ECE M146 Homework 7 Solution
Introduction to Machine Learning
Instructor: Lara Dolecek
TA: Zehui (Alex) Chen, Ruiyi (John) Wu

Please upload your homework to Gradescope by May 28, 11:59 pm.
Please submit a single PDF directly on Gradescope
You may type your homework or scan your handwritten version. Make sure all the work is discernible.

1. Suppose we have a data set $\{x_1, \cdots, x_N\}$ and out goal is to partition the data set in to K clusters with μ_k representing the center of the k-th cluster. Recall that in K-means clustering we are attempting to minimize an objective function defined as follows:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|_2^2,$$

where $r_{nk} \in \{0, 1\}$ and $r_{nk} = 1$ only if x_n is assigned to cluster k.

(a) What is the minimum value of the objective function when $K = n$ (the number of clusters equals to the number of samples)?
Solution: The minimum is 0 by assigning each x_i an unique cluster with the center also being x_i.

(b) Adding a regularization term, the objective function now becomes:

$$J = \sum_{k=1}^{K} \left[\lambda \|\mu_k\|_2^2 + \sum_{n=1}^{N} r_{nk} \|x_n - \mu_k\|_2^2 \right].$$

Consider the optimization of μ_k with all r_{nk} known. Find the optimal μ_k for

$$\text{argmin}_{\mu_k} \lambda \|\mu_k\|_2^2 + \sum_{n=1}^{N} r_{nk} \|x_n - \mu_k\|_2^2.$$

Discuss your answer. How would the regularization affect each step of the K-means clustering algorithm?
Solution: Let $f(\mu_k) = \lambda \|\mu_k\|_2^2 + \sum_{n=1}^{N} r_{nk} \|x_n - \mu_k\|_2^2$. Taking the gradient with respect to μ_k, we have:

$$\nabla f(\mu_k) = 2\lambda \mu_k - 2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k).$$

Letting the gradient to be 0, we get:

$$\mu_k^* = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\lambda + \sum_{n=1}^{N} r_{nk}}.$$

With the regularization in the denominator, the center for each cluster will tend to be close to the origin.

2. We have unlabeled data $x_n \in \mathbf{R}^M, n = 1, \cdots, N$. Suppose we want to use L_1 distance instead of L_2 distance to cluster the data into K clusters. The objective function we are minimizing becomes:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|_1,$$

where $\|z\|_1 = \sum_{i=1}^{M} |z_i|$ for $z \in \mathbf{R}^M$. The parameters $r_{nk} \in \{0, 1\}$ and $r_{nk} = 1$ only if x_n is assigned to cluster k.
In the maximization step, with r_{nk} fixed, define $C_k = \{n | r_{nk} = 1\}$ for the k-th cluster. Then we need to find μ_k that minimizes the following function:

$$f(\mu_k) = \sum_{n \in C_k} \|x_n - \mu_k\|_1. \tag{1}$$

Define x_{ni} to be the i-th element in x_n and μ_{ki} to be the i-th element in μ_k. We can expand (1) as following:

$$f(\mu_k) = \sum_{n \in C_k} \|x_n - \mu_k\|_1 = \sum_{n \in C_k} \sum_{i=1}^{M} |x_{ni} - \mu_{ki}| = \sum_{i=1}^{M} \sum_{n \in C_k} |x_{ni} - \mu_{ki}|.$$

The above expansion shows that we can optimize for each element in μ_k separately.

