CSci 543: Advanced Artificial Intelligence **November 9th, 2017**

Due: 5:00 PM, November 24th (Fri.) 2017 **Instructor: Dr. M. E. Kim**

Assignment 3: Learning from Examples + Semi/Non parametric methods (350 points)

Instruction:

For Q1, you have to show its essential computational steps.

For Q2 – Q4,

1) For each computation, **write** the corresponding mathematical formula (or algorithm) that will be used to answer a question in HW file.
2) **Implement** the algorithm or the proper mathematical formulas, to get the answer.
3) Your program should give the **FULL list of answers**.
4) Capture the screen image of your answer or prepare the output file.
5) Insert the image to the corresponding question in the HW file,
6) Submit the source codes + HW file + output files and/or image files in the .zip file.

Do implement the algorithm or the mathematical computation for yourself using your favorite language. Do **NOT** use their existing or online program. It will get NO point if you simply run the data with the online package/program of algorithm.

Any solution neither with the proper formulas nor with the computational steps will get **no** point.

You should work on the assignment**,** ***independently***. – Any plagiarism or collaboration is **not** allowed. – Refer to the syllabus and Code of Student Life.
http://und.edu/student-affairs/code-of-student-life/_files/codepdfs/appendix/iiia/iiia-3.pdf

Submission:

1. Name your HW and .zip file in the format**, HW3-YourLastName.(docx, zip)**: e.g.) HW3-Kim.docx, HW3-Kim.zip.
2. Upload it in the blackboard system.

Q1. [50] Decision Tree

Suppose we have a system that observes a person's TV watching habits in order to recommend other TV shows the person may like. Suppose that we have characterized each show by whether it is a comedy, whether it features doctors, whether it features lawyers, and whether it has guns. The examples about whether the person likes various TV shows are given in the table.

Example #	Comedy	Doctors	Lawyers	Guns	*Likes*
e1	false	true	false	false	*false*
e2	true	false	true	false	*true*
e3	false	false	true	true	*true*
e4	false	false	true	false	*false*
e5	false	false	false	true	*false*
e6	true	false	false	true	*false*
e7	true	false	false	false	*true*
e8	false	true	true	true	*true*
e9	false	true	true	false	*false*
e10	true	true	true	false	*true*
e11	true	true	false	true	*false*
e12	false	false	false	false	*false*

This data is used to learn the value of *Likes* (i.e. to predict which TV shows the person would like based on the attributes of the TV show). Suppose we measure the error of a decision tree as the number of misclassified examples. The optimal decision tree from a class of decision trees is an element of the class with minimal error.

(1) [10] (A) Give the optimal decision tree of depth 1, with only one node. Note that a depth of root is 0.
(B) Suppose that the error is the 0/1 loss. What is the error of this tree? .

(2) [5] Compute the initial entropy of *Likes*.

(3) [15] Draw the full decision tree that is learned from the above data by optimizing the information gain at each step. Show the computation of information gain to choose a root of each subtree.

(4) [10] For the given test data below, what is the test set error rate using 0/1 loss?

Example #	Comedy	Doctors	Lawyers	Guns	*Likes*
e13	true	true	true	true	*false*
e14	true	false	true	true	*false*
e15	false	true	false	true	*false*
e16	true	true	false	false	*true*

(5) [10] Is the data in (3) linearly separable? Explain why or why not..

Q2. [50 + 50(program)] *k*-Means Clustering

Implement the *k*-means algorithm in the slide with the 4-dimensional data at '*data1.xls*'. The initial starting points for *k* cluster means can be chosen randomly, where *k*=3. Run the algorithm 10 times with the different initial points, and compute the total sum of squared error after each run. Select the solution that gives the lowest sum of squared error over 10 runs. For the convergence, check if any of the data points change its cluster assignments relative to its previous assignment; i.e. if 1 or more points change cluster assignment, continue to optimize it.

$$L_2(\mathbf{x}_i, m_{C_i}) = \sum_{d=1}^{4} (x_{id} - m_{C_i})^2$$

$$\text{Total Sum of Squared Error} = \sum_{i=1}^{N} L_2(\mathbf{x}_i, m_{C_i})$$

For each cluster C_i, give

(A) [10] its mean (m_i),

(B) [10] the size of cluster (i.e. the number of data assigned to C_i), and

(C) [30] the list of data which are assigned to each cluster C_i.

Do not use the existing or online program of k-means algorithm, but do program for yourself, in your favorite language.

Q3. [60 + 50(program)] Gaussian Mixture Clustering

Similarly, implement Gaussian mixture clustering using EM algorithm for the data at *data1.xls*. Assume a Gaussian mixture model of k components for the data where k=3 and we find the parameters of the model using EM algorithm.

Initialize the parameter values using the result of Q4: i.e. C_i and m_{C_i} from Q4, thus, P(C_i) = # of data in C_i / N, and compute the covariance of each initial group using the data assigned to each cluster in Q2. For the convergence, compute the value of the log-likelihood after each iteration and stop when there is no significant change within the ε = 0.0001. The log-likelihood under i.i.d assumption is defined as:

$$L(\Phi|X) = \log \prod_{t=1}^{N} p(x^t | \Phi) = \sum_{t=1}^{N} \log \sum_{i=1}^{k} p(x^t|C_i)p(C_i)$$

where $\Phi = \{p(C_i), p(x^t|C_i)\}_{i=1}^{k}$ is the weight for the k^{th} component and the Gaussian density for the k^{th} mixture component, respectively.

For each component (i.e. cluster) C_i, give

(A) [5] its weight $w_i = p(C_i)$
(B) [5] its mean (m_i),
(C) [5] its covariance (S_i)
(D) [10] $p(C_i|x)$ for each data
(E) [5] the size of component, and
(F) [30] the list of data which are assigned to each component C_i.

Do not use the existing or online program, but do program it for yourself, in your favorite language.

Q4. [50 + 40(program)] *k*NN: Nonparametric Classification

The k-nearest neighbor class of estimators adapts the degree of smoothing to the local density of data, which is controlled by the number of neighbors, k << N where N is the size of data.
Using the same data at *data1.xls* and the clustering result from Q3,

(A) [10] Compute the smooth class-conditional density estimation of each class C_i, $\hat{p}(\mathbf{x}|\boldsymbol{C_i})$, using kNN where k=30 .

A distance d_k(x) is computed by Euclidean distance for a window size h.
Use a Spheric Gaussian Kernel function with Euclidean norm:

$$K(\mathbf{u}) = (\frac{1}{\sqrt{2\pi}})^d \exp(-\frac{\|\mathrm{u}\|^2}{2})$$

(B) [10] For each data x, compute the discriminant function g_i(x) for each class C_i.

(C) [30] Based on the result of (B), decide to which class C_i each data x is assigned and give the list of data assigned to each C_i .