STAT 161/261: Homework 3 Solutions
Due Wednesday, May 11 in class at 4pm.

1. Problem 3, parts (c) and (d) from Homework 2.

   (c) (8 points) The tricky part here is to generate the data. The likelihoods are

$$p(x|y=0) = \frac{1}{\lambda_0} e^{-x/\lambda_0},$$
$$p(x|y=1) = \frac{1-q}{\lambda_1} e^{-x/\lambda_1} + \frac{q}{\lambda_2} e^{-x/\lambda_2}.$$

   The second likelihood (which applies when $y = 1$) is equivalent to the following mixture model: with probability q, x is exponential with mean λ_2; and with probability $1 - q$, x is exponential with mean λ_1. Overall, since $P(y = 0) = P(y = 1) = \frac{1}{2}$, we have the following:

$$\begin{aligned} \text{With prob } 1/2: &\quad x \sim \text{Exp}(\lambda_0) \text{ and } y = 0 \\ \text{With prob } (1-q)/2: &\quad x \sim \text{Exp}(\lambda_1) \text{ and } y = 1 \\ \text{With prob } q/2: &\quad x \sim \text{Exp}(\lambda_2) \text{ and } y = 1 \end{aligned}$$

   To generate this data, we use the following code. We first create the following variables:

```
lamVal = [1 10 100]';      % lambda values
plam = [0.5 0.45 0.05]';   % P(lambda)
ntot = 1000;
```

   The vector `lamVal` has the three possible λ values and `plam` has the probabilities of the each of these λ values. Then, we run the following code:

```
q = [0; cumsum(plam,1)];
u = rand(ntot,1);
[~,ind] = histc(u,q);
lam = lamVal(ind);
xtot = exprnd(lam,ntot,1);
ytot = (ind > 1);
```

   The purpose of the first four lines is to create a vector `lam` of length `ntot` with the λ values of random x samples. There are many ways to do this. Here, `q` becomes a vector of bin edges to apply to the vector `u` of samples drawn from the uniform($[0, 1]$) distribution. Then, the `histc` command sets `ind(i)=j` when `u(i)` is in the interval `[q(j),q(j+1)]`.

This occurs with probability `plam`. The command `lam = lamVal(ind)` then looks up the correct λ value. Finally, the `exprnd` generates the exponential random variables.

The remainder of the code in the MATLAB file is identical to the earlier part. We see that the error increases and the model is not robust to outliers.

(d) (5 points) Again, the code is in the MATLAB published file. One way to do this is to discretize the values t, and for each possible threshold t to compute $\widehat{y}$ and measure the classification error rate. The code provides an efficient way of doing this without looping by sorting the data.

2. (a) (5 points) The Gamma distribution is (you can look this up on Wikipedia)

$$p(\theta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\theta^{\alpha-1}e^{-\beta\theta} \propto \theta^{\alpha-1}e^{-\beta\theta},$$

where the proportional to sign ($\propto$) indicates a constant that does not depend on θ. Hence,

$$p(\theta|\mathbf{x}) \propto p(\mathbf{x}|\theta)p(\theta) \propto \theta^N \prod_{i=1}^N e^{-x_i\theta}\theta^{\alpha-1}e^{-\beta\theta} = \theta^{\alpha'-1}e^{-\beta'\theta},$$

where

$$\alpha' = \alpha + N, \quad \beta' = \beta + \sum_{i=1}^N x_i.$$

(b) (5 points) Let $\mu = \mathbb{E}(\theta)$, and $\tau = \text{var}(\theta)$ be the prior mean and variance. We know from the Wikipedia page on the Gamma distribution,

$$\mu = \frac{\alpha}{\beta}, \quad \tau = \frac{\alpha}{\beta^2} \quad \Rightarrow \quad \beta = \frac{\mu}{\tau}, \alpha = \frac{\mu^2}{\tau}.$$

Hence, the posterior mean is

$$\mathbb{E}(\theta|\mathbf{x}) = \frac{\alpha'}{\beta'} = \frac{\alpha+N}{\beta+N\overline{x}} = \frac{\mu^2+N\tau}{\mu+N\overline{x}\tau},$$

where $\overline{x} = (1/N)\sum_i x_i$ is the sample mean. The posterior variance is

$$\text{var}(\theta|\mathbf{x}) = \frac{\alpha'}{(\beta')^2} = \frac{\alpha+N}{(\beta+N\overline{x})^2} = \tau\frac{\mu^2+N\tau}{(\mu+N\overline{x}\tau)^2}.$$

3. (8 points) Take the points 0, 1, and x, with $x > 3$, and suppose we use 2 clusters. Initialize k-mediods to mediods 1 and 2, and initialize k-means to means 1 and 2.

Applying the k-mediods algorithm, we get clusters $\{0\}$ and $\{1, x\}$, and there was no reason for the second mediod to be changed from 1, so this is the final clustering.

Applying the k-means algorithm, we also first get clusters $\{0\}$ and $\{1, x\}$. The mean for the second cluster is $(1+x)/2$, which is larger than 2. Thus, when the clusters are updated, they change to $\{0, 1\}$ and $\{x\}$. This is the final clustering.

This example demonstrates that k-mediods and k-means can arrive at different clustering even with the same initialization.

