CSE 291: Unsupervised learning **Spring 2008**

Lecture 2 — The k-means clustering problem

2.1 The k-means cost function

Last time we saw the k-center problem, in which the input is a set S of data points and the goal is to choose k representatives for S. The *distortion* on a point $x \in S$ is then the distance to its closest representative. The overall goal is to make sure that *every* point in S has low distortion; that is, to minimize the *maximum* distortion in S.

In most applications, we are more interested in minimizing the *typical* (i.e., average) distortion. The most popular formulation of this is the k-means cost function, which assumes that points lie in Euclidean space.

k-MEANS CLUSTERING

Input: Finite set $S \subset \mathbb{R}^d$; integer k.

Output: $T \subset \mathbb{R}^d$ with $|T| = k$.

Goal: Minimize $\text{cost}(T) = \sum_{x \in S} \min_{z \in T} \|x - z\|^2$.

It is interesting that cost function uses the square of the L_2 norm rather than L_2 norm. This is a fortuitous choice that turns out to simplify the math in many ways.

2.1.1 Voronoi regions

The representatives T induce a *Voronoi partition* of $\mathbb{R}^d$: a decomposition of $\mathbb{R}^d$ into k convex *cells*, each corresponding to some $z \in T$ and containing the region of space whose nearest representative is z.

This partition induces an optimal clustering of the data set, $S = \cup_{z \in T} C_z$, where

$$C_z = \{x \in S : \text{the closest representative of } x \text{ is } z\}.$$

Thus the k-means cost function can equally be written

$$\text{cost}(T) = \sum_{z \in T} \sum_{x \in C_z} \|x - z\|^2.$$

In analyzing algorithms, we'll sometimes need to consider suboptimal partitions of S. To this end, define

$$\text{cost}(C_1, \ldots, C_k; z_1, \ldots, z_k) = \sum_{j=1}^{k} \sum_{x \in C_j} \|x - z_j\|^2.$$

2.1.2 A single cluster

Suppose a single cluster C is assigned representative z. The cost is then

$$\text{cost}(C; z) = \sum_{x \in C} \|x - z\|^2.$$

This is minimized when $z = \text{mean}(C)$. Moreover, the additional cost incurred by picking $z \neq \text{mean}(C)$ can be characterized very simply:

Lemma 1. *For any set $C \subset \mathbb{R}^d$ and any $z \in \mathbb{R}^d$,*

$$\text{cost}(C; z) = \text{cost}(C; \text{mean}(C)) + |C| \cdot \|z - \text{mean}(C)\|^2.$$

Contrast this with the case of k-center, where there is no closed-form expression for the optimal center of a cluster. This is a perfect example of the benefit of using average distortion and the squared L_2 norm.

Lemma 1 follows from a generic bias-variance decomposition of random vectors.

Lemma 2. *Let $X \in R^d$ be any random variable. For any $z \in \mathbb{R}^d$,*

$$\mathbb{E}\|X - z\|^2 = \mathbb{E}\|X - \mathbb{E}X\|^2 + \|z - \mathbb{E}X\|^2.$$

Proof. Expanding the right-hand side,

$$\begin{aligned}
\mathbb{E}\|X - \mathbb{E}X\|^2 + \|z - \mathbb{E}X\|^2 &= \mathbb{E}\left[\|X\|^2 + \|\mathbb{E}X\|^2 - 2X \cdot \mathbb{E}X\right] + \left[\|z\|^2 + \|\mathbb{E}X\|^2 - 2z \cdot \mathbb{E}X\right] \\
&= \mathbb{E}\|X\|^2 + \|\mathbb{E}X\|^2 - 2\mathbb{E}X \cdot \mathbb{E}X + \|z\|^2 + \|\mathbb{E}X\|^2 - 2z \cdot \mathbb{E}X \\
&= \mathbb{E}\|X\|^2 + \|z\|^2 - 2z \cdot \mathbb{E}X \\
&= \mathbb{E}\|X - z\|^2.
\end{aligned}$$

□

To see how this implies Lemma 1, let X denote a uniform random draw from cluster C. Then

$$\mathbb{E}\|X - z\|^2 = \sum_{x \in C} \frac{1}{|C|}\|x - z\|^2 = \frac{1}{|C|}\text{cost}(C; z)$$

and

$$\mathbb{E}\|X - \mathbb{E}X\|^2 = \frac{1}{|C|}\text{cost}(C; \text{mean}(C)).$$

