COMPSCI 671D Fall 2020
Homework 4

Due 10:15 PM EST, November 11st

1 K-means and K-medians for Cute Raccoon

We can use k-means clustering to build a rudimentary image compression scheme. In this problem, we are going to "compress" a cute raccoon (Well, that's not real – the staff for this course are always friendly). The idea is that for each pixel in the image, instead of storing the actual floating point value, we just store a nonnegative integer, which acts as an index to a small set of representative pixel values called a **codebook**. Storing the codebook along with the index for each pixel requires many fewer bits than storing the floating point values for each pixel, which means we can store and transmit images much more efficiently. The process of mapping the floating point pixel values to a small set of nonnegative integers is called **quantization**.

1.1 K-means

How can we come up with a codebook and quantization that allows us to approximately recreate the original image? k-means provides one solution. The k group means will serve as the codebook, and each pixel value is replaced with its group membership. More precisely, let $I \in \mathbb{R}^{m \times n}$ be an image, let $\mu_1, ..., \mu_k \in \mathbb{R}$ be the k group means obtained by running the k-means algorithm on the mn pixels of image I, and let $C_{i,j}$ be the group membership of pixel $I_{i,j}$. We can approximately recover the original image from the compressed version by the image $\hat{I}$, which is defined by

$$\hat{I}_{i,j} = \mu_{C_{i,j}}. \tag{1}$$

(a) Implement the k-means algorithm in the language of your choice, using the squared distance (between the grayscale value of pixel i, j and the cluster center c_k) as the distance metric. The **arguments** of your function are k and the input dataset. **Return** both the group assignment for each point in the dataset, as well as the mean μ_j for each group/cluster. You can initialize each mean by randomly selecting a point from the input data. (See the Jupyter notebook for your reference).

(b) For $k = 2$, $k = 4$ and $k = 6$, perform this quantization technique on the image in the file `raccoon.png`. Plot the recovered image $\hat{I}$ in each case. Beyond what number of clusters do you visually not see any difference between the original image and the compressed image?

Figure 1: Raccoon Image

1.2 K-medians

An alternative possible solution to build an image compression scheme is to solve the following problem:

$$\min_{c_1,\ldots,c_K} \sum_{i,j=1}^{N} \min_{k\in\{1,\ldots,K\}} |x_{ij} - c_k|,$$

where $\{c_k\}_{k=1}^{K}$, $c_k \in \mathbb{R}$ are cluster centers, and $\{x_{ij}\}_{i,j=1}^{N=m\times p}$, $x_{i,j} \in \mathbb{R}$ represents the gray-scale value for pixel i, j.

(c) Using a similar derivation to k-means taught in class, derive the cluster assignment and cluster update steps that minimize the above objective function. Note that it may be slightly trickier than k-means. You might need to use that the derivative of the absolute value function is the sign function. Hint: the solution is partly given away in the title of this question.

(d) The ℓ^1 and ℓ^2 distances are the same in one dimension. Grayscale values are one dimensional. By comparing the two loss functions used in k-means and k-medians, and the updating process for the cluster centers, do you think the quantization will be different if use k-means versus using k-medians? Or will they always give the same result?

(e) For $k = 2$, perform the quantization technique on the same image file `raccoon.png` using the algorithm you derived above. The **arguments** of your function are k and the input dataset. **Return** both the group assignment for each point in the dataset, as well as the median ϕ_k for each group/cluster. Plot the recovered image $\hat{I}$. Do you get **exactly** the same image when using k-means and k-medians for $k = 2$? Does one seem qualitatively better than the other (if they are different)?

2 Ridge Regression

In this question, we will derive the Bayesian connection to Ridge regression.

(a) Suppose that $y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i$ where $\epsilon_1, \cdots, \epsilon_n$ are independent and identically distributed from a $N(0,1)$ distribution. Write out the likelihood for the data.

(b) Now assume the following prior for $\beta : \beta_1, \cdots, \beta_p$ are independent and identically distributed according to a normal distribution with mean zero and variance $1/\lambda$. Write out the posterior for one of the β's (for example, β_j where $1 \leq j \leq p$) in this setting.

(c) If we had not normalized the data, the generative process $y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i$ may not be realistic. Provide a case where it isn't realistic. To do this, you can provide the names of the variables and their typical ranges, and explain why the generative process is not realistic. Given that we should really normalize variables, what problem(s) could arise if we don't?