(a) Consider first the problem of finding $\bar{y}^*$ that minimizes $f(\bar{y}) = \sum_{j=1}^{N_k} |y_j - \bar{y}|$ for $y_j \in \mathbf{R}$. Because $f(\bar{y})$ is not differentiable everywhere, we need the notion of *subgradient*. We say $g \in \mathbf{R}$ is a subgradient of f at $x \in \mathbf{dom} f$ of for all $z \in \mathbf{dom} f$:

$$f(z) \geq f(x) + g(z - x).$$

The subgradient of f at point x where f is differentiable equals to the derivative of f at x. A function f is called subdifferentiable at x if there exists at least one subgradient at x. The set of subgradient of f at point x is called *subdifferential* of f at y and is denoted as $\partial f(x)$. Find $\partial f(x)$ of $f(x) = |x|$ for $x < 0$, $x > 0$ and $x = 0$.
Solution:

$$\partial f(x) = \begin{cases} -1, x < 0 \\ [-1, 1], x = 0 \\ 1, x > 0 \end{cases}$$

For both $x < 0$ or $x > 0$, $f(x)$ is differentiable so $\partial f(x)$ is the corresponding derivative. At $x = 0$ the subdifferential is defined by the inequality $|z| \geq gz$ for all z, which is satisfied if and only if $-1 \leq g \leq 1$.

(b) For simplicity, we assume $y_1 < y_2 < \cdots < y_{N_k - 1} < y_{N_k}$ and N_k being odd. Also assume that $f(\bar{y})$ is convex and is subdifferentiable everywhere. Use the following theorem:
A point x^* is a minimizer of a convex function f if and only if f is subdifferentiable at x^* and $0 \in \partial f(x^*)$, i.e., $g = 0$ is a subgradient of f at x^*.

Show that the $\bar{y}^*$ that minimizes $\sum_{j=1}^{N_k} |y_j - \bar{y}|$ is the median of $\{y_1, \cdots, y_{N_k}\}$, i.e., $\text{argmax}_{\bar{y}} f(\bar{y}) = y_{\frac{N_k+1}{2}}$.

Solution: We first show that $\bar{y}^*$ must take value in y_j. If $\bar{y} \neq y_i$, then from (b), we have $\partial f(\bar{y}) = N_+ - N_-$ where N_+ is the number of instances that $y_j > \bar{y}$ and N_- is the number of instances that $y_i < \bar{y}$. Because M is odd, $\partial f(\bar{y}) \neq 0$ for $\bar{y} \neq y_j$. Note: you can figure our by yourself what happens if M is even.

We next show that $0 \in \partial f(\bar{y})$ for only $\bar{y} = y_{\frac{N+1}{2}}$. Using result from (b), we have:

$$\partial f(\bar{y}) = [-1, 1] + N_+ - N_-, \text{for } \bar{y} = y_j.$$

In order to have $0 \in \partial f(\bar{y})$, we need $N_+ = N_-$ which gives us $\text{argmax}_{\bar{y}} f(\bar{y}) = y_{\frac{N+1}{2}}$.

(c) Write a two-step algorithm similar to the K-means algorithm that minimizes J. Comment on the advantage of this algorithm compared to the K-means algorithm.

Solution:

- Initialize the μ_k for each cluster.
- Expectation step. Assign the n-th data point to the closest cluster center using L1 distance as distance metric. Formally, this can be expressed as

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|x_n - \mu_j\|_1 \\ 0 & \text{otherwise.} \end{cases}$$

- Maximization step. For the k-th cluster, revise the center with new μ_k with $\mu_{ki} = \text{median}(\{x_{ni} | n \in C_k\})$ for $i = 1, \cdots, M$.

This is the K-medians algorithm, it is more robust to outliers.

3. Consider the matrix

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 0 \end{bmatrix}.$$

(a) Find the eigenvalues and eigenvectors of A. Show your steps. Make sure to normalize your eigenvectors to have unit norm.

Solution: The characteristic polynomial of A is

$$\pi(\lambda) = \det(A - \lambda I) = \lambda^2 - 3\lambda - 4.$$

Equating $\pi(\lambda)$ with 0, we get $\lambda_1 = 4$ and $\lambda_2 = -1$. For both eigenvalues, we solve for $(A - \lambda I)v = 0$. For $\lambda_1 = 4$, we get $v_1 = [2/\sqrt{5}, 1/\sqrt{5}]^T$. For $\lambda_2 = -1$, we get $v_2 = [-1/\sqrt{5}, 2/\sqrt{5}]^T$.

(b) Find the eigenvalue decomposition of A using the eigenvalues and eigenvectors you found. Hint: A is a symmetric matrix.

