Predicting one-year postoperative death rates following thoracic surgery

Craig Heptinstall Crh13- 110005643
SEM6420
Institute of Computer Science
Aberystywth University

***Abstract*—Predicting medical data is becoming more and more vital towards helping reduce death rates, or reducing risks in medical procedures. With the use of tools such as WEKA [1], the ability to build machine learning classifiers has become easier, and has helped produced more reliable results. This paper looks at death rate prediction following thoracic surgery, and compares different means of prediction and configurations on a given test set of data.**

I. Introduction

Thoracic surgery is a means of lung resection, and is performed in order to remove part or all of a lung from a patient who is suffering or has suffered from lung cancer [2]. The data provided for training a machine learning approach to predicting the death rate one year after surgery contains 300 patients, each of which hold several attributes, including age and 'Risk1Yr'. The most latter of these states is the example patient survived (indicated by a 0) or did not (indicate by a 1). The aim of the classifier trained at the end of this report is to be able to predict the survival rate in conjunction with an identical test set of data.

For easier statistical reading, and the ability to easily try a range of classifiers on the training data, Weka was chosen to perform experiments and training. Weka allows pre-processing of data, classifiers and filters to be applied, and also provides good visualisation of training results, such as ROC curves. Throughout this paper, both graphs generated within Weka, and ones created from comparing different classifiers will help to reinforce result conclusions.

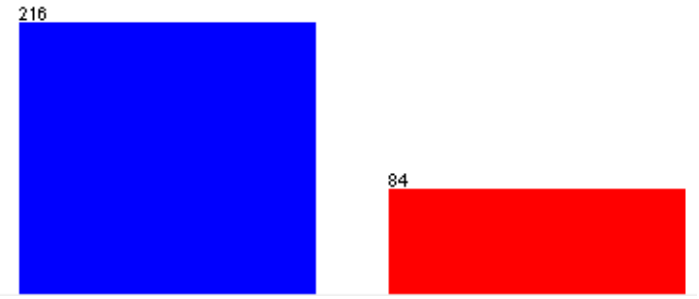
To visualise data quickly during the fine tuning and selection of classifiers during the start of this project, the Weka 3.7 Explorer application will be used, and then will be moved onto the experimenter during latter parts of the project in order to quickly run multiple experiments at once. Finally, the Weka Knowledge flow application will allow comparisons of the ROC curve of multiple solutions in order to choose the most suitable set up for predicting the test data results.

As stated in the assignment, more than one classifier will be selected and compared to see which one best suits predicting this type of data. Following that, more comparisons will be made looking at changing a range of hyper-parameters for chosen classifiers.

A. Data pre-processing

In Weka, and in any machine learning approach, some data pre-processing should be performed. Because the data set is real-life based, data is not always complete, or follow a linear pattern. Due to this, an issue known as class imbalance can hinder the performance of learning systems, and means that one instance of an attribute (in this example, surviving patients) is the majority sample causing a bias when training the machine learning algorithm. On first inspection of the data in the training set, it can be seen in Figure 1 that the 'Risk1Yr' attribute holds far more surviving patients that ones who did not.

Fig. 1. Visualisation of training set risk year 1 mortality rate. Left: Survivors, Right: Deaths



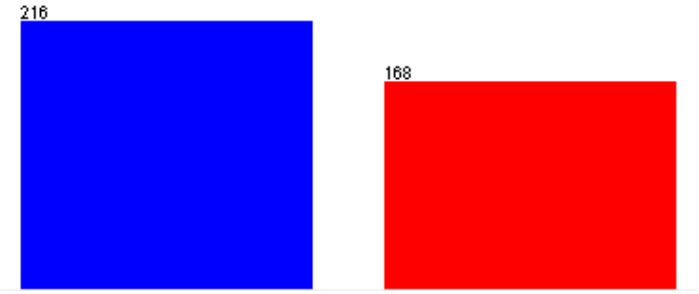
There are two means of settling imbalances in data however [3]:

1) Under-sampling where the size of the samples are reduced proportionally
2) Over-sampling where the size of samples are increased proportionally

In the case of Weka, there are filters available within the pre-processing stage of experiments allowing the addition of under or over-sampling. In this case, the SMOTE (Synthetic Minority Over-sampling Technique) [4] plugin for Weka offered an easy means of re-sampling data based on a given set of parameters. In an accompanying paper [5], SMOTE is described as creating 'synthetic examples' rather than replacements that are generated in feature space rather than data space. In the Weka library, synthetic samples are created by using a range of nearest neighbour minority samples, and placing new samples

between them. By default, the SMOTE filter uses five nearest neighbours to create new samples, and as this is recommended for this size dataset, this will remain the selected value.

