CSE 291: Unsupervised learning **Spring 2008**

Lecture 1 — Clustering in metric spaces

1.1 Why clustering?

In this course we will study two distinct uses of clustering:

1. To approximate a large/infinite/continuous set by a finite set of representatives.

   This is the case 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.

2. To find meaningful clusters in data.

   This is the case during exploratory data analysis, when a new data set is first examined to get a sense of its gross structure. At this stage, it is useful to get an idea of significant cluster structure.

   Another example is topic modeling, where documents or images need to be grouped in a manner that is meaningful to humans.

Sometimes the same algorithms are used in both these contexts. However, they have different notions of success and we will treat them separately in this course. We'll start with (1), which has traditionally been the more clearly defined.

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.

1.2.2 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

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

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 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 7. 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 8. 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 9. 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 10. 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 The k-center problem

Fix any metric space $(\mathcal{X}, \rho)$. The *k-center* problem asks: given a set S and an integer k, what is the smallest ϵ for which you can find an ϵ-cover of S of size k?

> k-CENTER CLUSTERING
>
> *Input:* Finite set $S \subset \mathcal{X}$; integer k.
>
> *Output:* $T \subset \mathcal{X}$ with $|T| = k$.
>
> *Goal:* Minimize $\mathrm{cost}(T) = \max_{x \in S} \rho(x, T)$.

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

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

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 $\mathrm{cost}(T^*) \geq r/2$. □

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.

1.3.2 Open problems

Although it is useful to be able to solve k-center in general metric spaces, the specific case of greatest interest is d-dimensional Euclidean space, $(\mathbb{R}^d, L_2)$. There are several open problems here.

Problem 1. Hardness results in Euclidean space. How does the hardness of the problem depend upon k, $|S|$, and d? For instance, is it hard even for $d = 2$? Certainly, when $d = 1$ the problem can be solved efficiently by dynamic programming. How about hardness of approximation?

Problem 2. For data in Euclidean space, is there an efficient algorithm that achieves better than a factor 2 approximation?

Problem 3. For data in Euclidean space, is there an algorithm that seems to work better in practice than farthest-first traversal?

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

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

This turns out to be a fundamental quantity in empirical process theory; later on, we'll see several reasons for wanting to compute it.

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 11, 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.

Problem 4. Develop a better approximation algorithm for covering numbers.