CSC2515 – Assignment #1

Due: Oct14, 2pm at the **START** of class
Worth: 18%
Late assignments not accepted.

1 Training/Testing Error Curves (2%)

This question asks you to show your general understanding of underfitting and overfitting as they relate to model complexity and training set size.

- For a fixed training set size, sketch a graph of the typical behaviour of training error rate versus model complexity in a learning system. Add to this graph a curve showing the typical behaviour of test error rate (for an infinite test set drawn independently from the same input distribution as the training set) versus model complexity, on the same axes. Indicate on your vertical axis where zero error is and draw your graphs with increasing error upwards and increasing complexity rightwards.

- For a fixed model complexity, sketch a graph of the typical behaviour of training error rate versus training set size in a learning system. Add to this graph a curve showing the typical behaviour of test error rate (again on an iid infinite test set) versus training set size, on the same axes. Indicate on your vertical axis where zero error is and draw your graphs with increasing error upwards and increasing training set size rightwards.

2 Learning Random Boolean Functions (2.5%)

Consider learning random boolean functions under the following setup.

- Target functions with k binary inputs and 1 binary output are generated by independently randomly assigning outputs (with probability one half) to each possible combination of inputs.
- Noiseless training data is generated by randomly selecting a setting of the inputs (uniformly), reporting its output, and repeating this process N times independently.
- Test data inputs are generated by randomly selecting a setting of the inputs (uniformly) and repeating this process M times independently.

Answer the following questions about this setup:

- Let a be the expected number of *distinct* training cases in a training set [Obviously $1 \leq a \leq \min(N, 2^k)$.] What is the expected number b of cases in the test set which *also* appeared in the training set? (In terms of a, k, M.)

- What is the expected test set error rate if we use the following learning algorithm? (In terms of b, M.) *Memorize the training data, and answer 1 as the output for any test case not seen during training.*

- What is the expected test set error rate if we use the following learning algorithm? (In terms of b, M.) *Memorize the training data, and randomly guess the output for any test case not seen during training.*

- Is there any learning algorithm which can do better than the two above? If so, give an example. If not, give an argument of why not in less than 25 words.

- **Hard**: Can you come up with an expression for a in terms of k and N?

3 Marginal and Conditional Numeric Distributions (3.5%)

This question is to get you comfortable with the idea of distributions as multidimensional tables. If you use MATLAB , you shouldn't have to write any programs; you should be able to do everything in the interpreter.
Look at `help sum` to find out how to sum along different dimensions.

- In MATLAB load the file `a1distribs.mat`, otherwise read in `a1distribs.txt`.
- Each multidimensional array `pA,pB,pC` represents a joint probability distribution over discrete random variables. For example `pA(3,4,5,1)` is the probability under distribution A that $\mathbf{x}_1$ takes on its third value, $\mathbf{x}_2$ takes on its fourth value, $\mathbf{x}_3$ takes on its fifth value and $\mathbf{x}_4$ takes on its first value.
- Given a joint distribution $p(a, b)$, recall that $p(a)$ is the marginal distribution of a, and that $p(a|b)$ is the probability of a given (conditioned on) b. Two variables are independent if $p(a, b) = p(a)p(b)$.
  Two variables are conditionally independent given a third if $p(a, b|c) = p(a|c)p(b|c)$.
- Calculate and print out (or copy down) the following distributions. (Only show up to 4 decimal places.)
  1. the marginal probability vector $pC(\mathbf{x}_1)$
  2. the conditional probability table $pA(\mathbf{x}_3, \mathbf{x}_4|\mathbf{x}_1$ takes its third value)
  3. the conditional probability vector $pB(\mathbf{x}_4|\mathbf{x}_2$ takes its first value)

- For each distribution {A,B,C} and each of the following statements, say whether the statement applies.
  (This question is harder than the last one. But don't panic. Remember that answering independence or conditional independence questions is just asking if $p(a, b) = p(a)p(b)$. Numerically, this the same as asking: "Is some matrix equal to the outer product of two vectors?". To that end, you may find the MATLAB commands `rank` and `svd` helpful.)
  1. $\mathbf{x}_1$ is conditionally independent of $\mathbf{x}_2$ given $\mathbf{x}_3$
  2. $\mathbf{x}_1$ is conditionally independent of $\mathbf{x}_2$ given $\mathbf{x}_3$ and $\mathbf{x}_4$
  3. $\mathbf{x}_1$ is marginally independent of $\mathbf{x}_2$
  4. $\mathbf{x}_3$ is conditionally independent of $\mathbf{x}_4$ given $\mathbf{x}_1$ and $\mathbf{x}_2$

4 Class-Conditional Gaussians (3%)

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 D features $\mathbf{x} = (x_1, x_2, \ldots, x_D)$:

$$\begin{aligned} p(y = k) &= \alpha_k \\ p(\mathbf{x}|y = k) &= (2\pi\sigma^2)^{-D/2} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{D} (x_i - \mu_{ki})^2 \right\} \end{aligned}$$

where α_k is the prior on class k, σ^2 is the shared variance for all features in all classes, and μ_{ki} is the mean of the feature i conditioned on class k.

- Use Bayes' rule $p(a|b) = p(b|a)p(a)/p(b)$ to invert the model above and write the expression for $p(y = k|\mathbf{x})$. [Hint: remember that $p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{x}|y = k)\alpha_k$.]

- 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 $\theta = \{\alpha, \mu, \sigma^2\}$. (Assume the data are iid.)

- Take partial derivatives of the likelihood with respect to each of the parameters μ_{ki} and with respect to the shared variance σ^2. (Don't worry about α.)

- Set these partial derivatives to zero and solve for the maximum likelihood parameter values μ_{ki} and σ^2

5 Handwritten Digit Classification (7%)

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 files `a1digits.mat` and `a1digits.zip`. 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`.

5.1 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.)

$$p(y = k) = \alpha_k$$

$$p(\mathbf{x}|y = k) = (2\pi\sigma^2)^{-D/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{D}(x_i - \mu_{ki})^2\right\}$$

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

5.2 Naive Bayes Classifier Training

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

$$p(y = k) = \alpha_k$$

$$p(b_i = 1|y = k) = \eta_{ki}$$

$$p(\mathbf{b}|y = k, \eta) = \prod_i \eta_{ki}^{[b_i=1]}(1 - \eta_{ki})^{[b_i=0]}$$

- You should get parameters $\eta_{ki} \equiv p(b_i = 1|y = k)$ for $k \in (0 \ldots 9), i \in (1 \ldots 64)$.
  (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 η_k, all ten side by side (try using `subplot`).

5.3 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.
- 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.