CSE 291: Geometric algorithms **Spring 2013**

Lecture 1 — Clustering in metric spaces

1.1 Why clustering?

A common use of clustering is to approximate a large/infinite/continuous set by a finite set of representatives. This is the case, for instance, when *vector quantization* is used in audio processing. A speech signal is broken down into (typically overlapping) windows, each representing 25 milliseconds, say. The continuous signal within each 25 msec window is then quantized by replacing it by its nearest representative in a finite *codebook* of 25 msec signals.

Before we can formalize clustering problems, we need to describe the kind of space in the which the data are contained.

1.2 Metric spaces

There is an endless diversity of data out there, and their underlying spaces have all kinds of geometries. There is no single umbrella notion of "distance" that captures all these possibilities. A reasonable starting point, however, is the notion of *metric space*.

1.2.1 Definition and examples

Definition 1. A metric space $(\mathcal{X}, \rho)$ consists of a set $\mathcal{X}$ and a distance function $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies the three properties of a metric:

1. Reflexivity: $\rho(x, y) \geq 0$ with equality iff $x = y$
2. Symmetry: $\rho(x, y) = \rho(y, x)$
3. Triangle inequality: $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$

Example 2. d-dimensional Euclidean space, $(\mathbb{R}^d, L_2)$. Here the distance function is

$$\rho(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}.$$

Example 3. $(\mathbb{R}^d, L_1)$. The L_1 metric is

$$\rho(x, y) = \|x - y\|_1 = \sum_{i=1}^{d} |x_i - y_i|.$$

Example 4. $(\mathbb{R}^d, L_\infty)$. The L_∞ metric is

$$\rho(x, y) = \|x - y\|_\infty = \max_i |x_i - y_i|.$$

Example 5. (M, ρ) where M is a Riemannian manifold and ρ is geodesic distance along the manifold.

Example 6. (V, ρ) where V are the vertices of an undirected graph with positive edge lengths, and $\rho(x, y)$ is the shortest path distance between x and y in the graph.

1.3 The k-center problem

Fix any metric space $(\mathcal{X}, \rho)$. The *k-center* problem asks: given a set S and an integer k, find the smallest radius r such that S is contained within k balls of radius r.

k-CENTER CLUSTERING

Input: Finite set $S \subset \mathcal{X}$; integer k.

Output: $T \subset \mathcal{X}$ with $|T| = k$.

Goal: Minimize $\text{cost}(T) = \max_{x \in S} \rho(x, T)$.

Here $\rho(x, T)$ is the distance from point x to the closest point in set T, that is to say, $\inf_{z \in T} \rho(x, z)$.

In an L_∞ space, this says: find the smallest r such that S can be covered by k boxes of side length $2r$. In an L_2 space, it says: find the smallest r such that S can be covered by k spheres of radius r. And so on.

1.3.1 Farthest-first traversal

A basic fact about the k-center problem is that it is NP-hard. Thus there is no efficient algorithm that always returns the right answer. But here's a good algorithm due to González (1985) called *farthest first traversal.*

```
pick any $z \in S$ and set $T = \{z\}$
while $|T| < k$:
    $z = \arg\max_{x \in S} \rho(x, T)$
    $T = T \cup \{z\}$
```

This builds a solution T one point at a time. It starts with any point, and then iteratively adds in the point furthest from the ones chosen so far.

Farthest-first traversal takes time $O(k|S|)$, which is fairly efficient. Its solution might not be perfect, but is always close to optimal, in the following sense.

Claim 7. *If T is the solution returned by farthest-first traversal, and T^* is the optimal solution, then*

$$\text{cost}(T) \leq 2\text{cost}(T^*).$$

Proof. Let $r = \max_{x \in S} \rho(x, T)$ be the cost of T, and let x_0 be the point at which this maximum is achieved. Then $T \cup \{x_0\}$ consists of $k + 1$ points which are all distance $\geq r$ apart. Two of these points must have the same closest representative in T^* (since $|T^*| = k$). So two points a distance $\geq r$ apart are assigned the same representative in T^*. This means $\text{cost}(T^*) \geq r/2$. □

Interestingly, it is not possible to achieve a better approximation ratio for arbitrary metric spaces: even getting a factor $2 - \epsilon$ (for any $\epsilon > 0$) is NP-hard. A variety of hardness results, for k-center and the closely related problem of *max-diameter clustering*, can be found in González (1985) and Feder and Greene (1988).

1.3.2 Covering numbers

Fix any metric space $(\mathcal{X}, \rho)$. For any $\epsilon > 0$, an ϵ-cover of a set $S \subset \mathcal{X}$ is defined to be any set $T \subset \mathcal{X}$ such that

$$\sup_{x \in S} \rho(x, T) \leq \epsilon.$$

In words, an ϵ-cover of S is a (typically smaller) set of points T which constitute a good approximation to S in the sense that any point in S can be replaced by a point in T that is at most ϵ away from it.