3 Neural Networks

Consider the neural network in Figure 2 with the weight parameters $\{w_j\}_{j=1}^m$ and $\{s_j\}_{j=1}^m$ for the hidden and output layers, respectively. The (normalized) output of this network can be written as

$$f(x) = \frac{1}{\sqrt{m}} \sum_{j=1}^{m} s_j \sigma(w_j^\top x), \tag{2}$$

where σ is the ReLU activation: $\sigma(x) = x\mathbb{I}_{[x \geq 0]}$.

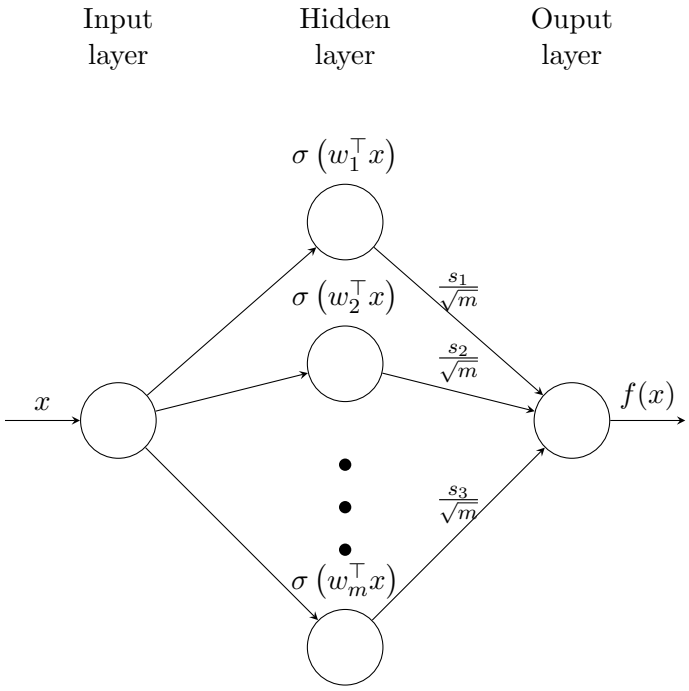


Figure 2

Consider the feature map represented by the hidden layer

$$\phi(x) := \begin{bmatrix} \frac{1}{\sqrt{m}} s_1 x^\top \mathbb{I}_{\left[w_1^\top x \geq 0\right]} \\ \vdots \\ \frac{1}{\sqrt{m}} s_m x^\top \mathbb{I}_{\left[w_m^\top x \geq 0\right]} \end{bmatrix}. \tag{3}$$

Here $\phi(x)$ is a matrix, with each row being $\frac{1}{\sqrt{m}} s_j x^\top \mathbb{I}_{\left[w_j^\top x \geq 0\right]}$. With this feature map, the model $f(x)$ can be written as $f(x) = tr\left(\phi\left(x\right) W\right)$, where $W = [w_1, w_2, \cdots, w_m]$ is the matrix of weights (columns are w_i).

Show that, if w_i follows a standard normal distribution ($w_i \overset{i.i.d.}{\sim} \mathcal{N}\left(0, I\right)$) and s_i follows the Rademacher distribution ($s_i \overset{i.i.d.}{\sim}$ Uniform$\{-1, +1\}$), then we have

$$\mathbb{E}_{w_1, \cdots, w_m, s_1, \cdots, s_m} \left\langle \phi(x_1), \phi(x_2) \right\rangle = x_1 x_2^\top \frac{\pi - \arccos\left(\frac{x_1^\top x_2}{\|x_1\| \|x_2\|_2}\right)}{2\pi}, \tag{4}$$

