CSE 347 Lecture 26

Smoothed Analysis

1. k-means clustering & Lloyd's Algorithm
2. Smoothed Analysis
3. Smoothed runningtime of Lloyd's Algorithm

"Clustering" is a widely used "unsupervised" machine learning problem formulation. Intuitively, we wish to discover some structure in a data set by identifying clumps of points that are near each other, but perhaps far from the points in another clump. These sets of points are called "clusters," and the hope is usually that the membership in a cluster indicates some interesting property of its constituent points that was not explicitly represented in the data set. For example, the points may correspond to documents, represented by a list of which words do or do not appear in that document. One may hope that the clusters represent the topics of the documents, for example. One can hope for many such things, and in general these hopes may not come true. But, clustering is widely used. To be more precise, here is one widely-used objective for clustering, known as <u>k-means</u>: for a set of points $\mathcal{P} \subseteq \mathbb{R}^n$, we wish to find a collection

	Words			
	Word 1	Word 2	⋯	Word n
Doc. 1	0	1		0
Doc. 2	1	1		0
⋮				
Doc m	1	0		1

of k <u>cluster centers</u> $c_1, \dots, c_k$ such that if $\mathcal{P}_i \subseteq \mathcal{P}$ is the set of points assigned to the center c_i, we minimize $\sum_{i=1}^{k} \sum_{x \in \mathcal{P}_i} \|x - c_i\|^2$ where $\|x - c_i\|$ is the standard, Euclidean distance

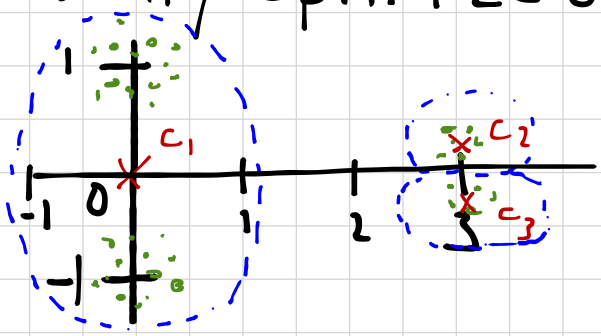
i.e., $\|x - c_i\|^2 = \sum_{j=1}^{n} (x_j - c_{ij})^2$

A very widely used algorithm for the k-means objective is Lloyd's Algorithm (so widely used that it is often called "<u>the</u> k-means algorithm," although this is an abuse of the term k-means...)

Given any initial set of k centers $c_1, \dots, c_k \in \mathbb{R}^n$,

Repeat:
  For each $x \in \mathcal{P}$, put x in $\mathcal{P}_i$ s.t. $\|x - c_i\|$ is minimized.
  For $i = 1, \dots, k$, put $c_i = \frac{1}{|\mathcal{P}_i|} \sum_{x \in \mathcal{P}_i} x$
Until no $\mathcal{P}_i$ changes during some iteration.

The first thing to note about k-means is that it does <u>not</u> actually optimize our objective function. Consider:



... where we'd prefer say, $c_1 = (0, 1)$, $c_2 = (0, -1)$, and $c_3 = (3, 0)$. But, Lloyd's Algorithm may return the shown clustering.

We can at least show that the algorithm does find some "local" optimum, i.e., that the loop terminates.

<u>Lemma 1</u> For a set of points $\mathcal{S}$, let $c(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} x$ be the <u>centroid</u>. Then for any c', $\sum_{x \in \mathcal{S}} \|x - c'\|^2 = \sum_{x \in \mathcal{S}} \|x - c(\mathcal{S})\|^2 + |\mathcal{S}| \cdot \|x - c'\|^2$

<u>Proof</u>: Write $\sum_{x \in \mathcal{S}} \|x - c'\|^2 = \sum_{x \in \mathcal{S}} \|x - c(\mathcal{S}) + c(\mathcal{S}) - c'\|^2$

$= \sum_{x \in \mathcal{S}} \|x - c(\mathcal{S})\|^2 + 2(c(\mathcal{S}) - c')(x - c(\mathcal{S})) + \|c(\mathcal{S}) - c'\|^2)$

$= \sum_{x \in \mathcal{S}} \|x - c(\mathcal{S})\|^2 + 2(c(\mathcal{S}) - c') \sum_{x \in \mathcal{S}} (x - c(\mathcal{S})) + |\mathcal{S}| \cdot \|c(\mathcal{S}) - c'\|^2$

since $\sum_{x \in \mathcal{S}} x = |\mathcal{S}| \cdot c(\mathcal{S})$. ▨

Now, why does Lemma 1 imply the loop terminates? If we consider the potential function $\varphi = \sum_{i=1}^{k} \sum_{x \in P_i} \|x - c_i\|^2$, we first observe that it is never negative, and the first step of Lloyd's algorithm re-assigns each x to minimize φ given $c_1, \ldots, c_k$, so it does not increase φ. Likewise, by our Lemma, re-assigning c_i to the centroid of P_i also decreases φ. Thus, φ decreases on every iteration. Since there are at most $k^{|P|}$ possible assignments and they can't repeat, the algorithm terminates.