Example 8. Suppose the metric space is $(\mathbb{R}^d, L_\infty)$ and $S = \{-1, 1\}^d$, the vertices of a d-dimensional hypercube.

In this case, there is a 1-cover consisting of just a single point, the origin. However, for $\epsilon < 1$, any ϵ-cover T must contain 2^d points. To see this, notice that T must have some point whose coordinates are all strictly positive, to cover $(1, 1, \ldots, 1) \in S$. Similarly, T must have a point whose coordinates are all strictly negative, to cover $(-1, -1, \ldots, -1) \in S$. Continuing in this fashion, T must contain points that lie strictly within every one of the 2^d orthants. Therefore $|T| \geq 2^d$. Of course, $T = S$ always works.

Example 9. Metric space $(\mathbb{R}^2, L_\infty)$ and $S = [-1, 1]^2$.

The simplest 1-cover is $T = \{(0, 0)\}$. The best $(1/2)$-cover consists of the four points $T = \{(\pm 1/2, \pm 1/2)\}$. When $\epsilon = 1/2^k$, an ϵ-cover needs to cover the square $[-1, 1]^2$ by smaller squares of side length 2ϵ; so a cover of size $1/\epsilon^2$ is necessary and sufficient.

Example 10. Metric space $(\mathbb{R}^d, L_\infty)$ and $S = [-1, 1]^d$.

This is just like the previous example, except that now the hypercube $[-1, 1]^d$ is to be covered by smaller hypercubes of side length 2ϵ. Therefore $1/\epsilon^d$ of them are needed.

Example 11. Metric space $(\mathbb{R}^2, L_1)$ and $S = [-1, 1]^2$.

The single point $\{(0, 0)\}$ is a 2-cover of S. To get a 1-cover, we can use the four points $\{(0, \pm 1), (\pm 1, 0)\}$.

Notice that the characteristic shape of the L_∞ metric is a box, while that of the L_1 metric is a diamond; that is to say, an ϵ-cover in L_∞ is a cover by boxes of size proportional to ϵ while an ϵ-cover in L_1 is a cover by diamonds of size proportional to ϵ. Similarly, the characteristic shape of the L_2 metric is the sphere.

1.3.3 Computing covering numbers

In a metric space $(\mathcal{X}, \rho)$, the *ϵ-covering number* of a set $S \subset \mathcal{X}$ is the size of its smallest ϵ-cover. Specifically, define

$$N(S, \epsilon) = \min\{|T| : T \text{ is an } \epsilon\text{-cover of } S\}.$$

One way to approximate $N(S, \epsilon)$ is by farthest-first traversal:

```
pick any $z \in S$ and set $T = \{z\}$
while $\max_{x \in S} \rho(x, T) > \epsilon$:
    $z = \arg\max_{x \in S} \rho(x, T)$
    $T = T \cup \{z\}$
return |T|
```

By Claim 7, the returned value $|T|$ is guaranteed to satisfy:

$$N(S, \epsilon) \leq |T| \leq N(S, \epsilon/2).$$

This is often not a very strong guarantee. For d-dimensional data, it is frequently the case that $N(S, \epsilon) \approx (1/\epsilon)^d$; in such situations, the approximate covering number could be off by a multiplicative factor of 2^d.

1.4 Problems

1. *Max-diameter clustering.* Here's a problem that is very similar to k-center:

   Input: A finite set of points S in some metric space, say $(\mathcal{X}, \rho)$; an integer k.

   Output: A partition of S into k clusters.

   Goal: Minimize the maximum diameter of the clusters. That is, the cost of a partition $S = C_1 \cup C_2 \cup \cdots \cup C_k$ is
   $$\max_j \max_{x,x' \in C_j} \rho(x, x').$$

   Show that the farthest-first heuristic also gives a factor-2 approximation for this problem.

2. *Another factor-2 approximation algorithm for max-diameter clustering.* Here's an alternative heuristic for the max-diameter clustering problem, due to Hochbaum and Shmoys (1985). Given as input a finite set of points S from a metric space, and an integer k:

   - Guess the optimal diameter D
   - $T = \emptyset$ (set of cluster centers)
   - while S is not empty:
     - Pick any $x \in S$ and add it to T
     - $S = S \setminus B(x, D)$

   (a) When this procedure terminates, the set T contains a subset of S such that every point in S lies within distance D of S: in other words, this is a clustering with diameter at most $2D$. Now, show that if we guess the correct optimal diameter, then $|T| \leq k$.

   (b) Of course, we have no idea what the optimal diameter is, so we need to try all possible values. At most how many possibilities do we need to investigate? What is the final running time of this algorithm?

Bibliography

Feder, T. and Greene, D. (1988). Optimal algorithms for approximate clustering. In *ACM Symposium on Theory of Computing*.

González, T. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306.

Hochbaum, D. and Shmoys, D. (1985). A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184.