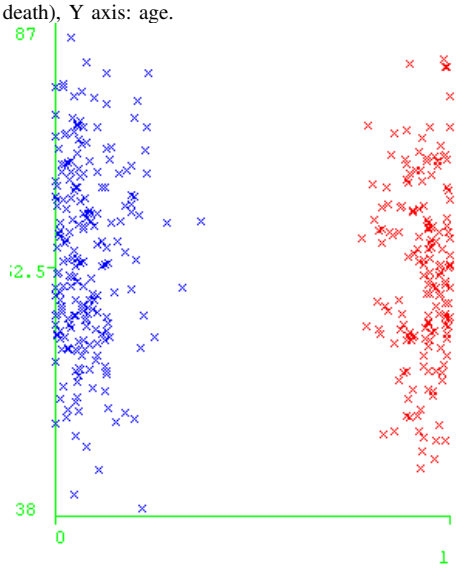
Fig. 2. Risk one year data after SMOTE filter applied.



Once the SMOTE filter was applied, there was a clear view of change in the proportions of the survival and death rates. As Figure 2 shows, over-sampling has been applied to 200%, which has meant the minority of samples showing death rates only has 50 samples less than the majority as opposed to the 138 sample difference previously.

In addition to balancing data, viewing the data in Weka explorer on scatter gave good insight into any data trends, and also gave hints into which attributes carried more weighting towards the risk attribute than others. At first, both age and risk were plotted against each other to see if there were any trends towards being younger meant better survival rates (as shown in Figure 3). In order to view the scatter plot well enough, Jitter was added in Weka to plot points further away from each other.

Fig. 3. Scatter plot comparing age and survival rates. X axis: Death rate (0- survival, 1- death), Y axis: age.



Although the plot looks fairly evenly spread on age for both survival and mortality rates, there is a slight higher number of patients who survived aged around 50 than those who were older.

By looking at the scatter plots for various combinations of data, dimensionality reduction is something that can be used to both increase the accuracy and speed of the resulting algorithm. There are various means of dimensionality reduction, such as PCA (Principle Component Analysis) and SOM (Self Organising Maps) /citedimensionality, whilst in Weka, attributes can be ranked, in order to see which ones carry the most important in a data set.

For this experiment, the InfoGainAttributeEvaluator was used to rank attributes in Weka. This algorithm was selected because of the simple output of 'worth of an attribute' that allows the least useful attributes or features to be omitted from any classification later on. The attribute evaluator configuration run can be found in Appendix A From first run, and shown in Figure 3 is the results from the ranking. It appears that three attributes listed have no effect on the prediction of one year risk, so these will be omitted when testing classifiers in the following section.

Fig. 4. Ranked attributes using the Information gain attribute evaluator in Weka.

```
Ranked attributes:
 0.07451525    1 DGN
 0.03291264    3 PRE5
 0.02731049   10 PRE14
 0.01366042   14 PRE30
 0.01060421    4 PRE6
 0.00950188    8 PRE10
 0.00826647    7 PRE9
 0.00600333    9 PRE11
 0.00567997    5 PRE7
 0.00317208   12 PRE19
 0.00210082   13 PRE25
 0.00161183    6 PRE8
 0.00000876   11 PRE17
 0            15 PRE32
 0             2 PRE4
 0            16 AGE
```

Although only the three at the bottom were removed from this process, it is possible to remove more features. Three was chosen because it appeared to be a global minimum feature set before over-fitting occurred. This is when the learning curve appears to go back above its minimum due to not enough descriptive features.

II. Classifying Data

For the set of experiments following, multiple classifiers have been selected to find the best prediction possible. In a notable paper by D. Wolpert (The Lack of A Priori Distinctions Between Learning Algorithms) [7], this points out that there is not one classifier that will work for every problem in machine learning. Therefore, by comparing a range of classifiers, the best traits will be found, and will contribute towards the final

predictor provided in the appendices. This section takes a look at what classifiers have been selected, how each should work, and what fine tuning was done to each.

