# CSC412 – Assignment #2

Due: Feb6, 10am at the **START** of class
Worth: 15%
Late assignments not accepted.

## 1 Gamma Distribution

The *gamma* distribution is a distribution on positive real values $0 < x < \infty$. It has the form:

$$p(x) = \frac{1}{sG(c)} \left(\frac{x}{s}\right)^{c-1} \exp\left\{-\frac{x}{s}\right\}$$

$G(c)$ is a function that makes the distribution normalize properly; $s, c > 0$ are positive scale and shape parameters.

- Write this distribution in the exponential family form.
- What are the sufficient statistics of a dataset $\mathcal{D} = \{x^1, x^2, \ldots, x^M\}$ for the gamma distribution?
- Write down the log-liklihood function $\ell(s, c; \mathcal{D}) = \log p(x^1, x^2, \ldots, x^M | s, c)$ in terms of the parameters $(s; c)$, the sufficient statistics of $\mathcal{D}$, and the function $G(c)$.
- For a fixed parameter $s^*$, find the maximum likelihood estimate $c^*$ in terms of the sufficient statistics and $s^*$.

## 2 Flipping Coins

- Write a MATLAB program or script that simulates measuring a biased coin which is flipped $M$ times.
  1. For p(heads)=$\theta$ and p(tails)=$(1-\theta)$, simulate $M$ iid coin tosses.
  2. Based on these $M$ tosses, compute the maximum likelihood estimate $\theta^*$ of the bias.
- For each of the four cases below, repeat the above process 5000 times, and plot a histogram of the 5000 estimates $\theta^*_1, \theta^*_2, \ldots, \theta^*_{5000}$ you generate. (See the function `hist`.) Also report the mean of the estimates.
  (Do not hand in code. Do not fix the random seed in MATLAB . Use 100 bins in your histogram.)
  a) $\theta = .5, M = 10$ b) $\theta = .5, M = 250$ c) $\theta = .9, M = 10$ d) $\theta = .2, M = 250$
- Hint: if you are clever, you can do this whole thing in a few lines, without any `for` loops.
  Consider generating a 5000 by $M$ matrix using the `rand` function.

# 3 Class-Conditional Gaussians

In this question, you'll derive for yourself the maximum likelihood estimates for class-conditional Gaussians with independent features (spherical covariance matrices). Start with the following generative model for a discrete class label $y \in (1, 2, \ldots, K)$ and a real valued vector of $N$ features $\mathbf{x} = (x_1, x_2, \ldots, x_N)$:

$$p(y = k) = \alpha_k$$
$$p(\mathbf{x}|y = k) = (2\pi\sigma^2)^{-N/2} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu_{kn})^2 \right\}$$

where $\alpha_k$ is the prior on class $k$, $\sigma^2$ is the shared variance for all features in all classes, and $\mu_{kn}$ is the mean of the feature $n$ conditioned on class $k$.

- Use Bayes' rule to invert the model above and write the expression for $p(y = k|\mathbf{x})$.
- Write down the expression for the likelihood function $\ell(\theta; \mathcal{D}) = \log p(y^1, x^1, y^2, x^2, \ldots, y^M, x^M|\theta)$ of a particular dataset $\mathcal{D} = \{y^1, x^1, y^2, x^2, \ldots, y^M, x^M\}$ with parameters $\theta = \{\alpha, \mu, \sigma^2\}$.
- Take partial derivatives of the likelihood with respect to each of the parameters $\mu_{nk}$ and with respect to the shared variance $\sigma^2$. (Don't worry about $\alpha$.)
- Set these partial derivatives to zero and solve for the optimal (maximum likelihood) parameter values $\mu_{nk}$ and $\sigma^2$ for classification.

# 4 Handwritten Digit Classification

For this question you will build two classifiers to label images of handwritten digits collected by the United States Post Office. The images `x` are 8 by 8 in size, which we will represent as a vector of dimension 64 by listing all the pixel values in raster scan order. The labels $y$ are $1, 2, \ldots, 9, 10$ corresponding to which character was written in the image. Label 10 is used for the digit "0". There are 700 training cases and 400 test cases for each digit; they can be found in the file `a2digits.mat`. Before we start, here are some MATLAB tips:

- The `imagesc` function can be used to display vectors as images. In particular, try the line:
  `imagesc(reshape(xx,8,8)'); axis equal; axis off; colormap gray;`
  to display the vector `xx`. The `subplot` command is useful for displaying many small images beside each other.
- The `repmat` command in conjunction with `sum` and the operators .* and ./ are helpful in renormalizing arrays so that the rows or columns sum to one.
- The expression `(M > a)` for a matrix `M` and a scalar `a` performs the comparison at every element and evaluates to a binary matrix the same size as `M`.

## 4.1 Conditional Gaussian Classifier Training

- Using maximum likelihood, fit a set of 10 class-conditional Gaussians with a single, spherical covariance $\sigma^2 I$ shared between them to the training data. (This is the same model as in the previous question.)

$$p(y = k) = \alpha_k$$
$$p(\mathbf{x}|y = k) = (2\pi\sigma^2)^{-N/2} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^{N} (x_n - \mu_{kn})^2 \right\}$$

- You should get parameters $\mu_{kn}$ for $k \in (0 \ldots 9), n \in (1 \ldots 64)$ and $\sigma^2$.
  (You can assume $\alpha_k$=1/10 since all classes have the same number of observations.)
- Hand in plot showing an 8 by 8 image of each mean $\mu_k$, all ten means side by side (try using `subplot`).
  Also write somewhere on the plot the value of $\sigma$, the pixel noise standard deviation.

## 4.2 Naive Bayes Classifier Training

- Convert the real-valued features $\mathbf{x}$ into binary features $\mathbf{b}$ by thresholding: $b_n$=1 if $x_n > 0.5$ otherwise $b_n = 0$.
- Using these new binary features $\mathbf{b}$ and the class labels, train a Naive Bayes classifier on the training set:

$$\begin{aligned} p(y = k) &= \alpha_k \\ p(b_n = 1|y = k) &= \eta_{kn} \\ p(\mathbf{b}|y = k, \eta) &= \prod_n \eta_{nk}^{[b_n=1]}(1 - \eta_{nk})^{[b_n=0]} \end{aligned}$$

- You should get parameters $\eta_{kn} \equiv p(b_n = 1|y = k)$ for $k \in (0 \ldots 9), n \in (1 \ldots 64)$.
  (You can assume all class priors are equal since all classes have the same number of observations.)
- Hand in plot showing an 8 by 8 image of each vector $\eta_k$, all ten side by side (try using `subplot`).

## 4.3 Test Performance

- Using the parameters you fit on the training set compute $p(y|\mathbf{x})$ for each of the test cases under both Naive Bayes and Gaussian-conditionals.
- Hand in a 3 by 7 plot showing the image of the test case, the histogram of $p(y|\mathbf{x})$ under Naive Bayes, and the histogram of $p(y|\mathbf{x})$ under Gaussian-conditionals for the following seven *test* cases:
  (digit2,case#3),(digit3,case#14), (digit5,case#242),(digit6,case#112),
  (digit7,case#364),(digit9,case#319),(digit0,case#72).
- Select the most likely class for each test case under each classifier. If this matches the label, the classifier is correct. If not, the classifier has made an error. Hand in a 2 by 11 table showing how many errors (out of 400) each classifier makes on each of the 10 test sets and what the overall error rate (in %) is.