**CSci 8980: Data Mining (Fall 2000) Project 1**
Due date: 20 Nov 2000

In this homework, you will use Clementine to apply clustering and classification algorithms to various datasets. Clementine is an interactive data-mining tool, having a nice user-interface for various classification, clustering, visualization and analysis tools. It is installed on ITLABS (all Sun workstations in 4-204) and some computers in CS domain (including but not limited to femto, deci, quasar, exa, tera and atto).

In order to run Clementine, you should first load the corresponding module by the command: "module load soft/clementine", and run the executable named "clementine" from any directory in which you have a write access.

For more information about the program and a few hints please see the end of this document, we may also add additional information on the web page if the need arises.

The datasets that you will be using for this project as well as some other datasets that you can play with are available at the following url: http://www-users.cs.umn.edu/~oztekin/courses/csci8980/datasets. Files with "data" extension are the actual datasets, and the ones with "names" extension contain information about the corresponding datasets if available.

The datasets that we will use for this project are:

for classification:

a) cmc.data
b) zoo.data
c) chessdata.data

for clustering:

a) chessdata.data
b) five_circles.data

For more information about these datasets you can read the corresponding information files. (cmc.names, zoo.names etc.)

In the case of two-dimensional dataset (chessdata), it would be a good idea to plot the points (using the plot function); for others, it is possible to use the "table" function to have a glance at the individual records. A small walk-through will be given at the end of this document.

## Questions

### 1. Classification

For each of the classification datasets do the following:

Divide the dataset into training and test sets: use about 70% of the data for training set and 30% for the test set.

Run 3 classification algorithms (build C5.0, train net, build rule) using default parameters.

For each dataset and algorithm pair

- Report test classification accuracy for both training and test sets.
- Comment on the quality of the classification. If for a given dataset, a method works better than others, write a few sentences about why do you think that happens. If a method turns out to be better or worst in general (or for most of the datasets), express your opinion about why that happens in a few sentences. (This whole thing, no more than half a page please).
- If you observe an unusually poor accuracy (below 10%), check the dataset and try to argue about its cause, explain why that might happen. Is there a way to handle it? Can you improve the accuracy? If the answer is yes, please do so.

Note that some of the methods such as neural networks have many parameters; the quality of the results depends heavily on the dataset, on the number of hidden units/layers, learning and momentum rates etc. Once you have a good feeling of the software and datasets, you may try the expert options and adjust parameters to improve quality. For the case of neural networks you may especially play with the number of hidden units to obtain better results. As a rule of thumb, number of hidden units must be the smallest number that "works". A large number results in overtraining/overfitting and poor test accuracy, a too small number will result in an over-simplified network, inadequate to deal with complex functions and/or complex decision surfaces your dataset requires. If you have used expert options, please write down the parameters you have chosen.

## 2. Clustering

For each of the clustering datasets (five_circles.data and chessdata.data)

- Run k-means using default parameters with number of clusters:
  1) Same as number of classes
  2) Twice as many as number of classes
  3) Four times as many as number of classes

  For the case of chessdata, you can consider number of natural clusters (9) instead of number of classes in the dataset which is 2.
- Plot histograms using the original class and the cluster numbers and provide discussion concerning the quality of clusters shown through squared error and the histogram.

To obtain the squared error, you can save the results from a table, use a tool like excel or write a small program in your favorite language to calculate the squared error.

Note that the added field $KMD-Kmeans is the Euclidian distance of the point from its cluster center.

### 3. Normalization

Normalization is an important preprocessing technique. It consists of mapping ranges of attributes to similar ranges so that none of them will take precedence over others.
For example:
Let us suppose we have a survey of people working in a company and we the following attributes:

1) Age (ranging from 35 to 50)
2) Salary (ranging from 80000 to 150000)
3) Number of children (ranging from 0 to 3)
4) Another attribute ranging from 11.12 to 11.24

Etc.

For example a straightforward way of normalizing this dataset is to map each range to [0,1] using linear mapping (and scaling). For the case of salary for example 80000 will be mapped to 0, and 150000 will be mapped to 1 etc. The mapping formula can be:

Newvalue=(oldvalue-minvalue)/(maxvalue-minvalue)

In this case the inverse transformation will be:

Oldvalue=Newvalue*(maxvalue-minvalue)+minvalue.

Here minvalue and maxvaule corresponds to the minimum and maximum values for a given attribute (they are different for each attribute, for the case of salary they are 80000 and 150000 respectively). Note that this is a one to one mapping.

