CSC412 – Assignment #2

Due: Feb4, 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 c^*, find the maximum likelihood estimate s^*_{ML} in terms of the sufficient statistics and c^*.
- Derive the mean of the gamma distribution.

2 Estimating the Bernoulli Parameter

- Write a program that simulates measuring a biased coin which is flipped M times.
  1. For p(heads)=θ and p(tails)=$(1-\theta)$, simulate M iid coin tosses.
  2. What is the form of the distribution governing the number of heads you expect to see (we want an analytic answer here)?
  3. Based on these M tosses, compute the maximum likelihood estimate θ^* 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.

  a) $\theta = .5, M = 7$ b) $\theta = .5, M = 256$ c) $\theta = .9, M = 7$ d) $\theta = .1, M = 256$

  Do not hand in code. Do not fix the seed of the random generator. Use about 100 bins in your histogram. (Hint: for speed, consider generating a 5000 by M random matrix first and then doing calculations on it.)

3 Handwritten Digit Classification

For this question you will build three classifiers to label images of handwritten digits collected by the United States Post Office. The images are 8 by 8 in size, which we will represent as a vector $\mathbf{x}$ 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` or `a2digits.zip`. DO NOT HAND IN ANY CODE.

- As a warm up question, load the data and plot a few examples. Decide if the pixels were scaned out in row-major or column-major order, and write that somewhere on your answer to this question.

3.1 Logistic Regression Classifier Training

- Using maximum conditional likelihood, fit a logistic regression classifier to the raw pixel data, after augmenting the vector $\mathbf{x}$ with a constant (bias) term. In particular, fit the model below to maximize the average of $\log p(y|\mathbf{x})$ on the training set:

$$p(y = k|\mathbf{x}, \theta) = \frac{e^{\theta_k^\top \mathbf{x}}}{\sum_j e^{\theta_j^\top \mathbf{x}}}$$

- You should get parameters θ_{kn} for for $k \in (0 \ldots 9), n \in (0 \ldots 64)$ (n=0 is the constant or bias term).

- Remember that you cannot find the model parameters *directly*; instead you need to compute the gradient of the average conditional log likelihood with respect to θ and use that gradient to iteratively maximize the objective. You are free to do this in any way you choose, but the following adaptive gradient procedure, starting with random parameters, should work pretty nicely:

  1. Measure the current objective and gradient.
  2. Adjust the parameters by subtracting Δ times the gradient.
  3. Measure the new objective and new gradient.
  4. If the new objective is better, set $\Delta = 1.1\Delta$
     else if the new objective is worse, go back to the old parameters, objective and gradient, set $\Delta = .5\Delta$.
  5. If $\Delta > \Delta_{min}$, go to (2).

3.2 Conditional Gaussian Classifier Training

- Using maximum likelihood, fit a set of 10 class-conditional Gaussians with a separate, full covariance matrix for each class. In particular, fit the model below to maximize the average of $\log p(\mathbf{x}, y)$ on the training set (where D is the dimension of $\mathbf{x}$ (64 in our case) and $|M|$ is the determinant of the matrix M).

$$p(y = k) = \alpha_k$$
$$p(\mathbf{x}|y = k) = (2\pi)^{-D/2}|\Sigma_k|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_k)^\top \Sigma_k^{-1}(\mathbf{x} - \mu_k)\right\}$$

- You should get parameters μ_{kn} and Σ_k for $k \in (0 \ldots 9), n \in (1 \ldots 64)$.
  (The ML estimates for α_k=1/10 since all classes have the same number of observations in this training set.)

- After fitting the Gaussians, regularize each class' covariance matrix by adding a small amount of the identity matrix. (For this assignment, add $0.01I$ to the sample covariance matrix.)

3.3 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. In particular, fit the model below to maximize the average of $\log p(\mathbf{b}, y)$ on the training set.

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

- You should get parameters $\eta_{kn} \equiv p(b_n = 1|y = k)$ for $k \in (0 \ldots 9), n \in (1 \ldots 64)$.
(Again, all class priors α_k are equal since all classes have the same number of observations.)

3.4 Performance Evaluation

- Using the parameters you fit on the training set compute $\log p(y|\mathbf{x})$ for each of the training and test cases under all three of the Gaussian-conditionals, logistic regression, and Naive Bayes models.
- Select the most likely posterior class for each training and 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 3 column by 22 row table showing how many errors (out of 400) each classifier makes on each of the 10 training and test sets and what the overall training and testing error rate (in %) is.
- What is the average conditional log likelihood achieved by each classifier on the training set and test set. Average both over data cases and digit classes.

3.5 Tips

If you are using MATLAB , here are some 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 `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`.