c}_i} x$.
  - **4.** Repeat Step-2 and Step-3 until $\mathcal{C}$ no longer changes.

- **Shortcoming of K-means clustering:**
  - It has been shown that the worst case running time of the algorithm is super-polynomial in the input size.
  - The approximation found can be arbitrarily bad with respect to the objective function compared to the optimal clustering
- The k-means++ algorithm addresses the second of these obstacles by specifying a procedure to initialize the cluster centers before proceeding with the standard k-means optimization iterations.

- **Overview on K-means++ Algorithm:**
  - It was proposed in 2007 by David Arthur and Sergei Vassilvitskii.
  - It is an approximation algorithm to solve the NP-hard k-means challenge
  - Proposed a specific way of choosing centers for the k-means algorithm.
  - k-means++ have slight variation in the choosing inital centers than the k-means algorithm.
  - k-means++ algorithm is same as k-means algorithm except choosing the inital cneters at step-1 in k-means
- Thumb Rule of K-Means++ algorithm

Basic Principle:

A variant that chooses centers at random from the data points, but weighs the data points according to their squared distance squared from the closest center already chosen

- **K-means++ Algorithm:**
- Let D(x) denote the shortest distance from a data point to the closest center, which we have already chosen.
  - **1a.** Take one center c_1, chosen uniformly at random from χ
  - **1b.** Take one center c_i, choosing $x \in \chi$ with probability $\frac{D(x)^2}{\sum_{x \in \chi} D(x)^2}$
  - **1c.** Repeat Step-1b. until we have taken k centers altogether.
  - **2-4**Proceed as with the standard k-means algorithm.
- The weighting used in Step-1b simply called as "D^2 weighting".
- With the k-means++ initialization, the algorithm is guaranteed to find a solution that is O(log k) competitive to the optimal k-means solution.
- k-means++ 1% slower due to initialization.
- Since the k-means++ initialization needs k passes over the data, it does not **scale very well to large data sets**. $\rightarrow$ **Scalable k-means++** by Bahman Bahmani et al.