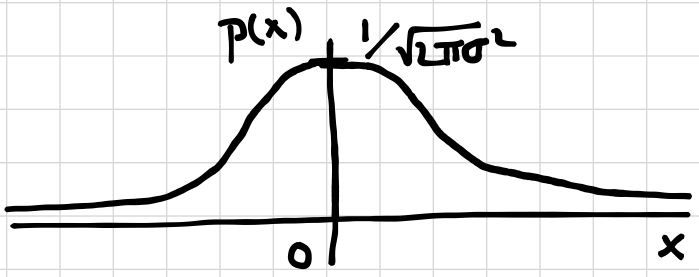
The key question is, how many iterations does this actually take? k-means is popular because empirically, it is very fast. It had been used for nearly 50 years before Arthur and Vassilivitskii found an example set of points (of size m) for which the loop runs for $2^{\Omega(\sqrt{m})}$ iterations.

Smoothed Analysis

How can we reconcile the apparent speed of Lloyd's Algorithm with the fact that it runs in exponential time? An answer is given by "smoothed analysis" — roughly, we assume that the data is a little noisy:

Definition For an algorithm that takes as input an $n \times m$ matrix X, and runs in time $T(X)$, and a noise parameter $\sigma > 0$, the **smoothed running time** on $n \times m$ inputs is $\max_X \mathbb{E}_G[T(X+G)]$ where G is an $n \times m$ matrix of Gaussian (Normal) random variables with mean 0 **and** variance σ^2. (We also consider the distribution of $T(X+G)$.)

Recall that the Gaussian distribution has probability density function $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|x-\mu|^2}{2\sigma^2}}$ (here, $\mu = 0$). It is the familiar "bell curve"



Natural variation, measurement error, and so on all empirically seem to roughly follow a Gaussian distribution (The "Central Limit Theorem" in Statistics gives some theoretical justification for this observation.) So, in contrast to average-case analysis, which requires us to specify the distribution of X, it seems reasonable in many cases to assume that the data indeed includes some noise, for a small but non-negligible σ.

How does the noise help us? Precisely, it ensures that the points cannot be carefully arranged:

Lemma 2 If g is a vector of n independent Gaussians of variance σ^2 (and any mean) then for any $y \in \mathbb{R}^n$, $Pr_g[\|g-y\| \leq \varepsilon] < \left(\frac{\varepsilon}{\sigma}\right)^n$.

Proof: Note that at any $x \in \mathbb{R}^n$, the probability density of g at x is at most $\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n$. Moreover, every x with $\|x-y\| \leq \varepsilon$ is within ε in each coordinate. So we can crudely approximate the probability:

$$\int_{\|x-y\|\leq\varepsilon} \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n e^{-\frac{\|x-\mu\|^2}{2\sigma^2}} dx \leq \int_{y_1-\varepsilon}^{y_1+\varepsilon} \cdots \int_{y_n-\varepsilon}^{y_n+\varepsilon} \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n dx_n \cdots dx_1 = \left(\frac{2\varepsilon}{\sqrt{2\pi\sigma^2}}\right)^n < \left(\frac{\varepsilon}{\sigma}\right)^n \blacksquare$$

So, how does this help? It will give us the following two key properties. (1) Consider any four values, (w, x, y, z) of the form $\frac{q}{r}c(\mathcal{A})$ where $\mathcal{A} \subseteq \mathcal{P}$ has at most $4kn$ points, and $\frac{q}{r}$ is in lowest terms with $q \leq m^2$ and $r < m$. $\mathcal{P}$ is "δ-km-sparse" if any such (w,x,y,z) with $\|w+x-(y+z)\| \leq \delta$ has identical lin. comb. of $\mathcal{P}$: $w+x \equiv y+z$. (2) $\mathcal{P}$ is "ε-separated" if for any (n-1 dimensional) hyperplane $\mathcal{H}$, at most $2n$ points in $\mathcal{P}$ lie within distance ε of $\mathcal{H}$. (Recall, $\mathcal{H} = \{x : \langle x, v\rangle = b\}$ for some choice of $v \in \mathbb{R}^n$ and $b \in \mathbb{R}$.)

