CSE 291: Unsupervised learning

Homework 2

This homework is due on Thursday April 27 at 11.59pm.

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

1. *Distribution of mass in a high-dimensional ball.* Suppose a point X is drawn uniformly at random from the d-dimensional ball $B = \{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$.

   (a) Show that X is overwhelmingly likely to be very close to the surface of the ball, by obtaining an expression for $\Pr(\|X\|_2 \leq 1 - \epsilon)$. *Hint:* In $\mathbb{R}^d$, what is the ratio of the volume of $B(0, 1-\epsilon)$ to the volume of $B(0,1)$, where $B(z,r)$ denotes the ball of radius r centered at z?

   (b) Suppose $d \geq 3$ and we pick polynomially many, say d^{100}, points uniformly at random from the ball. Show that there is at least a 99% probability that *all* these points will have length $> 1-(c \ln d)/d$, for some suitable constant c.

2. *Correlation between linearly related variables.* Let X be any random variable that takes values in $\mathbb{R}$, and let $Y = aX + b$ for some constants a, b.

   (a) Give a formula for the covariance between X and Y, in terms of the variance of X.

   (b) What is the correlation between X and Y?

3. *Do deterministic relationships imply correlation?* Suppose $X \in \{-1, 0, 1\}$ takes each value with probability exactly $1/3$. Specify a function f on $\{-1, 0, 1\}$ such that $Y = f(X)$ is uncorrelated with X.

4. *Online computation of the variance.* Demonstrate an online algorithm that, given an infinite sequence of inputs $x_1, x_2, \ldots$, always maintains the variance $v_t = \text{var}(x_1, \ldots, x_t)$ of the inputs so far.

5. *Data-dependent sampling in an online setting.* Consider a setting in which an infinite stream of data $x_1, x_2, \ldots \in \mathcal{X}$ is being received. Show how to maintain a single random sample $z_t \in \{x_1, \ldots, x_t\}$ such that

   $$\Pr(z_t = x_j) \propto f(x_j) \quad \text{for any } 1 \leq j \leq t.$$

   Here $f : \mathcal{X} \to \mathbb{R}^+$ is an arbitrary function that is known in advance, e.g. $f(x) = \|x\|^2$. *Hint:* The probability of having x_j at time t is

   $$\frac{f(x_j)}{f(x_1) + \cdots + f(x_t)}.$$

   You should maintain this denominator at all times.

6. *Suboptimality of Lloyd's algorithm.* Consider the following data set consisting of five points in $\mathbb{R}^1$:

   $$-10, -8, 0, 8, 10.$$

   We would like to cluster these points into $k = 3$ groups.

(a) What is the optimal k-means solution? Give the locations of the centers as well as the k-means cost.

(b) Suppose we call Lloyd's k-means algorithm on this data, with $k = 3$ and with initialization $\mu_1 = -10, \mu_2 = -8, \mu_3 = 0$. What is the final set of cluster centers obtained by the algorithm? What is the k-means cost of this set of centers?

7. *The k-center cost function.* An alternative to k-means is the *k-center* clustering problem, defined as follows:

- Input: Data points $x_1, \ldots, x_n$ in some metric space $(\mathcal{X}, d)$; integer $k > 0$
- Output: "Centers" $\mu_1, \ldots, \mu_k \in \mathcal{X}$
- Goal: Minimize
$$\max_i \min_{1 \le j \le k} d(x_i, \mu_j).$$

Whereas k-means minimizes the *total* (squared) distance from datapoints to their closest centers, this cost function minimizes the *largest* distance between a datapoint and its closest center.

The k-center cost function is NP-hard to optimize. However, there is a good heuristic for it known as *farthest-first traversal*:

- Set μ_1 to be any of the data points
- Repeat for $j = 2, 3, \ldots, k$:
  - Set μ_j to be the data point farthest from $\mu_1, \ldots, \mu_{j-1}$

(a) Consider the following data set in $\mathcal{X} = \mathbb{R}^2$:

$$(1,1), (1,2), (1,8), (2,1), (2,2), (5,1), (6,1)$$

What is the optimal k-center solution for this data set (using ℓ_2 distance), for $k = 3$? Remember that the centers can be arbitrary points in $\mathcal{X}$. Give the optimal set of centers as well as the cost.

(b) Suppose we run farthest-first traversal on this data set, starting with $\mu_1 = (1,1)$. Which centers will it pick as μ_2 and μ_3? What is the k-center cost of this solution?

(c) It turns out that farthest-first traversal is guaranteed to return a solution whose k-center cost is at most twice the cost of the best solution. Briefly explain why we cannot hope for a better ratio from any algorithm that returns data points as centers.

8. For this problem, we'll be using the *animals with attributes* data set. Go to

`http://attributes.kyb.tuebingen.mpg.de`

and, under "Downloads", choose the "base package" (the very first file in the list). Unzip it and look over the various text files.

This is a small data set that has information about 50 animals. The animals are listed in `classes.txt`. For each animal, the information consists of values for 85 features: does the animal have a tail, is it slow, does it have tusks, etc. The details of the features are in `predicates.txt`. The full data consists of a 50×85 matrix of real values, in `predicate-matrix-continuous.txt`. There is also a binarized version of this data, in `predicate-matrix-binary.txt`.

Load the real-valued array, and also the animal names, into Python. Now hierarchically cluster this data, using `scipy.cluster.hierarchy.linkage`. Choose Ward's method, and plot the resulting tree

using the `dendrogram` method, setting the `orientation` parameter to `'right'` and labeling each leaf with the corresponding animal name.

You will run into a problem: the plot is too cramped because the default figure size is so small. To make it larger, preface your code with the following:

```
from pylab import rcParams
rcParams['figure.figsize'] = 5, 10
```

(or try a different size if this doesn't seem quite right).

(a) Show the dendrogram that you get.

(b) Ward's method of average linkage is essentially trying to minimize the k-means cost function. Let's see how well it does. Take $k = 10$ in what follows:

- Show the k-clustering returned by Ward's method. What is its cost?
- Run k-means on this data, 10 times (each time initializing with 10 centers chosen at random from the data). Pick out the best (lowest cost) solution and show it. What is its cost?