Mathematical and Software Engineering, Vol. 1, No. 1 (2015), 1-5
Varεpsilon Ltd, http://varepsilon.com

Predictive Models for Post-Operative Life Expectancy after Thoracic Surgery

A. Nachev and T. Reapy

Business Information Systems, Cairnes Business School, National University of Ireland, Galway, Ireland

Abstract

This paper studies data mining techniques used in medical diagnosis, particularly for predicting chance of survival of a patient after undergoing thoracic surgery. We discuss models built using decision trees, naive Bayes and support vector machines and explore suitability of each of the algorithms to perform on such data.

Subject Codes (ACM): I.5.2

Keywords: data mining applications, support vector machines, decision trees, naive Bayes, thoracic surgery.

1 Introduction

A major clinical decision problem in thoracic surgery is selecting patients for surgery, taking into account possible risks and benefits for the patient. Among the factors considered are long-term, related to life expectancy and mortality prognosis in a time horizon one to five years, and short-term, related to post-operative complications.

Traditional methods for decision support include standard statistical modelling, based on Kaplan–Meier survival curves, hierarchical statistical models, multivariable logistic regression, or Cox proportional hazards regression [1, 2, 3]. Other methods used to predict post-optative survival are risk-scoring systems [5], web-based applications [4] or statistical software packages. Zeiba et al. [1] also proposed boosted support vector machines for clinical diagnosis by using imbalanced datasets.

Taking into account limitations of the predictive methods for post-operative life expectancy and the data used for that, we explore the usage and performance of several machine learning and data mining techniques by empirical analysis

based on a real-life dataset.

The paper is organised as follows: In Section 2 we review the classification techniques and methods applied to predict life expectancy. In Section 3 we discuss the experimental results. Section 4 provides conclusions.

2 Methods and Algorithms

Support vector machines (SVM) are common machine learning techniques, used for classification or regression. Training data is a set of points of the form

$$D = \{(x_i, c_i) \mid x_i \in \Re^p, c_i \in \{-1,1\}\}_{i=1}^n \quad , \tag{1}$$

where the c_i is either 1 or -1, indicating the class to which the point x_i belongs. During training, SVM constructs a p-1-dimensional hyperplane that separates the points into two classes (Figure 1). Any hyperplane can be represented by $w \cdot x - b = 0$, where w is a normal vector. Among all possible hyperplanes that might classify the data, SVM selects one with maximal distance (margin) to the nearest data points (support vectors). Building a linear SVM classier is formally a constrained optimization problem (2).

$$\min_{w,b,\xi_k} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i \tag{2}$$
$$s.t. w \cdot x + b \geq 1 - \xi_i$$

In dual form, (2) can be represented by:

$$\min_{\alpha_i} \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j - \sum_{i=1}^n \alpha_i \tag{3}$$

$$s.t. 0 \leq \alpha_i \leq C, \sum_{i=1}^n \alpha_i c_i = 0$$

The resulting decision function $f(x) = w \cdot x + b$ has a weight vector $w = \sum_{k=1}^n a_k y_k x_k$. Data points x_i for which $\alpha_i > 0$ are called support vectors, as they uniquely define the maximum-margin hyperplane.

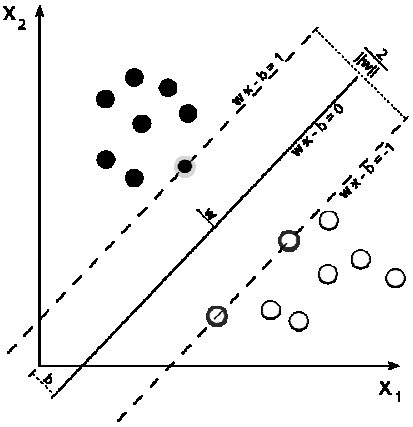

Figure 1. Hyperplane for a SVM trained with two classes. Samples on the margin are support vectors.

The SVM's major advantage lies with their ability to map variables onto an extremely high feature space.

