MIR Assignment 2. Spring 2014 (100 pts)

This assignment requires some programming. You can choose any programming language/environment you like. For this assignment I would recommend Matlab/Octave or Python/Numpy/Scipy/Matplotlib at least for the plots and Weka or Python/scikit-learn for machine learning. The goal of the assignment is to explore the concepts we learned about statistical machine learning. The assignment is due on Feb 19.

IMPORTANT Some questions have additional deliverables for the graduate students taking the course. These are marked with [CSC575]. Undergraduate students are welcome to do them but they are optional and will not affect their grade. The assignment is worth 10% of the final grade. There is some variance in the amount of time each question probably will require. Therefore dont expect them to be equally difficult even though they are all worth the same number of points. Please provide your answers in a SINGLE PDF file. There is no need to copy the assignment specification. Also please answer the questions in order and explicitly mention if you have decided to skip a question. For questions that involve programming provide a listing of the relevant code but not all the details that are not directly relevant to the question. You can either hand me a paper copy of your submission in class or email me an electronic copy.

Hope you find it interesting, George Tzanetakis

Question 1 (20 points)

Most programming environments provide a function that returns uniformly-distributed random numbers in some range. Frequently you are required to generate random numbers that follow other distributions. The exponential probability density function is defined as:

$$p(x) = \theta e^{-\theta x} \tag{1}$$

1. **(4pts)** Plot the exponential probability density function for $0 < x < 100$ and $\theta = 0.1$. Write a function *randExp* that takes as arguments θ and n (number of samples) and returns n random samples of the corresponding exponential density. In order to do that you need to be able to convert the uniformly distributed random numbers your programming language returns to random numbers that follow the exponential probability density function. Read the Wikipedia article on "Inverse Transform Sampling" to see how this can be achied. You will need to calculate the cummulative probability density function (The Wolfram symbolic integration `http://integrals.wolfram.com/index.jsp` can be helpful if you are rusty with calculus). You will also need to invert the cdf i.e express the x as a function of y.

2. **(4pts)** Make a plot of 100 random samples with θ equal to 0.1. The mean and standard deviation of an exponential distribution are both equal to $\frac{1}{\theta}$. Show that the mean and standard deviation of your generated samples is the same.

3. **(4pts)** Create a histogram of your exponentially distributed random samples with 100 bins such that the sum of bars is 1.0. Plot the histogram for 10^2, 10^3 and 10^4 random samples. Confirm that the resulting histograms approximate the original pdf plot.

4. **(4pts)** The goal of this part is to show that the average of many independent random variables that have an arbitrary probability distribution with the same mean and variance is normally distributed (the so-called Central Limit Theorem). Using the *randExp* function create a matrix where each column corresponds to an independent random variable and the number of rows is equal to the number of samples generated by one call to *randExp*. By computing the mean for each column you will obtain essentially n samples of the average distribution. Plot a histogram of those samples for 10^2, 10^3 random variables and 10^2, 10^3 samples. What do you observe ? What is the mean and standard deviation of the histogram ?

5. **[CSC575]** Read the Wikipedia article on rejection sampling and implement *randExp* using the rejection sampling method in addition to the *Inverse Transform Sampling*. Confirm that the results are similar.

Question 2 (30 points)

In this part we are going to explore statistical machine learning by writing classifiers based on likelihood and probability density functions.

1. **(10 points)**

   Write a function to generate a 'training set represented as a matrix where each row is $[x, y, l]$, 'where x and y are your features and l is your class label (1 or 2). 'To generate the training samples you will utilize the *randExp* function you wrote above. Class 1 should contain (x,y,1) points , where x is exponentially distributed with mean value of 20.0 and y is exponentially distributed with mean value of 15.0. Similarly class 2 should contain (x,y,2) points, where x is exponential distributed with mean value of 30.0 and y is exponentially distributed with mean value of 30.0. Create a training set with 50 examples for each class. Make a scatter plot of the resulting training set (the points are put on a plane based on their (x,y) coordinates with different symbol/color for each class).

2. **(10 points)**

   Calculate the Bayes classification error (the "lowest" possible) by estimating the class of each point in your training set using the likelihoods calculated from using the exponential distributions used to create the training set. (Because we created the training set we can know the "best" answer). How many errors are made by this process ?

3. **(10 points)**

   Now, assume that you model the class dependent probability density functions using a multivariate Gaussian classifier with parameters estimated from the training set (basically the empirical mean and standard deviation for each feature). You can assume that the features are statistically independent. Calculate the classification error by estimating the class of each point in your training set using the likelihood using the Gaussian classifier. How many errors are made by this process ?

4. **[CSC575]** Calculate the classification error if you use the Nearest Neighbor rule with one neighbor.

Question 3 (20 points)

Read the paper "Detecting Adversarial Advertisements in the Wild" by D. Sculley, Matthew Otey, Michael Pohl, Bridget Spitznagel, John Hainsworth, Yunkai Zhou available at `http://research.google.com/pubs/archive/37195.pdf`. Frequently machine learning papers focus on algorithms with little information about the context, data preparation, and deployment. This paper presents a straighforward algorithm (Naive Bayes) to classify good vs bad ads. However it mentions many important issues related to actual implementation and building an effective machine learning system. Answer each of the questions below with a minimum of 3-4 sentences each.

1. **(5 points)** What is the class imbalance problem ? How is it handled by the authors ?

2. **(5 points)** Describe what ensembling is ? What are cascade models ?

3. **(5 points)** What aspects of the system are monitored as the inputs change over time ?

4. **(5 points)** Briefly describe in your own words the following ways humans experts are used in the system: active learning, hard-coded rules, and multiple expert judgements.

5. [**CSC575**] Think of a music information retrieval classification task for which the techniques described in this paper would be useful. Describe the task and how the techniques described in the paper would have to be adapted in about 1-2 paragraphs.

Question 4 (30 pts)

To answer the question you will need to download and install Weka from: http://www.cs.waikato.ac.nz/ml/weka/. For all Weka-related questions you will be using the Explorer interface. Alternatively you can also do this question using *Python* and *scikit-learn*.

You will need the following datasets in the WEKA .arff format:
`http://www.cs.uvic.ca/~gtzan/mir_course/Music_Sing_LPCC_.arff`
`http://www.cs.uvic.ca/~gtzan/mir_course/Music_Sing_MFCC_.arff`
In this question you are asked to explore using Weka two data sets that utilize different audio features (Mel-Frequency Cepstrum Coefficients and Linear Prediction Cepstral Coefficients) for classifying audio files to a particular singer and different classifiers that can be used for this supervised learning task.

1. **(5 pts)** Load the Music Sing MFCC .arff dataset. Click on the Visualize All option on the right hand side. If you were forced to use only 1 feature for classification which of the following two features would you choose: Std Mem40 MFCC 12 or Mean Mem40 MFCC 9. Explain your answer briefly. What do the numbers (218, 418) mean in the last graph labeled output ?

2. **(5 pts)** Click on the Classify tab window. Select the ZeroR classifier and 10- fold cross-validation. The classification accuracy is 65.7233. Explain how this number is calculated (hint: look at the output graph of the previous question)

3. **(10 pts)** Select the Bayes category NaiveBayesSimple classifier and 10-fold crossvalidation. The classification accuracy is 87.1069. Based on the confusion matrix in the botoom what percentage of class 0 feature vectors are misclassified as class 1 ? Select use Training Set. The classification accuracy is 88.2. Explain why it is different than the cross-validation approach.

4. **(10 pts)** Select the functions category SMO classifier and 10-fold cross-validation. What is the classification accuracy ? Which class (class 0 or class 1) is easier to classify ?