2.2 The k-means algorithm

The name "k-means" is applied both to the clustering task defined above and to a specific algorithm that attempts (with mixed success) to solve it. Here's how the algorithm works, given a data set $S \subset \mathbb{R}^d$ and an integer k:

```
initialize centers z_1, ..., z_k ∈ R^d and clusters C_1, ..., C_k in any way
repeat until there is no further change in cost:
   for each j:  C_j ← {x ∈ S whose closest center is z_j}
   for each j:  z_j ← mean(C_j)
```

This is simple enough, and takes $O(k|S|)$ time per iteration. What can one possibly prove about it?

2.2.1 Convergence

Lemma 3. *During the course of the k-means algorithm, the cost monotonically decreases.*

Proof. Let $z_1^{(t)}, \ldots, z_k^{(t)}, C_1^{(t)}, \ldots, C_k^{(t)}$ denote the centers and clusters at the start of the t^{th} iteration of k-means. The first step of the iteration assigns each data point to its closest center; therefore

$$\text{cost}(C_1^{(t+1)}, \ldots, C_k^{(t+1)}; z_1^{(t)}, \ldots, z_k^{(t)}) \leq \text{cost}(C_1^{(t)}, \ldots, C_k^{(t)}; z_1^{(t)}, \ldots, z_k^{(t)}).$$

On the second step, each cluster is re-centered at its mean; by Lemma 1,

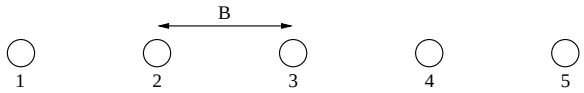
$$\text{cost}(C_1^{(t+1)}, \ldots, C_k^{(t+1)}; z_1^{(t+1)}, \ldots, z_k^{(t+1)}) \leq \text{cost}(C_1^{(t+1)}, \ldots, C_k^{(t+1)}; z_1^{(t)}, \ldots, z_k^{(t)}).$$

□

2.2.2 Initialization

We've seen that the k-means algorithm converges to a local optimum of its cost function. The quality of its final clustering depends heavily on the manner of initialization. If this isn't done right, things could go horribly wrong.

Here's an example. Suppose the data set consists of n points in five tight clusters (of some tiny radius δ) arranged in a line, with some large distance B between them:



The optimal 5-clustering has cost roughly $\delta^2 n$. If we initialize k-means by choosing five centers at random from the data, there is some chance that we'd end up with no centers from cluster 1, two centers from cluster 3, and one center each from clusters 2, 4, and 5:

In the first round of k-means, all points in clusters 1 and 2 will be assigned to the leftmost center. The two centers in cluster 3 will end up sharing that cluster. And the centers in clusters 4 and 5 will move roughly to the centers of those clusters.

Thereafter, no further changes will occur. This local optimum has cost $\Omega(B^2 n)$. We can make this arbitrarily far away from the optimum cost by setting B large enough. Thus, good initialization is crucial.

2.3 The k-means++ initializer

One idea for initializing k-means is to use a farthest-first traversal on the data set, to pick k points that are far away from each other. However, this is too sensitive to outliers. Instead, Arthur and Vassilvitskii suggest the following procedure, called k-means++: pick the k centers one at a time, but instead of always choosing the point farthest from those picked so far, choose each point at random, with probability proportional to its squared distance from the centers chosen already. Here's the algorithm (given S and k).

```
pick x ∈ S uniformly at random and set T ← {x}
while |T| < k:
   pick x ∈ S at random, with probability proportional to cost({x}; T) = min_{z∈T} ‖x − z‖²
   T ← T ∪ {x}
```

This initialization takes time $O(k|S|)$, about the same as a single iteration of k-means. Arthur and Vassilvitskii show that this initialization is itself a pretty good clustering. And subsequent iterations of k-means can only improve things.

Theorem 4. *Let T be the initial centers chosen by k-means++. Let T^* be the optimal centers. Then $\mathbb{E}[\text{cost}(T)] \leq \text{cost}(T^*) \cdot O(\log k)$, where the expectation is over the randomness in the initialization procedure.*

We won't get into the proof here, except to point out one small aspect of it. The initial centers T are not arbitrary points in $\mathbb{R}^d$ but are points from the data set itself. It turns out that in general, one loses at most a factor of two on account of this. The following lemma can be applied to each cluster within the optimal k-means clustering.