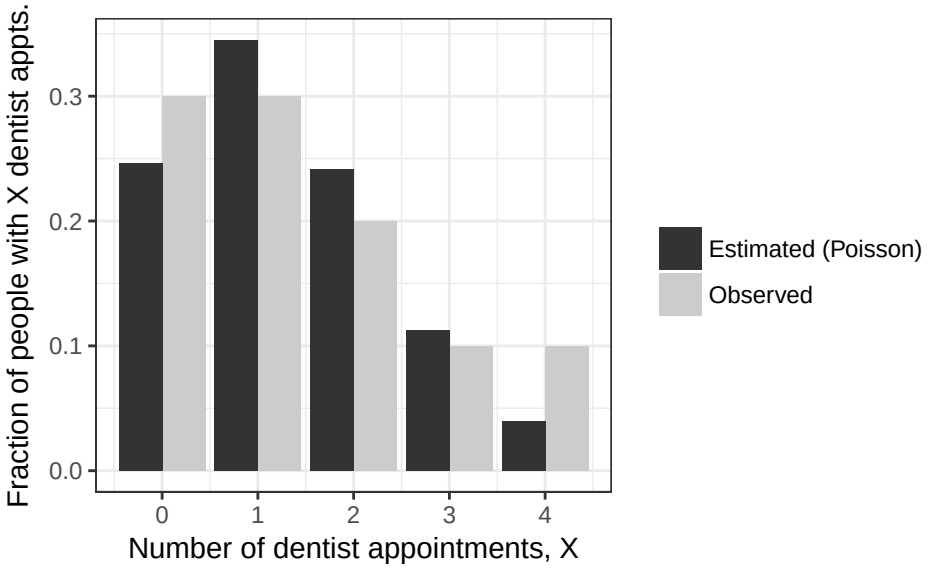
where the inner product of two matrices A and B is defined as $\langle A, B \rangle = tr(A^\top B)$.

Hint: $w_j^\top x \geq 0$ when the angle between w_j and x is smaller than $\frac{\pi}{2}$. For any two fixed vectors x_1 and x_2, and a random weight vector $w \sim \mathcal{N}\left(0, I\right)$, we have

$$\mathbb{P}_w \left(\arccos \frac{w^\top x_1}{\|w\|_2 \|x_1\|_2} \leq \frac{\pi}{2} \quad \text{and} \quad \arccos \frac{w^\top x_2}{\|w\|_2 \|x_2\|_2} \leq \frac{\pi}{2} \right) = \frac{\pi - \arccos \frac{x_1^\top x_2}{\|x_1\|_2 \|x_2\|_2}}{2\pi}. \tag{5}$$

4 Expectation Maximization (EM) for a Mixture of Two Different Distributions

Figure 3



Suppose $N = 100$ people were surveyed about how many dentist appointments they made in the past year. One way to model this data is to assume the responses $\{x_i\}_{i=1}^{100}$ are drawn i.i.d. from

a Poisson distribution, so $p(X_i = x_i \mid \lambda) = \frac{e^{-\lambda}\lambda^{x_i}}{x_i!}$. Figure 3 shows a histogram of the observed number of appointments and the predicted number of appointments from maximizing the assumed likelihood.

(a) Derive the maximum likelihood estimator $\hat{\lambda}$ for λ as a function of the x_i's.

We will now try a new model. Suppose that each person i is one of two types, denoted by a latent variable Z_i:

- If $Z_i = 1$, then person i never goes to the dentist (with probability 1), so $p(X_i = x_i \mid Z_i = 1, \lambda) = \mathbb{I}_{[x_i=0]}$.
- If $Z_i = 2$, then $p(X_i = x_i \mid Z_i = 2, \lambda) = \text{Poisson}(\lambda)$, as before.

We model each person as a mixture of these two types, letting $w = p(Z_i = 1 \mid \lambda)$ and $1 - w = p(Z_i = 2 \mid \lambda)$ denote the mixture weights. In general, the presence of a latent variable like Z_i can make it difficult to maximize the likelihood. We use the EM algorithm as a remedy to this problem.

E-step

In this step we compute the probability of each type assignment for each person. That is, we compute $\gamma_{i,k} := p(Z_i = k \mid X_i = x_i, \lambda)$ for all people $i \in \{1, \ldots, N\}$ and all types $k \in \{1, 2\}$.

(b) Write a formula for $p(X_i = x_i \mid Z_i = k, \lambda)$, the likelihood of observing outcome x_i given that person i is of type $Z_i = k$.

(c) Write a formula for $p(X_i = x_i \mid \lambda)$, the likelihood of observing outcome x_i.

(d) Write a formula for the type assignments $\gamma_{i,k} := p(Z_i = k \mid X_i = x_i, \lambda)$.

M-step

In this step we maximize a lower bound of the log likelihood:

$$A(w, \lambda) = \sum_{i=1}^{N} \sum_{k=1}^{K} \gamma_{i,k} \log \frac{p(X_i = x_i, Z_i = k \mid \lambda)}{\gamma_{i,k}}$$

over w and λ. Note that in this step the type assignments $\gamma_{i,k}$ are fixed.

(e) Maximize $A(w, \lambda)$ in λ. How does this compare to the maximum likelihood estimate when there was no latent variable (derived in part (a))?

(f) Maximize $A(w, \lambda)$ in w. How does this compare to the mixture of Gaussian distributions case, as derived in class?