Solution: The eigenvectors for symmetric matrix are orthonormal to each other. We therefore have:

$$A = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T = 4 \times \begin{bmatrix} \frac{4}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{1}{5} \end{bmatrix} - 1 \times \begin{bmatrix} \frac{1}{4} & -\frac{2}{5} \\ -\frac{2}{5} & \frac{4}{5} \end{bmatrix}.$$

4. Answer the following questions regarding positive (semi-)definite matrix. A symmetric real matrix M is said to be positive definite if the scalar $z^T M z$ is positive for every non-zero column vector z.

   (a) Consider the matrix

   $$A = \begin{bmatrix} 9 & 6 \\ 6 & a \end{bmatrix}.$$

   What should a satisfy so that the matrix A is positive definite?
   Solution:

   $$z^T A z = 9z_1^2 + 12z_1z_2 + az_2^2 = (3z_1 + 2z_2)^2 + (a-4)z_2^2.$$

   It is then clear that $z^T A z > 0$ for every $z \neq 0$ if $a > 4$.

   (b) Suppose we know matrix B is positive definite. Show that B^{-1} is also positive definite. Hint: use the definition and the fact that every positive definite matrix is non-singular (invertible).
   Solution: Since B is positive definite, $z^T B z > 0$ for every $z \neq 0$. Rewrite $z^T B z > 0, \forall z \neq 0$. We get

   $$z^T B B^{-1} B z > 0, \forall z \neq 0.$$

   Let $y = Bz$, because B is non-singular, $y = 0 \iff z = 0$. Also because B is non-singular, we can find the corresponding z for any y using $z = B^{-1}y$. We therefore get:

   $$y^T B^{-1} y > 0, \forall y \neq 0.$$

   The above shows that B^{-1} is also positive definite.

   (c) Show that the data covariance matrix S in PCA is positive semi-definite.
   Solution: By definition,

   $$S = \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})(x_n - \bar{x})^T.$$

   It is easy to show that

   $$\forall z \neq 0, z^T S z = \frac{1}{N} \sum_{n=1}^{N} z(x_n - \bar{x})(x_n - \bar{x})^T z = \frac{1}{N} \sum_{n=1}^{N} ((x_n - \bar{x})^T z)^2 \geq 0$$

5. One application of the K-means algorithm is image segmentation and image compression. The goal of image segmentation is to partition an image into regions that have relatively similar visual appearance. Each pixel in an image can be viewed as a point in a 3-dimensional space which contains the intensity of the 3 color red, green and blue. K-means algorithm can be used to cluster the points in the 3-dimensional space in to K clusters therefore achieve segmentation. After segmentation, compression is achieved by replacing each pixel with the {R,G,B} triplet given by μ_k, the center the cluster to which it is assigned.
In this exercise, you will implement the K-means algorithm to segment and compress the image *UCLA_Bruin.jpg*. Note: for submission, you may turn in the required images in black and white.

   (a) **Visualization.** The picture of the famous Bruin bear contains 300×400 pixels. Read the image into MATLAB using *imread* and show the image using *imshow*.

   (b) **K-means Algorithm with $K = 4$.** Implement the K-means algorithm using all of the following specifications:

      - Partition the pixels into $K = 4$ clusters.
      - To allow for a deterministic result, initialize the cluster centers using the *furthest-first* heuristic on page 180 of *A Course in Machine Learning*. The heuristic is sketched below:
         - Pick the first pixel of the image, whose {R,G,B} values are $[229, 249, 250]$, as the center for the first cluster, i.e., $\mu_1 = [229, 249, 250]$.
         - For $k = 2, \cdots, K$: find the example n^* that is as far as possible from all previously selected means. Namely, $n^* = \underset{n}{\text{argmax}} \min_{k'<k} \|x_n - \mu_{k'}\|^2$. Set $\mu_k = x_{n^*}$.
      - Run the K-means algorithm for 10 iterations. An iteration consists the following two steps:

– Step 1, assign each example to the cluster whose center is the closest.