Lemma 5. *For any $C \subset \mathbb{R}^d$, pick Z uniformly at random from C. Then $\mathbb{E}[\text{cost}(C; Z)] = 2\text{cost}(C; \text{mean}(C))$.*

Proof. Let $\mu = \text{mean}(C)$. By Lemma 1,

$$\begin{aligned}
\mathbb{E}[\text{cost}(C; Z)] &= \sum_{z \in C} \frac{1}{|C|} \text{cost}(C; z) \\
&= \frac{1}{|C|} \sum_{z \in C} \left[\text{cost}(C; \mu) + |C| \cdot \|z - \mu\|^2\right] \\
&= \text{cost}(C; \mu) + \sum_{z \in C} \|z - \mu\|^2 \quad = \quad 2\text{cost}(C; \mu).
\end{aligned}$$

□

2.4 A local search heuristic for the k-means problem

Now let's look at a different heuristic that is guaranteed to return a set of k centers whose cost is within a (large) constant factor of optimal. Given a data set $S \subset \mathbb{R}^d$ and an integer k, the algorithm maintains a set $T \subset S$ of centers, $|T| = k$, and incrementally adjusts this set, changing one center at a time.

```
initialize centers T ⊂ S arbitrarily
while ∃t ∈ T, t' ∈ S such that cost(T + {t'} − {t}) < cost(T):
    T ← T + {t'} − {t}
```

This is a local search heuristic. Such algorithms are notoriously difficult to analyze, but with quite a bit of ingenuity, Arya et al were able to show the following.

Theorem 6. *Let $O \subset S$ be the subset of k data points which minimize $\text{cost}(O)$. Let T be the solution returned by the local search procedure. Then $\text{cost}(T) \leq 25\text{cost}(O)$.*

By Lemma 5, this means T is at worst a factor-50 approximation to the original k-means problem.

2.4.1 A single swap

Let's start by pinning down some notation. The algorithm returns some set of centers $T \subset S$, which partition the data into clusters $\{C_t : t \in T\}$. The optimal solution (with centers restricted to S) is $O \subset S$, with clusters $\{C_o^*; o \in O\}$. For any point x, let $t(x)$ be its closest representative to in T, and $o(x)$ its closest representative in O.

It is very difficult to keep track of what goes on during the successive iterations of local search. Instead, we will work only with the termination condition. We know that for any pair $\langle o, t\rangle$ (with $o \in O$ and $t \in T$), swapping o for t cannot decrease the cost. In other words,

$$\text{cost}(T + \{o\} - \{t\}) - \text{cost}(T) \geq 0.$$

Our overall strategy will be to carefully select k pairs $\langle o, t\rangle$, such that these k inequalities add up to something that looks roughly like $25\text{cost}(O) - \text{cost}(T) \geq 0$.

To start with, we consider a single swap, and rewrite the corresponding inequality in terms of its constituent distances.

Lemma 7. *Pick any pair $\langle o, t\rangle$ with $o \in O$, $t \in T$, that satisfies the following property:*

(*) *For each $x \in C_t$, either $o(x) = o$ or $t(o(x)) \neq t$.*

Then

$$\sum_{x \in C_o^*} \left[\|x - o(x)\|^2 - \|x - t(x)\|^2\right] + \sum_{x \in C_t} \left[\|x - t(o(x))\|^2 - \|x - t(x)\|^2\right] \;\geq\; 0.$$

Proof. We know $\text{cost}(T + \{o\} - \{t\}) - \text{cost}(T) \geq 0$. We will get an upper bound on $\text{cost}(T + \{o\} - \{t\})$ by considering a particular (possibly suboptimal) partitioning of S for centers $T + \{o\} - \{t\}$. This partitioning assigns each $x \in S$ to center $t(x)$, with two exceptions:

- any point in C_o^* gets assigned to o
- any point in $C_t \setminus C_o^*$ gets assigned to $t(o(x))$

Notice that we are removing center t, and that any point $x \in C_t$ now gets assigned either to the new center o, or to $t(o(x))$, which by hypothesis is not t. Thus this is a valid assignment of points to centers.

