CSE 291: Unsupervised learning

Homework 1

This homework is due on Thursday April 20 at 11.59pm.

- All homeworks must be typewritten and uploaded to Gradescope.
- No late homeworks will be accepted.

1. *Checking metric properties.* Which of these distance functions is a *metric*? If it is not a metric, state which of the four metric properties it violates.
   (a) Let $\mathcal{X} = \mathbb{R}$ and define $d(x, y) = x - y$.
   (b) Let Σ be a finite set and $\mathcal{X} = \Sigma^m$. The *Hamming distance* on $\mathcal{X}$ is $d(x, y) = \#$ of positions on which x and y differ.
   (c) Squared Euclidean distance on $\mathbb{R}^m$, that is, $d(x, y) = \sum_{i=1}^m (x_i - y_i)^2$. (It might be easiest to consider the case $m = 1$.)

2. *Norms.* In class, we talked about ℓ_p norms on $\mathbb{R}^m$, which include the ℓ_1, ℓ_2, and ℓ_∞ norms. We now define norms more generally. A function $f : \mathbb{R}^m \to \mathbb{R}$ is a *norm* if:
   - It is nonnegative: $f(x) \geq 0$ always.
   - $f(x) = 0$ if and only if $x = 0$.
   - It is homogeneous: $f(tx) = |t|f(x)$ for any $x \in \mathbb{R}^m$ and $t \in \mathbb{R}$.
   - It satisfies the triangle inequality: $f(x + y) \leq f(x) + f(y)$.

   Note, for instance, that the ℓ_1 norm satisfies these properties.

   (a) *Convexity of norms.* Show that any norm $f : \mathbb{R}^m \to \mathbb{R}$ is a *convex function*, that is,

   $$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

   for any $x, y \in \mathbb{R}^m$ and any $\theta \in [0, 1]$. This means we can easily incorporate norms into objective functions we are optimizing.
   (b) *Sparsity norm?* For $x \in \mathbb{R}^m$, define $f(x)$ to be the number of non-zero entries of x. For instance, $f((1, 0, 2, 0, 0)) = 2$. Is this is a norm? Either show that it is, or explain why it isn't.
   (c) *Norms yield metrics.* Let $\| \cdot \|$ be any norm on $\mathbb{R}^m$. Show that $d(x, y) = \|x - y\|$ is a distance metric on $\mathbb{R}^m$.

3. *Total variation distance.* Let p and q be probability distributions on a set of m outcomes Ω; thus $p, q \in \Delta_m$. For any $S \subseteq \Omega$, we will write $p(S) = \sum_{x \in S} p(x)$ (and likewise $q(S)$). The *total variation distance* between p and q is defined to be

   $$\mathrm{TVD}(p, q) = \max_{S \subseteq \Omega} |p(S) - q(S)|.$$

(a) Suppose there are four outcomes, $\Omega = \{1, 2, 3, 4\}$ and we have $p = (1/2, 1/4, 1/8, 1/8)$ and $q = (1/8, 1/2, 1/8, 1/4)$. What is $\text{TVD}(p, q)$? For what set S is it realized?

(b) In the example from part (a), what is $\|p - q\|_1$?

(c) Show that in general (for finite outcome spaces) $\|p - q\|_1 = 2 \cdot \text{TVD}(p, q)$.

For more general (e.g., infinite) outcome spaces Ω, we define $\text{TVD}(p, q)$ to the supremum of $|p(S) - q(S)|$ over all measurable sets $S \subseteq \Omega$.

4. *A coupling inequality for total variation distance.* This is a key property of total variation distance. Let p, q be distributions over a space of outcomes Ω, and consider any pair of random variables (X, Y) where X (considered on its own) has distribution p and Y has distribution q. Then it can be shown that $\Pr(X \neq Y) \geq \text{TVD}(p, q)$.

In fact, $\text{TVD}(p, q)$ is the infimum of $\Pr(X \neq Y)$ over all pairs of random variables (X, Y) with $X \sim p$ and $Y \sim q$. More formally, define $\Gamma(p, q)$ to be the set of all *couplings* of p and q, that is, the set of all distributions ν over $\Omega \times \Omega$ such that the restriction of ν to the first coordinate is p and the restriction of ν to the second coordinate is q. Then

$$\text{TVD}(p, q) = \inf_{\nu \in \Gamma(p,q)} \Pr_{(X,Y)\sim\nu}(X \neq Y).$$

Let's see an example of this.

Suppose $\Omega = \{1, 2, 3, 4\}$, $p = (1/2, 1/4, 1/8, 1/8)$, and $q = (1/4, 1/8, 1/2, 1/8)$. Any coupling $\nu \in \Gamma(p, q)$ is a distribution over $\Omega \times \Omega$. Here's an example of such a distribution:

(x, y)	Prob
$(1, 1)$	$1/4$
$(1, 3)$	$1/4$
$(2, 2)$	$1/8$
$(2, 3)$	$1/8$
$(3, 3)$	$1/8$
$(4, 4)$	$1/8$

The following questions refer to this specific example.

(a) What is the distribution of X? What is the distribution of Y? And what is $\Pr(X \neq Y)$?