Bayesian classifiers operate by using the Bayes theorem, saying that: Let *X* be the data record (case) whose class label is unknown. Let *H* be some hypothesis, such as "data record *X* belongs to a specified class *C*." For classification, we want to determine *P(H|X)* - the probability that the hypothesis *H* holds, given the observed data record *X*. *P(H|X)* is the posterior probability of *H* conditioned on *X*. Similarly, *P(X|H)* is posterior probability of *X* conditioned on *H*. *P(X)* is the prior probability

of *X*. Bayes theorem is useful in that it provides a way of calculating the posterior probability, *P(H|X)*, from *P(H), P(X)*, and *P(X|H)*. The Bayes theorem is

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)} \tag{4}$$

A difficulty arises when we have more than a few variables and classes - we would require an enormous number of records to estimate these probabilities. *Naive Bayes* (NB) classification gets around this problem by not requiring that we have lots of observations for each possible combination of the variables. In other words, NB classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. Studies comparing classification algorithms have found the NB to be comparable in performance with classification trees and with neural network classifiers. They have also exhibited high accuracy and speed when applied to large databases.

A *decision tree* (DT) is a formalism for expressing mappings between attributes and their classes. It is made up of nodes that are linked to two or more sub-trees, and leaves or end-nodes that are the ultimate decision. DT are praised for their transparency in decision making. A path from the root to a leaf node is essentially a decision rule, or classification rule. There are two stages to building a decision tree - growing and pruning. In the growing stage, the dataset is partitioned recursively until either every record that is associated with each leaf node has the same class, or else the record's cardinality is below a specific threshold value. Pruning the tree involves using a validation sample to essentially cut off the branches lower down in the tree. There are a number of recognised algorithms for building DT, among which ID3, and its upgrade, C4.5. Both have a statistical grounding. ID3 uses information gain to ensure that the best splitting is achieved. The information gain of an attribute can be formally defined as:

$$Gain(S, A) = Ent(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Ent(S_v) \tag{5}$$

where S_v is a subset of S; A has the value *v,* and $|S|$ is the size of S. Whichever attribute A gives the greatest gain is the attribute that should be used. The information gain can therefore be used as a ranking mechanism, where the attribute with the greatest gain not yet considered in the path through the decision tree is at each node. Decision trees, while extremely simple to understand, even to the untrained eye, remain very popular in data mining and classification for that very reason.

3 Results and Discussion

For the purposes of data pre-processing, model building, and analysis, we used tools such as R, Keel, and Weka. The primary source for model estimation is the confusion matrix (a.k.a. contingency table), illustrated in Figure 2.

Results from experiments were summarized in four categories: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The

numbers along the primary diagonal in the matrix represent correct predictions, as long as those outside the diagonal represent the errors.

Table 1 summarises experiment results and reveals that the three algorithms perform differently predicting the cases. Using these summations, a number of measures can be derived, namely precision, recall, specificity, and accuracy. Summary of the results are presented in Table 2. The *precision* is the percentage of positive predictions that are correct, i.e. TP/(TP+FP). The precision of the DT classifier is 58.333%, SVM provides 78.3% precision, but NB precision is 0% - the it performs very poorly having no true positives.

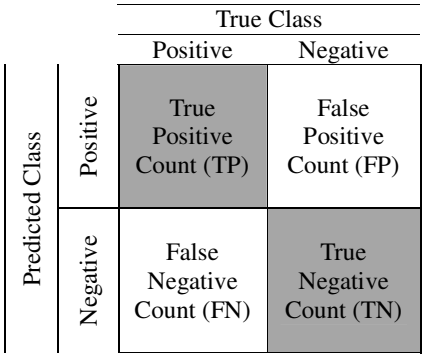

Figure 2. Confusion matrix.

Classifier	*TP*	*FN*	*FP*	*TN*
DT	21	37	15	83
NB	0	57	2	97
SVM	36	22	10	88

Table 1. Confusion matrix values - summary.