We can show that with high probability, both hold:

Lemma 3 A σ-smoothed $\mathcal{P}$ is δ-km-sparse with probability at least $1-m^{16kn+12}\left(\frac{m^4\delta}{\sigma}\right)^n$

Lemma 4 A σ-smoothed $\mathcal{P}$ is ε-separated with probability at least $1-m^{2n}\left(\frac{4n\varepsilon}{\sigma}\right)^n$

Given these properties of the data, we can bound the number of iterations as follows. First, we'll argue

Lemma 5: If $\mathcal{P}$ is ε-separated and some cluster gains or loses more than $2kn$ points in an iteration, then $\varphi\left(=\sum_{i=1}^{k}\sum_{x\in P_i}\|x-c_i\|^2\right)$ drops by $\geq \frac{4\varepsilon^2}{m}$.

Then we'll show

Lemma 6: If $\mathcal{P}$ is $2m^2\sqrt{C}$-km-sparse, then after 2^k iterations, either a cluster gains or loses $\geq 2kn$ points in an iteration, or the potential φ drops by at least C.

Taking an appropriate choice of $C\left(=\frac{\sigma^2\delta^{2/n}}{4m^{32k+36}}\right)$ and $\varepsilon\left(=\frac{\sigma\delta^{1/n}}{4nm^2}\right)$ gives that with probability $1-2\delta$, $\mathcal{P}$ is both ε-separated and $2m^2\sqrt{C}$-km-sparse, so φ decreases by at least $C\left(<\frac{4\varepsilon^2}{m}\right)$ every 2^k iterations, and thus since $\varphi \leq n\cdot\max_{x,y\in\mathcal{P}}\|x-y\|^2$ after one iteration,

Theorem With probability $1-2\delta$, Lloyd's Algorithm runs in time polynomial in m^k, $\left(\frac{1}{\delta}\right)^{1/n}$, and $\frac{\max_{x,y\in\mathcal{P}}\|x-y\|}{\sigma}$ on σ-smoothed $\mathcal{P}$.

(A better bound, that is polynomial in k, is also possible...)

This is our desired running time bound. So let's see the lemmas.

Proof of Lemma 5: Consider a P_i that gains or loses more than $2kn$ points: some other $P_{i'}$ must exchange $\geq 2n+1$ of these by the Pigeonhole principle. Consider the hyperplane $\mathcal{H}$ bisecting the segment $(c_i, c_{i'})$. Since $\mathcal{P}$ is ε-separated, some x that

switched must lie distance $\geq \varepsilon$ from $\mathcal{H}$, so φ decreases by at least $\left| \|c_i - x\|^2 - \|c_{i'} - x\|^2 \right| = \left| \langle c_i, c_i \rangle - 2\langle x, c_i \rangle + \langle x, x \rangle - (\langle c_{i'}, c_{i'} \rangle - 2\langle x, c_{i'} \rangle + \langle x, x \rangle) \right|$

$$= \left| \langle c_i, c_i - c_{i'} \rangle + \langle c_i, c_{i'} \rangle + 2\langle x, c_{i'} - c_i \rangle - \langle c_{i'}, c_{i'} - c_i \rangle - \langle c_{i'}, c_i \rangle \right|$$

$$= \left| \langle 2x - c_i - c_{i'}, c_{i'} - c_i \rangle \right|$$

where $\mathcal{H}$ is centered at $\frac{c_i + c_{i'}}{2}$, and x lies at least ε-far from $\mathcal{H}$ in the orthogonal direction $\frac{c_i - c_{i'}}{\|c_i - c_{i'}\|}$. Moreover, since x was in one of the clusters, which contain $\leq m$ points, the center of this cluster is $\geq \frac{\varepsilon}{m}$-far from $\mathcal{H}$. But $\mathcal{H}$ bisects $(c_i, c_{i'})$, so $\|c_i - c_{i'}\| \geq \frac{2\varepsilon}{m}$, and thus the potential drops by $\geq 2\varepsilon \cdot \frac{2\varepsilon}{m}$ ☐

Proof of Lemma 6 We'll show that as long as no cluster gains or loses $2kn$ points and the potential does not drop by C, each cluster can take at most two distinct point sets (hence, centers). The 2^k iteration bound is then immediate. So, we suppose for contradiction that some cluster i takes three sets, $\mathcal{S}_1, \mathcal{S}_2$, and $\mathcal{S}_3$. WLOG, both $\mathcal{S}_1$ and $\mathcal{S}_3$ switch to or from $\mathcal{S}_2$ at some point. Let $A = \mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{S}_3$, and $\Delta_1 = \mathcal{S}_1 - A$, $\Delta_2 = \mathcal{S}_2 - A$, $\Delta_3 = \mathcal{S}_3 - A$. We find