(b) What is $\text{TVD}(p, q)$ according to the original definition? For what set $S \subseteq \Omega$ is it realized?

5. *Wasserstein distance.* Let $(\mathcal{X}, d)$ be a metric space and let p and q be distributions over $\mathcal{X}$. We define the *Wasserstein* distance (or *earthmover distance*) between p and q to be

$$W_1(p, q) = \inf_{\nu \in \Gamma(p,q)} \mathbb{E}_{(X,Y)\sim\nu}[d(X, Y)],$$

where $\Gamma(p, q)$ is the set of all couplings of p and q, as described earlier.

We'll now do a small example. Let $\mathcal{X} = \mathbb{R}^2$ and let d be Euclidean distance. Suppose p and q are defined as follows:

- p puts probability mass $1/4$ at each of the points $(0, 0), (1, 0), (0, 1), (1, 1)$.
- q puts probability mass $1/8$ at each of the points $(0, 0), (1, 0), (0, 1), (1, 1)$ and probability mass $1/4$ at $(0.5, 0)$ and $(0.5, 1)$.

What is $W_1(p, q)$?

6. *Experiments with distance concentration.* High-dimensional spaces are full of strange effects. One of these is *distance concentration*, which we'll explore in this problem.

   Suppose we draw n points $x_1, \ldots, x_n$ uniformly at random from the unit sphere in d dimensions, that is, $S^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ (the superscript is $d-1$ because this is a $(d-1)$-dimensional manifold in $\mathbb{R}^d$). When d is small, say 1 or 2 or 3, the distances between the points will take on a fairly broad range of values in $[0, 2]$. But as d grows, the interpoint distances get more concentrated. We will investigate this phenomenon experimentally.

   First off, how does one generate a random point $X \in S^{d-1}$? Here's an easy way to do it:

   - Let $Z_1, \ldots, Z_d$ each be chosen from a standard normal distribution (a Gaussian with mean zero and variance 1).
   - Define $Z = (Z_1, \ldots, Z_d)$ and $X = Z/\|Z\|$.

   (a) Using the procedure above, write a function that returns n points chosen at random from S^{d-1}, given d and n. (In Python, you can sample from the standard normal using `numpy.random.normal`.) For $d = 2, 5, 10, 20, 100$ and $n = 100$:

   - Generate these samples, compute all $\binom{n}{2}$ interpoint distances, and plot a histogram of these values. There should be a separate histogram for each choice of d. In each case, allot 20 bins for the histogram and have the horizontal axis run from 0 (minimum possible distance) to 2 (maximum possible distance).

   (b) In your histograms from part (a), you should see the distances concentrating around a particular value $v \in [0, 2]$ as d grows. What do you think this value is?

   (c) Now focus on a particular choice of d, say $d = 1000$. In $\mathbb{R}^d$, there can be at most $d + 1$ points that are *exactly* the same distance from each other. But there can be $2^{O(d)}$ points that are *approximately* the same distance from each other. To get a taste of this, try out the following procedure:

   - Pick $x^{(1)}$ at random from S^{d-1}
   - For $i = 2, 3, 4, \ldots, 10000$:
     - Generate $x^{(i)}$ at random from S^{d-1}
     - Compute distances from $x^{(i)}$ to $x^{(1)}, \ldots, x^{(i-1)}$. Let u_i be the largest of these distances and s_i the smallest distance.

   (If your computer is sluggish, you might need to limit i to 5000 rather than 10000.) In a single plot, show both the u_i and s_i values for $i > 1$. Set the vertical axis to stretch from 0 (minimum possible distance) to 2 (maximum possible distance).

7. *Experiments with using k-d trees for (comprehensive) proximity search.* As we discussed in class, one very bad scenario for nearest neighbor data structures is when the data are uniformly distributed over a unit sphere in $\mathbb{R}^d$; in even moderate dimension d, such data points are likely to be almost equidistant from each other.

   In this problem, we will start by seeing how k-d trees perform on data of this kind, for various values of d. We will then use k-d trees to answer nearest neighbor queries on MNIST data, and use this to get a sense of the *intrinsic dimension* of MNIST.

   (a) For dimensions $d = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$, do the following:

- Generate 60,000 training points uniformly at random from the unit sphere in $\mathbb{R}^d$.
- Generate 100 test points uniformly at random from the unit sphere in $\mathbb{R}^d$.
- Build a k-d tree on the training points using `sklearn.neighbors.KDTree`. Specify `leaf_size=2`.
- Use this tree to compute the nearest neighbors of the 100 test points. Keep track of the total number of distance computations performed, using `reset_n_calls()` and `get_n_calls()`. In this way, obtain the average number of distance computations per query; this will be a value in the range 1 to 60,000.

Plot the average number of distance computations per query against d.

(b) Load in the MNIST data set. Build a k-d tree on the 60,000 training points, again using `leaf_size = 2`. Now use this tree to find the nearest neighbor of the first 100 test points. What is the average number of distance computations per query?

(c) By comparing your answers to (a) and (b), give a rough answer to the following question: for what value of d does k-d tree performance on MNIST behave like k-d tree on uniform-random data in $\mathbb{R}^d$? We can think of this as one measure of the *intrinsic dimension* of MNIST.