Long Term Memory in AI - Vector Search and Databases **COS 597A Fall 2023**

Class 7 - Clustering

Lectures: Lectures: Edo Liberty and Matthijs Douze

Warning: *Please do not cite this note as a peer reviewed source. Please submit requests and corrections as issues or pull requests at github.com/edoliberty/vector-search-class-notes*

1 K-means clustering

Definition 1.1 (k-means)**.** *Given n vectors $x_1 \ldots, x_n \in \mathbb{R}^d$, and an integer k, find k points $c_1, \ldots, c_k \in \mathbb{R}^d$ which minimize the expression:*

$$f = \sum_{i \in [n]} \min_{j \in [k]} \|x_i - c_j\|^2$$

In words, we aim to find k cluster centers. The cost is the squared distance between all the points to their closest cluster center. k-means clustering and Lloyd's algorithm [6] are probably the most widely used clustering objective and algorithm. This is for three main reasons:

- The objective function is simple and natural.
- Lloyd's algorithm (which we see below) is simple, efficient in practice, and often results in optimal or close to optimal results.
- The results are easily interpretable and are often quite descriptive for real data sets.

In 1957 Stuart Lloyd suggested a simple alternating minimization algorithm which efficiently finds a local minimum for this problem. This algorithm (a.k.a. Lloyd's algorithm) seems to work so well in practice that it is sometimes referred to as k-means or the k-means algorithm.

Algorithm 1 Lloyd's Algorithm

```
$c_1, \ldots, c_k \leftarrow$ randomly chosen centers
while Objective function still improves do
  $S_1, \ldots, S_k \leftarrow \phi$
  for $i \in 1, \ldots, n$ do
    $j \leftarrow \arg\min_{j'} \|x_i - c_{j'}\|^2\}$
    add $i$ to $S_j$
  end for
  for $j \in 1, \ldots, k$ do
    $cc_j = \frac{1}{|S_j|} \sum_{i \in S_j} x_i$
  end for
end while
```

This algorithm can be thought of as a potential function reducing algorithm. The potential function is our objective function from above.

$$f = \sum_{j \in [k]} \sum_{i \in S_j} \|x_i - c_j\|^2.$$

Where the sets S_j are the sets of points to which c_j is the closest center. In each step of the algorithm the potential function is reduced. Let's examine that. First, if the set of centers c_j are fixed, the best assignment is clearly the one which assigns each data point to its closest center. Then, is the set S_j are fixed, the optimal center is $c_j = \frac{1}{|S_j|}\sum_{i \in S_j} x_i$ (can easily be seen by derivation of the cost function). Therefore, moving c_j to it's optimal position can only reduce the potential function. The algorithm therefore terminates in a local minimum. There are only two questions. One, whether the number of iterations for convergence is bounded. Two, whether we can guaranty that the solution is close to optimal.

2 k-means and PCA

This section will present a simple connection between k-means and PCA (similar ideas given here [3]). First, consider the similarity between the k-means cost function. Let $C_k = \{c_1, \ldots, c_k\}$

$$f_{k-means} = \min_{C_k} \sum_{i \in [n]} \min_{c \in C_k} \|x_i - c\|^2$$

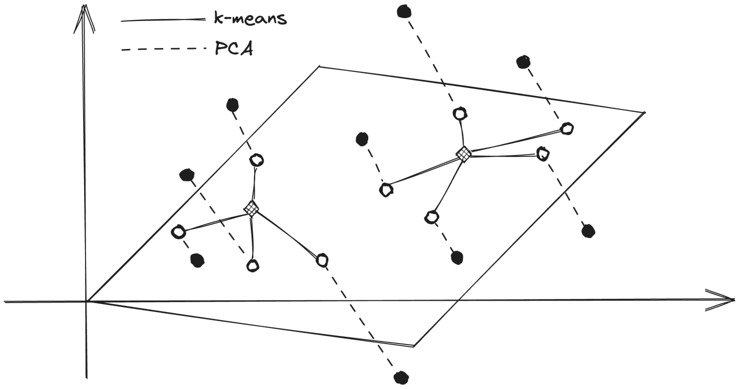
and that of PCA

$$f_{PCA} = \min_{P_k} \sum_{i \in [n]} \min_{z \in P_k} \|x_i - z\|^2$$

where P_k is a projection into dimension k and $z \in P_k$ means that $P_k z = z$. Now, think about the subspace P_k^* which contains the k optimal centers C_k^*. Since $C_j^* \subset P_k^*$ we have that:

$$f_{k-means} = \sum_{i \in [n]} \min_{c \in C_k^*} \|x_i - c\|^2 \geq \sum_{i \in [n]} \min_{z \in P_k^*} \|x_i - z\|^2 \geq f_{PCA}$$