$$|\Delta_1| \leq |\mathcal{S}_1 - \mathcal{S}_1 \cap \mathcal{S}_2| + |\mathcal{S}_1 \cap \mathcal{S}_2 - \mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{S}_3| \leq \underbrace{|\mathcal{S}_1 - \mathcal{S}_1 \cap \mathcal{S}_2|}_{\leq 2kn} + \underbrace{|\mathcal{S}_2 - \mathcal{S}_2 \cap \mathcal{S}_3|}_{\leq 2kn}$$

since no cluster gains/loses $2kn$ points on a transition. Similarly, also, $|\Delta_2|, |\Delta_3| \leq 4kn$. We also find by Lemma 1 that if a cluster center moves by $\sqrt{C}$ on any iteration, that φ drops by $\geq C$. Therefore also $\|c(\mathcal{S}_1) - c(\mathcal{S}_2)\| \leq \sqrt{C}$. Observe that we can write

$$c(\mathcal{S}_2) - c(\mathcal{S}_1) = \frac{|A|\,c(A) + |\Delta_2|\,c(\Delta_2)}{|A| + |\Delta_2|} - \frac{|A|\,c(A) + |\Delta_1|\,c(\Delta_1)}{|A| + |\Delta_1|}.$$

"Solving" for A,

$$(|\Delta_1| - |\Delta_2|)\,|A|\,c(A) = (|A| + |\Delta_1|)(|A| + |\Delta_2|)\underbrace{(c(\mathcal{S}_2) - c(\Delta_1))}_{\leq \sqrt{C}} - |\Delta_2|\underbrace{(|A| + |\Delta_1|)}_{\leq |\mathcal{S}_1|}\,c(\Delta_2) + |\Delta_1|\underbrace{(|A| + |\Delta_2|)}_{\leq |\mathcal{S}_2|}\,c(\Delta_1)$$

Observe that we can't have $|\Delta_1| = |\Delta_2|$, since then we'd get

$|\Delta_1|(|A|+|\Delta_2|)c(\Delta_1)$ and $|\Delta_2|(|A|+|\Delta_1|)c(\Delta_2)$ are within $m^2\sqrt{c}$ of each other, so since $\mathcal{P}$ is $2m^2\sqrt{c}$-km-sparse, these are identical linear combinations over $\mathcal{P}$, so $\Delta_1=\Delta_2$, implying $\mathcal{S}_1=\mathcal{S}_2$ ✳

So, we can divide by $|\Delta_1|-|\Delta_2|$ to solve for $|A|c(A)$. We find $\frac{|\Delta_1|(|A|+|\Delta_2|)}{|\Delta_1|-|\Delta_2|}c(\Delta_1)-\frac{|\Delta_2|(|A|+|\Delta_1|)}{|\Delta_1|-|\Delta_2|}c(\Delta_2)$ is within $m^2\sqrt{c}$ of $|A|c(A)$ and by a similar argument applied to $\mathcal{S}_2$ and $\mathcal{S}_3$, also $\frac{|\Delta_3|(|A|+|\Delta_2|)}{|\Delta_3|-|\Delta_2|}c(\Delta_3)-\frac{|\Delta_2|(|A|+|\Delta_1|)}{|\Delta_3|-|\Delta_2|}c(\Delta_2)$ is within $m^2\sqrt{c}$ of $|A|c(A)$. Thus, these expressions are within $2m^2\sqrt{c}$, so since $\mathcal{P}$ is $2m^2\sqrt{c}$-km-sparse, these are identical linear combinations over $\mathcal{P}$. We note that we can't have both $\Delta_1\cap\Delta_2=\phi$ and $\Delta_2\cap\Delta_3=\phi$, since then the points in the first expression with nonzero coefficient from Δ_1 must be the same as with nonzero coefficient in the second, so $\Delta_1=\Delta_3$ and then $\mathcal{S}_1=\mathcal{S}_3$ ✳ So, suppose there is a point in $\Delta_1\cap\Delta_2$. Its coefficient in the first expression is $\frac{|\Delta_1|(|A|+|\Delta_2|)}{|\Delta_1|-|\Delta_2|}\frac{1}{|\Delta_1|}-\frac{1}{|\Delta_2|}\frac{|\Delta_2|(|A|+|\Delta_1|)}{|\Delta_1|-|\Delta_2|}$ $=-1$, but since this point is not in A, it can't be in Δ_3, so its coefficient in the second expression must be $-\frac{|A|+|\Delta_3|}{|\Delta_3|-|\Delta_2|}$. But, since Δ_2 is nonempty, $|A|+|\Delta_2|>0$, so $\frac{|A|+|\Delta_3|}{|\Delta_3|-|\Delta_2|}>1$, and we see $-\frac{|A|+|\Delta_3|}{|\Delta_3|-|\Delta_2|}$ can't be -1 ✳ By essentially the same argument, there can't be a point in $\Delta_2\cap\Delta_3$, either ✳ Thus, no cluster uses three distinct sets. ▨