4. (a) (4 points) The matrix

$$\mathbf{A} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} \\ 1 & x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} & x_{N3} \end{bmatrix}$$

places the model in the form

$$\mathbf{r} = \mathbf{A}\mathbf{w} + \epsilon,$$

where

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix} \quad \text{and} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix}.$$

(b) (4 points) $\widehat{\mathbf{w}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{r}$

(c) (4 points)

$$\begin{aligned} \mathbb{E}[\widehat{\mathbf{w}} \,|\, \mathbf{w}] &= \mathbb{E}[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{r} \,|\, \mathbf{w}] = \mathbb{E}[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{w} + \epsilon) \,|\, \mathbf{w}] \\ &= \mathbb{E}[\mathbf{w} + (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\epsilon \,|\, \mathbf{w}] = \mathbf{w}, \end{aligned}$$

where the last step uses linearity of the expectation and ϵ having mean zero. In words, this shows that the estimate $\widehat{\mathbf{w}}$ is unbiased.

$$\begin{aligned} \mathbb{E}[\widehat{\mathbf{w}}\widehat{\mathbf{w}}^T \,|\, \mathbf{w}] &= \mathbb{E}[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T(\mathbf{A}\mathbf{w} + \epsilon)(\mathbf{A}\mathbf{w} + \epsilon)^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T} \,|\, \mathbf{w}] \\ &\overset{(a)}{=} \mathbb{E}[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}\mathbf{w}\mathbf{w}^T\mathbf{A}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T} \,|\, \mathbf{w}] \\ &\quad + \mathbb{E}[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\epsilon\epsilon^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T} \,|\, \mathbf{w}] \\ &= \mathbf{w}\mathbf{w}^T + (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbb{E}[\epsilon\epsilon^T]\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T}, \end{aligned}$$

where in (a) we could drop cross terms because ϵ has mean zero. Thus, $\text{var}(\widehat{\mathbf{w}} \,|\, \mathbf{w}) = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbb{E}[\epsilon\epsilon^T]\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-T}$.

5. (4 points for covariance matrix, 4 points for showing $\mathbf{1}$ is an eigenvector, 4 points for some additional discussion) Let x_{ij}, $j = 1, \ldots, p$, be the components of $\mathbf{x}_i$. Then

$$\mathbb{E}(x_{ij}) = \frac{1}{p}\mathbb{E}(a)$$

because x_{ij} has probability $1/p$ of equaling a and probability $1 - 1/p$ of equaling 0. For two different components $j \neq k$ is

$$\mathbb{E}(x_{ij}x_{ik}) = 0, \quad j \neq k,$$

since at least one of the components must be zero. For $j = k$,

$$\mathbb{E}(x_{ij}x_{ik}) = \mathbb{E}(x_{ij}^2) = \frac{1}{p}\mathbb{E}(a^2).$$

Hence the correlation matrix is

$$C_{jk} = \mathbb{E}(x_{ij}x_{ik}) - \mathbb{E}(x_{ij})\mathbb{E}(x_{ik}) = \frac{1}{p}\mathbb{E}(a^2)\delta_{jk} - \frac{\mathbb{E}^2(a)}{p^2}.$$

Thus,

$$C_{jk} = \mathbb{E}(x_{ij}x_{ik}) - \mathbb{E}(x_{ij})\mathbb{E}(x_{ik}) = \lambda\delta_{jk} - \mu.$$

where

$$\lambda = \frac{1}{p}\mathbb{E}(a^2), \quad \mu = \frac{\mathbb{E}^2(a)}{p^2}.$$

In matrix form,

$$\mathbf{C} = \lambda\mathbf{I} - \mu\mathbf{1}\mathbf{1}^T,$$

where $\mathbf{1}$ is the all-ones vector.

Now, to understand the value of PCA, we need to compute the eigenvalues of $\mathbf{C}$. First, to check that $\mathbf{1}$ is an eigenvector as stated in the problem statement, compute

$$\begin{aligned}\mathbf{C}\mathbf{1} &= \left[\lambda - \mu\mathbf{1}\mathbf{1}^T\right]\mathbf{1} \\ &= \lambda\mathbf{1} - \mu\mathbf{1}p = (\lambda - p\mu)\mathbf{1}.\end{aligned}$$

Thus $\mathbf{1}$ is an eigenvector with associated eigenvalue

$$\lambda - p\mu = \frac{1}{p}\mathbb{E}(a^2) - \frac{1}{p}\mathbb{E}^2(a) = \frac{1}{p}\operatorname{var}(a).$$

For any vector $\mathbf{v}$ orthogonal to $\mathbf{1}$ (i.e. $\mathbf{1}^T\mathbf{v} = 0$),

$$\mathbf{C}\mathbf{v} = \left[\lambda - \mu\mathbf{1}\mathbf{1}^T\right]\mathbf{v} = \lambda\mathbf{v} - \mu\mathbf{1}\mathbf{1}^T\mathbf{v} = \lambda\mathbf{v}.$$

So, the remaining eigenvalues are λ (which is larger than $\lambda - p\mu$). Thus, the top $p-1$ eigenvalues are the same, equal to λ. PCA is a bad way to select features because the top $p-1$ out of p features are essentially arbitrary (restricted only to be orthogonal to $\mathbf{1}$) even though the data has a very distinctive structure where all the data lies on the coordinate axes.

6. (15 points) See the MATLAB published file `simpGMM.m`.

7. (15 points) Repeat problem 6 for the dataset `simple2.csv`.

8. (15 points) See the MATLAB published file `genecluster.m`.

9. (15 points) See the MATLAB published file `imageseg.m`.