For PCA, we conveniently have a closed form expression $\min_{z \in P_k} \|x_i - z\|^2 = \|x_i - P_k x_i\|^2$. The equality stems from the fact that for any point x and any projection operation P we have that $P(x) = \arg\min_{z \in P} \|x - z\|$. Now, consider solving k-means on the points $y_i = P_k x_i$ instead. This intuitively will be an easier task because y_i are embedded into a lower dimension, namely k (by the projection P_k).



Before we do that though, we should argue that a good clustering for y_i results in a good clustering for x_i. Note that all y_i are projected on a subspace P. If we project the optimal centers C^* onto P as well, we will get a solution with a lower cost than f_{k-mean}. The optimal solution for y_i will clearly be even better (or at least, not worse). Therefore $\hat{f}_{k-mean} \leq f_{k-mean}$ where $\hat{f}_{k-mean} = f_{k-mean}(y_1, \ldots, y_n)$.

The following gives us a simple algorithm. Compute the PCA of the points x_i into dimension k. Solve k-means on the points y_i in dimension k. Output the resulting clusters and centers.

$$f_{alg} = f_{PCA} + \hat{f}_{k-means} \leq 2 f_{k-means}$$

3 ε-net argument for fixed dimensions

Since computing the SVD of a matrix (and hence PCA) is well known. We get that computing a 2-approximation to the k-means problem in dimension d is possible if it can be done in dimension k.

To solve this problem we adopt a brute force approach. Let Q_ε be a set of points inside the unit ball B_1^k such that:

$$\forall z \in B_1^k \;\; \exists\, q \in Q_\varepsilon \; s.t. \;\; \|z - q\| \leq \varepsilon$$

Such sets of points exist such that $|Q_\varepsilon| \leq c(\frac{1}{\varepsilon})^k$. There are probabilistic constructions for such sets as well but we will not go into that. Assuming w.l.o.g. that $\|x_i\| \leq 1$ we can constrain the centers of the clusters to one of the points in the ε-net Q_ε. Let q_j be the closest point in Q_ε to c_j (so $\|c_j - q_j\| \leq \varepsilon$). From a simple calculation we have that:

$$\sum_{j\in[k]} \sum_{i\in S_j} \|x_i - q_j\|^2 \leq \sum_{j\in[k]} \sum_{i\in S_j} \|x_i - c_j\|^2 + 5\varepsilon.$$

To find the best clustering we can exhaustively search through every set of k points from Q_ε. For each such set, compute the cost of this assignment on the original points and return the one minimizing the cost. That will require $\binom{c(\frac{1}{\varepsilon})^k}{k}$ iterations over candidate solutions each of which requires $O(ndk)$ time. The final running time we achieve is $2^{O(k^2 \log(1/\varepsilon))} nd$.

4 Sampling based seeding for k-means

Another simple idea is to sample sufficiently many points from the input as candidate centers. Ideas similar to the ones described here can be found here [7].