Using this assignment, we get the upper bound

$$\text{cost}(T + \{o\} - \{t\}) - \text{cost}(T) \;\leq\; \sum_{x \in C_o^*} \left[\|x - o\|^2 - \|x - t(x)\|^2\right] + \sum_{x \in C_t \setminus C_o^*} \left[\|x - t(o(x))\|^2 - \|x - t\|^2\right].$$

Notice that the term in the second parenthesis is always positive (since $t = t(x)$ for all such points); thus we are allowed to add in more such terms while maintaining a valid inequality. Specifically, we extend the summation to be over all of C_t, and the lemma follows. □

2.4.2 Summing over a set of swaps

The expression in Lemma 7 is suggestive. If we were to pair up the centers $\langle o, t\rangle$, and sum the resulting expressions, we would get

$$\text{cost}(O) - \text{cost}(T) + R - \text{cost}(T) \geq 0,$$

where $R = \sum_{x \in S} \|x - t(o(x))\|^2$. This is very close to the kind of bound we seek. There are only two difficulties: first, we need to upper-bound R; second, we have to limit ourselves to pairs $\langle o, t\rangle$ that satisfy the special additional condition (*) stated in Lemma 7.

Let's deal with these problems one by one. First we upper bound R. Looking at its definition, we see that each point x is being assigned to a center in T, but not necessarily the best choice $t(x)$. Thus we can expect R to be more than $\text{cost}(O)$ and more than $\text{cost}(T)$. Using the triangle inequality, we get the following upper bound.

Lemma 8. $R \leq 2\text{cost}(O) + \text{cost}(T) + 2\sqrt{\text{cost}(O)\text{cost}(T)}$.

Proof. To be filled in. □

Next, we need to carefully select pairs $\langle o, t\rangle$. Ideally, we like to find some pairing of the centers in O and T that satisfy condition (*), but this may not be possible. Instead, we will find k pairs $\langle o, t\rangle$ such that:

- Each $o \in O$ occurs in exactly one pair.
- Each $t \in T$ occurs in at most two pairs.
- Each $\langle o, t\rangle$ satisfies condition (*).

(Details of how to do this pairing need to be filled in.) Summing the expressions in Lemma 7, we then get

$$\begin{aligned} 0 &\leq \sum_{x\in S}\left[\|x - o(x)\|^2 - \|x - t(x)\|^2\right] + 2\sum_{x\in S}\left[\|x - t(o(x))\|^2 - \|x - t(x)\|^2\right] \\ &= \text{cost}(O) - 3\text{cost}(T) + 2R \\ &\leq 5\text{cost}(O) - \text{cost}(T) + 4\sqrt{\text{cost}(O)\text{cost}(T)} \\ &= \left(5\sqrt{\text{cost}(O)} - \sqrt{\text{cost}(T)}\right)\left(\sqrt{\text{cost}(O)} + \sqrt{\text{cost}(T)}\right) \end{aligned}$$

It follows that $\sqrt{\text{cost}(T)} \leq 5\sqrt{\text{cost}(O)}$ and thus $\text{cost}(T) \leq 25\text{cost}(O)$.

2.5 Open problems

Problem 1. It is known that the k-means problem is NP-hard even when $k = 2$. This particular reduction creates instances whose dimension is $d = \Theta(n)$. Is k-means clustering hard even when $d = 2$? Is it hard to approximate within a constant factor?

Problem 2. Are there better, practical approximation algorithms for k-means clustering?

Problem 3. Characterize the various kinds of local optima into which the k-means algorithm can fall. A possible starting point is to consider the case of well-separated clusters.

Problem 4. We've seen two reasonable ways to initialize the k-means algorithm: by k-means++ and by the local search heuristic. Which works better in practice?

Problem 5. The EM clustering algorithm (for mixtures of Gaussians) is more general than k-means in that it allows *soft* clustering (in which each data point can be divided fractionally between clusters rather than being definitively assigned to just one of them) and permits clusters of arbitrary ellipsoidal shapes (while k-means is geared towards spherical clusters of the same radius). Is there a way to use ideas from our two k-means approximation algorithms to give an approximation algorithm for the EM objective function? This would be interesting even for the simpler case of soft k-means.