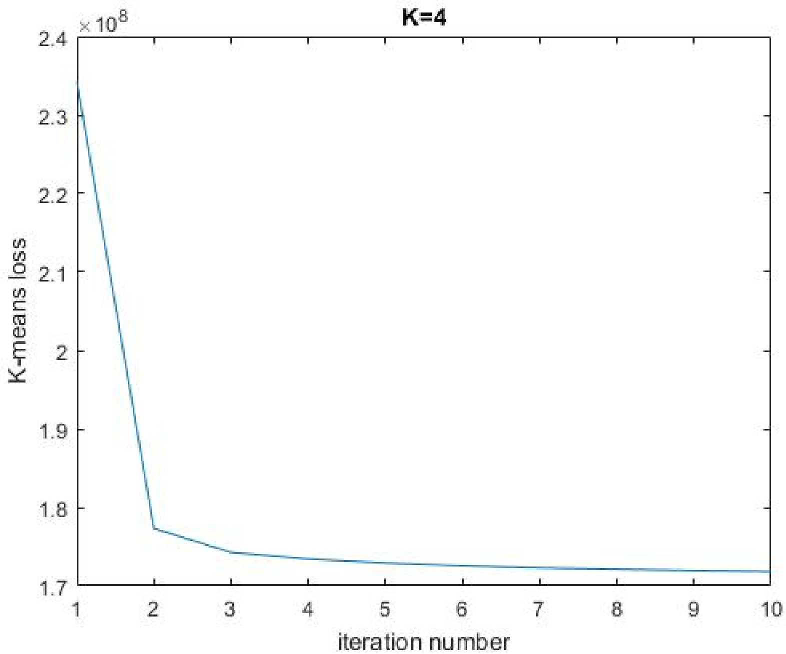
– Step 2, re-estimate the center of each cluster.

Calculate the K-means objective function:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|_2^2,$$

at the end of each iteration. The parameters $r_{nk} \in \{0, 1\}$ and $r_{nk} = 1$ only if x_n is assigned to cluster k. For this image, we have $x_n \in \mathbf{R}^3, i = 1, \cdots, N$, $N = 120000$. Generate a plot showing Js v.s. iterations. Comment on the convergence of the K-means algorithm.

Solution:



The K-means objective function decreases as expected for more iterations.

(c) **Compression with** $K = 4, 8$ **and** 16**.** For $K = 4, 8$ and 16, compress the *UCLA_Bruin* image using your K-means algorithm. For compression, replace the {R,G,B} values of each pixel with the center of the cluster to which it belongs. Use the same specifications in (b) and report the value of the objective function after your last iteration. Show your compressed image using *imshow*. Comment on how K affects the quality of the compressed image.

Solution:

K=4

K=8

The image quality is better for larger K. For $K = 4$, $J = 1.7183 \times 10^8$ (Any solution in the same order of magnitude is also correct). For $K = 8$, $J = 8.0434 \times 10^7$ (Any solution in the same order of magnitude is also correct). For $K = 16$, $J = 3.7430 \times 10^7$ (Any solution in the same order of magnitude is also correct). The objective function also becomes smaller for larger K.

(d) **Compression Ratio.** In the original image, each of the 300×400 pixels comprises {R,G,B} values each of which is stored with 8 bits of precision, i.e., $0 - 255$.How many bits do you need to store the original image?
Now you have compressed your image using the K-means algorithm. For each pixel, you store only the index of cluster to which it is assigned. You also need to store the value of the K centers with 8 bits of precision per color. How many bits do you need to store the compressed image with $K = 4, 8$ and 16? What are the compression ratios?
Solution: To store the original image, we need $24 \times 300 \times 400 = 2880000$ bits. For certain K, the number of bits we need are $24K + 300 \times 400 \log_2 K$. Therefore, we need $240096, 360192$ and 480384 bits for $K = 4, 8$ and 16, respectively. The compression ratios are 8.34%(K=4), 12.51%(K=8), and 16.68%(K=16), respectively.