DSC 257R: Unsupervised learning

Homework 3

Online unsupervised learning

1. *Online computation of the mean.* The course webpage contains a file `data.txt` that has a list of 5000 integers. Load it into memory (e.g., using `numpy.loadtxt`) for this problem.

   (a) Implement the online algorithm for computing the mean that was presented in lecture. Run it on the data set and print the final mean obtained. Is this identical to the true mean of the data?

   (b) Plot all (5000) intermediate values of the mean along the way.

2. *Online computation of the variance.* Recall that the variance of a random variable X is $\text{var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$. We can compute the variance of data in an online setting in much the same way as we did the mean.

   (a) Give pseudocode for 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.

   (b) Implement this algorithm, run it on `data.txt`, and compare the final result to the true variance of the data.

3. *Online approximation of the median.*

   (a) What is the true median of the numbers in `data.txt`?

   (b) Implement the algorithm from lecture for maintaining a random sample with replacement. We can use a random sample of size 100 to estimate the median of the numbers in `data.txt`. Repeat this 10 times and print the estimate obtained each time.

   (c) Now do (b) again, but using random samples of size 500.

4. In a stream of items $x_1, x_2, \ldots, x_m$, an ϵ-heavy hitter is a item that occurs at least ϵm times. For $\epsilon = 0.05$, at most how many ϵ-heavy hitters can there be?

Basics of clustering

5. *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?

6. For this problem, we'll be using the *animals with attributes* data set. You can find information about it at:

   `https://cvml.ista.ac.at/AwA/`

   From the course website, download the `awa.zip` file. 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. (You can reconstruct this by processing the matrix returned by `scipy.cluster.hierarchy.linkage`, which spells out the sequence of mergers that produced the hierarchy.) What is its k-means 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?

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 $\mathcal{X} = \mathbb{R}^2$:

$$\begin{bmatrix}1\\1\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}1\\8\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}5\\1\end{bmatrix}, \begin{bmatrix}6\\1\end{bmatrix}$$

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 $\mu_1 = \begin{bmatrix}1\\1\end{bmatrix}$. Which centers will it pick as μ_2 and μ_3? What is the k-center cost of this solution?

Interestingly, farthest-first traversal is guaranteed to return a solution whose k-center cost is at most twice the cost of the best solution.