A. Classifier selection

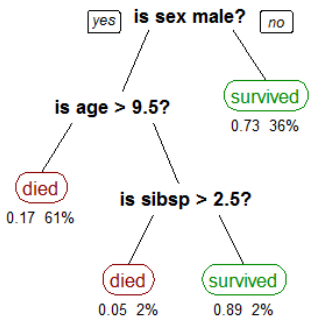
Three classifiers have been chosen for this part of the paper, and will be compared directly and indirectly in the results section of this report. The three classifiers selected are as follows:

- Decision tree - J48 for generating C4 trees
- Naive Bayes - A Naive Bayes classifier
- Neural network - Multilayer perception classifier using back-propagation

The first of which was selected because of both its practicality and popularity, which has been proved useful for data mining. Decision trees are used in a range of medical diagnoses, and because decision trees allow human readable rules of classification and are easy to interpret, they are very useful in this field of research.

With J48, similarly to most decision trees, the structure created appears like a flow chart, with each node denoting a test, each branch denoting a an outcome of a test, and each leaf node holding a class label. This classifier has two parts [8] to its use: the growth phase and the pruning phase. Trees split training sets based on the criteria (in this case mortality rate), and perform this until all records belonging to each split hold the same class label. Over-fitting is the outcome of creating trees because of this, therefore pruning is added to the scenario. Outliers and fuzzy data from the tree to ensure it holds only useful information. Construction of flow chart is quick, therefore the overall prediction rate in this paper should be quicker than other algorithms.

Fig. 5. Example of a decision tree, showing the attempt of using different ages to seperate cases.



The second classifier listed that will be compared in order to find out which algorithm will be most suited to this problem is a using a Naive Bayes approach. The simplest of the three, Weka offers an implementation of this which uses Bayes rule as shown in Figure 6. From first observation, it does appear that Bayes may work better with more evidence, though this may not be the case, as over-fitting may occur.

Fig. 6. Naive Bayes rule, where more evidence (attributes can be added after the first and multiplied).

$$P(\text{outcome}|\text{evidence}) = \frac{P(\text{Likelihood of Evidence}) \times \text{Prior prob of outcome}}{P(\text{Evidence})}$$

Bayes has always been another popular machine learning approach to medical data, and many papers have justified the use of it. One such paper which looked at predicting heart disease [9], gives good reasons for the choice of the algorithm. When data amounts are high, this way of training and predicting does not exponentially increase in terms of time. Where attributes are independent of each other, Naive Bayes can handle it better than other algorithms (as shown in pre-processing, there are attributes in this papers example for the need of this), and output is more efficient compared to other methods.

The final classifier that will be used in the prediction of thoracic surgery mortality risk will be with neural networks. In Weka, by default one neural network library comes pre-installed, Multilayer perception. Artificial neural networks (ANN) are a more recent development in medical diagnosis, and come from more of an artificial intelligence background. These are used widely in science, and are popular where attribute relationships may be unknown or very complex [10].

B. Configuration

III. Results discussion

IV. Conclusion

APPENDIX

All configurations have been run in Weka version 3.7.13

A. Data pre-processing/ attribute selection

weka.attributeSelection.InfoGainAttributeEval
weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1

REFERENCES

[1] University of Waikato, *Weka 3: Data Mining Software in Java*, Machine Learning Group at the University of Waikato, *http://www.cs.waikato.ac.nz/ml/weka/*, 2013.

[2] Boston Medical Center, *Center for Thoracic Oncology*, Boston University school of medicine, *http://www.bmc.org/thoraciconcology/treatments/lung-resection.htm*, 2014.

[3] M. Rahman, *Machine learning based data pre-processing for the purpose of medical data mining and decision support*, Hull University, 2014.

[4] Microsoft, *SMOTE*, Microsoft Aure Modules, *https://msdn.microsoft.com/en-us/library/azure/dn913076.aspx*, 2016.

[5] N. Chawla et al, *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research 16 (2002) 321357. 2001.

[6] S. Reid, *Dimensionality Reduction Techniques*, Turing Finance, *http://www.turingfinance.com/*, 2014.

[7] DH. Wolpert, *The Lack of A Priori Distinctions Between Learning Algorithms*, Neural Computation. 1996;8:1341-1390. 1995.

[8] D. Lavanya, *Performance Evaluation of Decision Tree Classifiers on Medical Datasets*, International Journal of Computer Applications (0975 8887). 2011.

[9] R. R.Patil, *Heart Disease Prediction System using Naive Bayes and Jelinek-mercer smoothing*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 5. 2014.

[10] F. Amato et al, *Artificial neural networks in medical diagnosis*, Journal of applied biomedicine 11: 4758. 2013.