10-401 Machine Learning: Homework 5

Due 5:30 p.m. Friday, May 4, 2018

Instructions

- **Submission:** You must submit your solutions on time electronically by submitting to autolab by 5:30 p.m. Friday, May 4, 2018. On the Homework 5 autolab page, you can click on the "download handout" link to download the submission template, which is a tar archive containing a Octave `.m` file for each programming question. Replace each of these files with your solutions for the corresponding problem, create a new tar archive of the top-level directory, and submit your archived solutions online by clicking the "Submit File" button.

  DO NOT change the name of any of the files or folders in the submission template. In other words, your submitted files should have exactly the same names as those in the submission template. Do not modify the directory structure.

- **Grading**: This homework is extra credit. There are 20 total points available on this homework. All of the points on this homework will be applied to your lowest, non-dropped homework.

- **Late homework policy**: No late submissions will be accepted and no credit will be given after the deadline since this is an extra-credit homework.

- **Collaboration policy**: You are welcome to collaborate on any of the questions with anybody you like. However, you must write up your own final solution, and you must list the names of anybody you collaborated with on this assignment.

1 Implementing k-means Clustering [12 pts]

In this problem you will implement Lloyd's method for the k-means clustering problem and answer several questions about the k-means objective, Lloyd's method, and k-means++.

Recall that given a set $S = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ of n points in d-dimensional space, the goal of k-means clustering is to find a set of centers $c_1, \ldots, c_k \in \mathbb{R}^d$ that minimize the k-means objective:

$$\sum_{j=1}^{n} \min_{i \in \{1,\ldots,k\}} \|x_j - c_i\|^2, \tag{1}$$

which measures the sum of squared distances from each point x_j to its nearest center.

In class we discussed that finding the optimal centers for the k-means objective is NP-hard, which means that there is likely no algorithm that can efficiently compute the optimal centers. Instead, we often use Lloyd's method, which is a heuristic algorithm for minimizing the k-means objective that is efficient in practice and often outputs reasonably good clusterings. Lloyd's method maintains a set of centers $c_1, \ldots, c_k$ and a partitioning of the data S into k clusters, $C_1, \ldots, C_k$. The algorithm alternates between two steps: (i) improving the partitioning $C_1, \ldots, C_k$ by reassigning each point to the cluster with the nearest center, and (ii) improving the centers $c_1, \ldots, c_k$ by setting c_i to be the mean of those points in the set C_i for $i = 1, \ldots, k$. Typically, these two steps are repeated until the clustering converges (i.e, the partitioning $C_1, \ldots, C_k$ remains unchanged after an update). Pseudocode is given below:

1. Initialize the centers $c_1, \ldots, c_k$ and the partition $C_1, \ldots, C_k$ arbitrarily.
2. Do the following until the partitioning $C_1, \ldots, C_k$ does not change:
   i. For each cluster index i, let $C_i = \{x \in S : x \text{ is closer to } c_i \text{ than any other center}\}$, breaking ties arbitrarily but consistently.
   ii. For each cluster index i, let $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$.

Implementing Lloyd's Method

In the remainder of this problem you will implement and experiment with Lloyd's method and the k-means++ algorithm on the two dimensional dataset shown in Figure 1.

(b) [3 pts] Complete the function `[a] = update_assignments(X, C, a)`. The input `X` is the $n \times d$ data matrix, `C` is the $k \times d$ matrix of current centers, and `a` is the $n \times 1$ vector of current cluster assignments. That is, `C(i,:)` is the center for cluster i and the j^{th} data point, `X(j,:)`, is assigned to cluster `a(j)`. Your function should output a new $n \times 1$ vector of cluster assignments so that each point is assigned to the cluster with the nearest center.

(c) [3 pts] Complete the function `[C] = update_centers(X, C, a)`. The input arguments are as in part (b). Your function should output a $k \times d$ matrix `C` whose i^{th} row is the optimal center for those points in cluster i.

(d) [3 pts] Complete the function `[C, a] = lloyd_iteration(X, C)`. This function takes a data matrix `X`, initial centers `C` and runs Lloyd's method until convergence. Specifically, alternate between updating the assignments and updating the centers until the the assignments stop changing. Your function should output the final $k \times d$ matrix `C` of centers and final the $n \times 1$ vector `a` of assignments.

(e) [3 pts] Complete the function `[obj] = kmeans_obj(X, C, a)`. This function takes the $n \times d$ data matrix `X`, a $k \times d$ matrix `C` of centers, and a $n \times 1$ vector `a` of cluster assignments. Your function should output the value of the k-means objective of the provided clustering.

We provide you with a function `[C, a, obj] = kmeans_cluster(X, k, init, num_restarts)` that takes an $n \times d$ data matrix `X`, the number `k` of clusters, a string `init` which must be either `'random'` or `'kmeans++'`, and a number of restarts `num_restarts`. This function runs Lloyd's method `num_restarts` times and outputs the best clustering it finds.

We encourage you to try out this function on the provided dataset and visualize clusters to learn more about k means.