After doing the normalization for each of the attributes you use classification or clustering algorithms. Some of the algorithms may benefit dramatically from the normalization in terms of convergence speed and/or quality.

For example for the case of neural networks, normalization does not add much to the quality of the results obtained but, it may decrease the time required for convergence considerably for some datasets (not necessarily the ones we use in the project).

- Consider the standard K-Means clustering algorithm. Let us suppose that we have a dataset in which some of the attributes have a very large range with large numbers, and some of them are limited to a small range (just like the example given above). Write a short paragraph about the results that you would obtain using K-Means with and without using normalization in a dramatized case (some of the attribute ranges may be orders of magnitudes larger than others).
- Let us suppose that we used normalization before applying K-Means, and obtained k cluster centers. For a given cluster center. We can apply the inverse mapping for each coordinate (attribute) of a cluster center and obtain the mapping into the original space. Let us use the inverse mapping to map all of the cluster centers to the original space. Consider two cases for determining which points belongs to which cluster.

1) Apply normalization to the original dataset, work completely in the new coordinate system. Find K-Means cluster centers, assigns points to the cluster whose center is closest to the point in the normalized space.
2) Use the cluster centers found in the previous case. Map the cluster centers back to the original space and assign points to the clusters in the original space i.e. use original points and mapped cluster centers to determine the distances.

Will the two methods produce the same clusters? Is it possible that a given data point be assigned to different clusters by the first and second method (ignoring round-off errors)? Write a short paragraph explaining why or why not that can happen.

Hint for the whole question: standard K-Means works in Euclidian space and uses Euclidian distances.

**4. Decision trees**

Assume that we are using decision trees in classification and we do not use any pruning i.e. the tree can go as deep as it is required to learn the training data completely (it can overfit, and actually we are trying to overfit as much as possible). Is it possible for some dataset to have less than 100% training accuracy in this case? Explain your answer with a few sentences, its probable causes, or why this cannot happen.

**Clementine, a quick walkthrough**

Download the datasets to a directory in your account, do the steps explained in the beginning of the document to run the program.

Click on “tool -> sources -> var file” to select it and click on somewhere on the workspace to insert a dataset. Double click on the newly created icon to attach a dataset to it. Important, do not forget to check the “Get fields names from file” if the dataset has description of the fields as the first line (most of the datasets do).

Select “records ops -> sample” and put it somewhere in the workspace, connect the first icon to the newly created one using the second mouse button (or third button depending on your system). “Sample” is used to select only a portion of the dataset. For example, if you want to divide your dataset of size say 100 entries into two sets of size 40 and 60 records, you can attach two sample nodes to a data source and set the first one to get the first 40 records, the second one to skip the first 40 and get the rest. If you want to use the whole dataset, you can directly connect the source to the types node discussed next.

Use “field ops -> types” to supply the correct inputs to further nodes in the stream. If you double click on this one, you will see that you have options to make some attributes “in” or “out”. If you will use clustering for example, you would like to suppress the class information. To do so, you can set the class field as “out”, and let the others stay as “in”. The next step is to add a classification or clustering method from the modeling tools. Attach the “types” node to the modeling node you have selected, double click on the icon, adjust parameters, and hit execute. This will build a new model and place it on the right. In order to test it, plot results, analyze it etc., you should bring it to the workspace (do this just like adding any other components). You can now feed it the test set (or the training set) if this is a classification, which will add to the stream additional attributes such as k-means cluster that the record maps to etc. You can plot, analyze, take histogram by attaching, analyze, plot, histogram or other components afterwards. Note that as far as I observed, Clementine does not update the models already placed on the workspace, so if you update some parameters, you should delete the old one and reinsert the new one. (selecting a component and hitting the del key should remove it).

Due to the fact that some xwindows implementations bring forward the window that the mouse points to, some of the dialog boxes or error boxes may go under the main window. An easy way to get back the window that went under is to right-click on the title bar of the main Clementine window.

Some of the methods may require more memory then the amount allocated by default. You may want to increase this from Options -> Memory requirements. But please do not overkill, as other people will also use the same machines. Allocating more than enough memory will degrade overall performance for everyone.

Please start to work on the project as soon as possible, getting used to the user-interface takes some time. Getting the results and improving the quality also should take time. If you start one or two days before the due date, you may not have enough time to finish it, moreover by that time most of the students will probably try to finalize their projects and the machines will be more overloaded then usual.

You can contact Uygar Oztekin (oztekin@cs.umn.edu) for questions regarding this assignment. If you put Csci8980 to the subject line, it would be easier and quicker for me to distinguish them from other emails, allowing me to respond faster.