Project Description

Background Information

Machine learning is the application of statistical techniques to data, allowing conclusions to be drawn about existing data, or the prediction of values in new data. It is a subset of artificial intelligence. It effectively allows the machine to make some decisions, without exhibiting some of the biases which a human would. The reliability of a model is still heavily dependent on the data which was used to train it though, which makes it important for a human to choose this carefully.

A dataset is a collection of related data, which has the same format. In our case, it will be in a tabular form. The columns will be referred to as 'attributes', or particular characteristics of a particular row. Each of these columns will have a singular data type, but may have multiple allowed values if the data supplied is categorical in nature. A row will be referred to as an instance, and should contain either a valid value or missing value for every attribute.

Datasets with missing values are an interesting problem, since we have no way of knowing what the missing value represents. There are several approaches to solving this problem, but the most common solutions involve either ignoring the instance containing the missing value, or else making an educated guess about what the missing value represents. Both mentioned approaches have positives as well as negatives. Ignoring an instance prevents you from using unreliable data, but also may lead to valuable data in the other attributes of that instance being discounted. Attempting to predict a value will probably result in you getting the value wrong, but it may be worth the risk to gain value from the other complete attributes.

Problem Description - Extending Classifiers for Incomplete Data

This project aims to investigate the usage of machine learning techniques on incomplete data sets of discrete data. It will attempt to fill in missing data in certain data sets by using training data to build a model, which we will then apply to the target data set. We will keep using this model to generate predictions on produced data sets recursively, and will stop whenever the produced data set stops changing. It is expected that the prediction will trend towards the mean, and more reliable results will be achieved.

Given sufficient time, it is also intended that this approach can be used to infer 'hidden variables', similarly to how neural networks produce hidden layers to create relationships between data. This will be achieved by adding a new attribute to a dataset, and initially filling it full of random valid data. We will then repeatedly build, apply and rebuild models to classify this attribute until it stops changing, in an identical manner to how we're filling in new columns. New data will be added to the target data set by this method, and we will investigate how useful this hidden data is.

A package in Java for the machine learning platform Weka will be created to accomplish the goals described above. Weka is an open source machine learning platform which was initially developed at the University of Waikato, in New Zealand. It has a lot of pre-existing implementations of popular machine learning algorithms, as well as a wide range of plugins which implement more specialised functionality. It is currently my intention to package our solution as a plugin, and to distribute it. However, this depends on the utility of the results.

Graphical representations of the results which are observed using this method will be produced, and compared against results which are observed using other standard methods. This functionality is built into the Weka Experimenter, and should allow the project to easily be tested against other algorithms which are already implemented within Weka.

Input datasets should be in the .arff format which has been developed for Weka. This is effectively the same as a .csv file, except with some additional data at the top and using the '?' symbol to represent values which are missing.

Input datasets should only have nominal attributes (attributes which fit into categories), at least initially. This is for two main reasons:

1. Only having nominal attributes should simplify the implementation. This will allow us to determine whether or not the experimental technique is valid more easily, so that it could be extended to numeric types in future.
2. It should ensure convergence. With numeric data, it is probable that our data would never stop changing as we create and reapply the model. This appear as a predicted result fluctuating around a particular value from prediction to prediction, without ever settling on a number. While it would be possible to write checks and validation to prevent this, it adds extra complexity without adding much additional value.

Acceptance Criteria

- The project will be successful if the implemented model can be iteratively applied to data sets until no further change is observed.
    - This should at least be implemented as an unsupervised Weka classifier
    - It may also be implemented as a filter, and as a supervised classifier.
- Graphs should be produced comparing the results obtained through the Weka experimenter against at least the following algorithms:
    - 0-R
    - J48
    - Naïve Bayes
    - Random Forest
- If it is possible to use this on hidden variables, then this will be further success. Any further interpretation of these hidden layers will be interesting from a research perspective.
- If the observed results are useful, then the project should be distributed as a Weka plugin.

Expected Schedule

I'm studying three modules before Christmas and one module after, so most of the work will probably be done in the second semester. Also, it'll be important to react to the results that we're seeing, so the end of the project will really depend on observations.

Semester 1

Week	Date	Tasks
1	31/10/2016	Get familiar with Weka and previous projects which have been produced in a similar area.
2	7/11/2016	Develop an understanding of machine learning + classification. Do tutorials and read up on things in that field.
3	14/11/2016	Start development in Weka
4	21/11/2016	Further development
5	28/11/2016	Produce a mostly working implementation of the prediction algorithm in Java
6	5/12/2016	Fine tune + refactor implementation, preparation for demonstration.

At the time of producing this document, much of this work is already done. The code isn't as clean or concise as I would like, but I've really been pushed for time due to assignments in other modules.

Semester 2

On the week following the conclusion of my exam, I intend to start working in the second semester. I should have more time at this stage. The schedule runs until the end of February, even though we are advised to have development completed by March. This is because the tasks at the end of the project will be heavily dependent on observed results. I think this flexibility is necessary for a more research oriented project. It is therefore impossible to have a complete plan for the month of March, other than to say it will be approached pragmatically.

Month	**Date**	**Tasks**
January	26/01/2016	Clean up existing codebase, and ensure all functionality works as expected through all possible interfaces (GUI, CLI). This will be difficult due to the pure quality of Weka documentation. At this stage, options should be added to allow the user to select which classifier we should be using internally for our method. This should allow for experiments to be performed more easily.
February	2/02/2016 – 01/03/2016	Implement a way of adding hidden attributes, and of instantiating these columns with random valid values. Investigate if they improve predictions. If observed predictions are better, then investigate the effect of changing the number of hidden attributes added. Also, retrieving further publically available .arff test files could be of use at this stage. These may need to be cleaned up, or trivially modified in order to make them viable test data sets for this project. Either way, it makes sense to test on a number of different data sets before making any kind of judgement on the utility of the classifier.
March	02/03/2016	Further investigation. Adapt project based on earlier observed results.