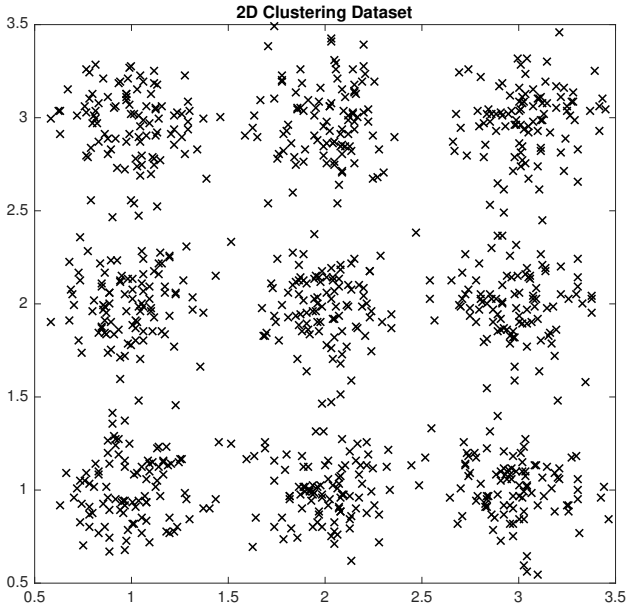

Figure 1: The 2D dataset used for this problem

2 Theory

2.1 k-means on the real line [4 pts]

In this problem you will show that the k-means objective (1) can be minimized in polynomial time when the data points are single dimensional ($d = 1$), despite the fact that in general finding the optimal centers is NP-hard.

(a) [1 pts] Consider the case where $k = 3$ and we have 4 data points $x_1 = 1$, $x_2 = 3$, $x_3 = 6$, $x_4 = 7$. What is the optimal clustering for this data? What is the corresponding value of the objective (1).

(b) [1 pts] One might be tempted to think that Lloyd's method is guaranteed to converge to the global minimum when $d = 1$. Show that there exists a suboptimal cluster assignment for the data in part (a) that Lloyd's algorithm will not be able to improve (to get full credit, you need to show the assignment, show why it is suboptimal *and* explain why it will not be improved).

(c) [1 pts] Assume we sort our data points such that $x_1 \leq x_2 \leq \cdots \leq x_n$. Prove that an optimal cluster assignment has the property that each cluster corresponds to some interval of points. That is, for each cluster j, there exits i_1 and i_2 such that the cluster consists of $\{x_{i_1}, x_{i_1+1}, \ldots, x_{i_2}\}$.

(d) [1 pts] Develop an $O(kn^2)$ dynamic programming algorithm for single dimensional k-means. (Hint: from part (c), what we need to optimize are $k-1$ cluster boundaries where the i^{th} boundary marks the largest point in the i^{th} cluster.)

2.2 PCA: Maximizing the variance [2 pts]

Consider N data points $X_1, ..., X_N \in \mathbb{R}^p$ such that the sample mean of X is zero: $\frac{1}{N}\sum_{i=1}^{N} X_i = \mathbf{0}$. Also consider projection vector $\mathbf{u} \in \mathbb{R}^p$, where $||\mathbf{u}||_2 = 1$. We would like to maximize the sample variance $\tilde{V}[\mathbf{u}^T X]$, which is the sample variance of X projected onto $\mathbf{u}$. The sample variance of N samples of a variable z is

$\tilde{V}[z] = \frac{1}{N}\sum_{i=1}^{N} z_i^2$, where $\frac{1}{N}\sum_{i=1}^{N} z_i = 0$. We can write an optimization problem to maximize the sample variance:

$$\max_{\substack{\mathbf{u} \\ s.t. ||\mathbf{u}||_2=1}} \tilde{V}[\mathbf{u}^T X] \tag{2}$$

Reformulate Eq.2 to the following optimization problem, and define the *pxp* matrix Σ:

$$\max_{\substack{\mathbf{u} \\ s.t. ||\mathbf{u}||_2=1}} \mathbf{u}^T \Sigma \mathbf{u} \tag{3}$$

2.3 PCA: Minimizing the reconstruction error [2 pts]

Consider the same variables as those defined in the previous question. Instead of maximizing the projected variance, we now seek to minimize the reconstruction error. In other words, we would like to minimize the difference between X and the reconstructed $\mathbf{u}\mathbf{u}^T X$. Note that $\mathbf{u}\mathbf{u}^T X$ first projects X onto $\mathbf{u}$, and then projects this scalar back into the p-dimensional space along the $\mathbf{u}$ axis. Minimizing the reconstruction error can be written as the following optimization problem.

$$\min_{\substack{\mathbf{u} \\ s.t. ||\mathbf{u}||_2=1}} \frac{1}{N}\sum_{i=1}^{N} ||X_i - \mathbf{u}\mathbf{u}^T X_i||_2^2 \tag{4}$$

Reformulate Eq. 4 to the following optimization problem with the same Σ as in the previous question.

$$\max_{\substack{\mathbf{u} \\ s.t. ||\mathbf{u}||_2=1}} \mathbf{u}^T \Sigma \mathbf{u} \tag{5}$$