Classifier	*Precision*	*Recall*	*Specificity*	*Accuracy*
DT	58.33%	36.22%	84.67%	66.67%
NB	n.a.	n.a.	97.99%	62.10%
SVM	78.30%	62.00%	89.78%	79.40%

Table 2. Performance metrics - summary.

Recall (a.k.a. sensitivity) is TP/(TP+FN), and shows how good the classifier is at picking out instances of a particular class. Once again, the SVM has the best recall rate of 62%, whilst the NB performs poorly once again. The DT has a moderare recall rate of 36.22%. However, as the number of negative samples heavily outweighs the number of positive samples to begin with, the next measure, specificity, may actually give a better indication of how well each classifier is performing. *Specificity*, which is TN/(TN+FP) is in fact the inverse of the recall. In this measure, the NB outperforms both of the other classifiers. *Accuracy* is probably the most intuitive of all of the performance measures, it uses all of the values in the confusion matrix: (TP+TN)/(TP+TN+FP+FN). In this case, the NB has once again finished bottom of the three, with an accuracy of 62.1%. The DT, whilst slightly better than the NB, still has a relatively poor accuracy rate. While 67% may seem decent to some, if one considers a concerned patient presenting with symptoms and about to undergo thoracic surgery, 67% certainty is not confident enough. On the other hand, the SVM has good accuracy of 79.4%.

In conclusion, it is evident that the SVM classifier was the most consistent throughout, scoring very well on three of the metrics and acceptable on the fourth. DT was the next best, as even though it has poor recall and fairly poor precision, it at least had figures for these two metrics, unlike NB.

4 Conclusion

The goal of this research was to analyse several data mining techniques in search of discovering their strengths and weaknesses, dealing with an imbalanced dataset of thoracic surgery patient details. We aimed to identify a method that would perform with a high degree of accuracy in order to provide basis for future work on improving the model performance and tweak it to be transferable to similar datasets. Three very different classifiers were explored in detail: naive Bayes, decision trees, and support vector machines. Each of them manifested specific benefits and drawbacks. The SVM was deemed to be most suited, but its nature implies that changing any one of the input criteria can cause a big change. The imbalance in the classes does no allow it to obtain good margins in the hyper-plane; however, further pre-processing or expansion of the algorithm would improve the overall classification performance. While neither naive Bayes nor decision trees proved up to the task of classifying this particular dataset with a high accuracy, it is also probable that with further algorithmic expansion, their accuracy could be improved.

References

[1] M. Zieba, J. Tomczak, M. Lubicz, & J. Swiatek, Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. *Applied Soft Computing,* vol. 14 (2013), 99-108.

[2] M. Shapiro, S.J. Swanson, C.D. Wright, C. Chin, S. Sheng, J. Wisnivesky, T.S. Weiser, Predictors of major morbidity and mortality after pneumonectomy utilizing the society for thoracic surgeons general thoracic surgery database, *Annals of Thoracic Surgery*, vol. 90 (2010) 927–935.

[3] P. Icard, M. Heyndrickx, L. Guetti, F. Galateau-Salle, P. Rosat, J.P. Le Rochais, J.L. Hanouz, Morbidity, mortality and survival after 110 consecutive bilobectomies over 12 years, *Interactive Cardiovascular and Thoracic Surgery* vol. 16 (2013) 179–185.

[4] P.E. Falcoz, M. Conti, L. Brouchet, S. Chocron, M. Puyraveau, M. Mercier, J.P. Etievent, M. Dahan, The thoracic surgery scoring system (thoracoscore): risk model for in-hospital death in patients requiring thoracic surgery, *The Journal of Thoracic and Cardiovascular Surgery* vol. 133 (2007) 325–332.

[5] A. Barua, S.D. Handagala, L. Socci, B. Barua, M. Malik, N. Johnstone, Accuracy of two scoring systems for risk stratification in thoracic surgery, *Interactive Cardiovascular and Thoracic Surgery* vol. 14 (2012) 556–559.