First, assume we have only one set of points S and $|S| = n$. Also, denote by c the centroid of S, $c = \frac{1}{n}\sum_{i\in S} x_i$ and assume w.l.o.g. $c = 0$. We will claim that picking a random members of S as a centroid is not much worse than picking $c = 0$. Let q be a member of S chosen uniformly at random. Let us compute the expectation of the cost function.

$$\begin{aligned}
\mathbb{E}[\sum_{i\in S} \|x_i - q\|^2] &= \sum_{i\in S}\sum_{j\in S} \frac{1}{n}\|x_i - x_j\|^2 && (1)\\
&= \sum_{i\in S}\|x_i\|^2 - \frac{2}{n}(\sum_{i\in S} x_i)^T(\sum_{j\in S} x_j) + \sum_{j\in S}\|x_j\|^2 && (2)\\
&\leq 2\sum_{i\in S}\|x_i - c\|^2. && (3)
\end{aligned}$$

Using Markov's inequality we get that

$$\Pr[\sum_{i\in S}\|x_i - q\|^2 \leq 4\sum_{i\in S}\|x_i - c\|^2] \geq 1/2$$

If this happens we say that q is a good representative for S (at least half of the points are good representatives!) Now consider again the situation where we have k clusters $S_1, \ldots, S_k$. If we are given a set Q which contains a good candidate for each of the sets. Then, restricting ourselves to pick centers from Q will result in at most a multiplicative factor of 4 to the cost.

The set Q can be quite small if the set are roughly balanced. Let the smallest set contain n_s points. We therefore succeed in finding a good representative for any set with probability at least $\frac{1}{2}\frac{n_s}{n}$. The probability of failure for any set is thus bounded by $k(1 - \frac{n_s}{2n})^{|Q|}$. Therefore $|Q| = O(k\log(k))$ if $n_s \in \Omega(n/k)$.

Again, iterating over all subsets of Q of size k we can find an approximate solution is time $O(\binom{ck\log(k)}{k}knd) = 2^{O(k\log(k))}nd$.

5 k-means++

In the above, we gave approximation algorithms to the k-means problem. Alas, any solution can be improved by performing Lloyds algorithm on its output. Therefore, such algorithms can be considered as 'seeding' algorithms which give initial assignments to Lloyds algorithm. A well known seeding procedure [2] is called k-means++. In each iteration, the next center is chosen randomly from the input points. The distribution over the points is not uniform. Each point is picked with probability proportional to the minimal square distance from it to a picked center. Surprisingly, This simple and practical approach already gives an $O(\log(k))$ approximation guarantee. More precisely, let $f_{k-means}(C)$ denote the cost of k-means with the set of centers C. Also, denote by C^* the optimal set of centers. Then

Algorithm 2 k-means++ algorithm [2]

```
$C \leftarrow \{x_i\}$ where $x_i$ is a uniformly chosen from $[n]$.
for $j \in [k]$ do
    Pick node $x$ with probability proportional to $\min_{c \in C} \|x - c\|^2$
    Add $x$ to $C$
end for
return: $C$
```

$$\mathbb{E}[f_{k-means}(C)] \le 8(\log(k) + 2) f_{k-means}(C^*)$$

In [1] the authors give a streaming algorithm for this problem. They manipulate ideas from [2] and combine them with a hirarchical divide and conquer methodology. See also [4] for a thorough survey and new techniques for clustering in streams.

Another problem which is very related to k-means is the k-medians problem. Given a set to points $x_1, \ldots, x_n$ the aim is to find centers $c_1, \ldots, c_k$ which minimize:

$$f_{k-medians} = \sum_{i \in [n]} \min_{j \in [k]} \|x_i - c_j\|$$

Both k-means and the k-median problem admit $1 + \varepsilon$ multiplicative approximation algorithms but these are far from being simple. See [5] for more details, related work, and a new core set based solution.

6 The inverted file model

One of the most common approaches in vector search is to begin with clustering the set of points using k-mean and, at search time, consider only the points within the nearest clusters. This is called the inverted file model (IVF) and is used extensively in practice. We will expand on this discussion in class.

References

[1] Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *NIPS*, pages 10–18, 2009.

[2] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In *SODA*, pages 1027–1035, 2007.

[3] Chris H. Q. Ding and Xiaofeng He. K-means clustering via principal component analysis. In *ICML*, 2004.

[4] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice. *IEEE Trans. Knowl. Data Eng.*, 15(3):515–528, 2003.

[5] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. pages 126–134, 2005.

[6] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

[7] Hongyuan Zha, Xiaofeng He, Chris H. Q. Ding, Ming Gu, and Horst D. Simon. Spectral relaxation for k-means clustering. In *NIPS*, pages 1057–1064, 2001.