

**MODELLING THE RELATIONSHIP BETWEEN PROBLEM
CHARACTERISTICS AND DATA MINING ALGORITHM
PERFORMANCE USING NEURAL NETWORKS**

KATE A. SMITH AND FREDERICK WOO

School of Business Systems, Monash University, Victoria 3800, Australia

VIC CIESIELSKI AND REMZI IBRAHIM

Department of Computer Science, Royal Melbourne Institute of Technology

ABSTRACT

As more data mining algorithms become available, the answer to one question becomes increasingly important: which techniques are best suited to which data mining problems? This study attempts to address this question by examining the performance of six leading data mining algorithms across a collection of 57 well-known classification problems from the machine learning repository. For each data set, a number of statistics are collected in order to measure the size and complexity of the problem. Using a neural network, the performance results of the six algorithms are then combined with the problem characteristics to build a predictive model of the relative performance of each algorithm on a given problem.

INTRODUCTION

Classification problems find significance in many practical applications such as credit and risk assessment, medical diagnosis, fraud detection, and quality control. A large number of techniques have emerged over many years for solving such problems, from the early methods of logistic regression through to more modern techniques such as neural networks and decision trees that belong to the data mining repertoire. Many of these techniques have been evaluated on benchmark data sets such as the collection of classification problems at University of California, Irvine [1]. As more techniques become available for solving classification problems however, it becomes increasingly important to know which techniques are suited to which types of classification problems [2]. Of even more relevance for data mining practitioners it to know in advance which technique might perform best on a given problem, in order to eliminate the need for evaluating all techniques to find the best model.

This paper describes the results of our efforts to address this issue. We have examined 57 well-known classification problems from the UCI Repository, and attempted to measure the complexity and characteristics of each problem by considering three types of

measures: simple, statistical, and information theoretic as suggested by Henery [3]. These measures are described briefly in the next section of the paper. For each problem we also evaluate the performance of six popular data mining algorithms. The six data mining algorithms are:

1. IBK, an implementation of a k-means nearest neighbour classifier [4]
2. the decision tree algorithm C4.5 [5]
3. PART, a classifier that generates a decision list [6]
4. Naïve Bayes (NB) [7]
5. OneR [8]
6. KD, a kernel density estimator [9]

We then use a neural network to model the relationship between the characteristics of each data set and the obtained performance of the six data mining algorithms. In this way, we aim to develop a predictive tool to pre-determine the likely performance of each data mining algorithm on a given problem, that can be rapidly evaluated via a single pass through a trained neural network. Similar approaches have been proposed using rule generation methods such as C4.5 rather than a neural network [2,10]. Apart from the different modelling approach used here, the data sets are different, as are the data mining algorithms.

MEASURING COMPLEXITY OF DATA SETS

Each data set can be described by a number of simple, statistical and information theoretical measures. Many of the following measures are described in [3].

Simple and Statistical Measures

Some simple measures are shown in Table 1. In addition, a number of statistical measures can be used for continuous variables as described below:

<i>Measure</i>	<i>Notation</i>
Number of variables	p
Number of instances	N
Number of classes	q
Percentage of discrete variables	disc
Percentage of continuous variables	cont
Percentage of missing values	%_missing

Table 1: Simple measures for characterisation of each data set

Standard deviation ratio (SD_ratio)

$$SD_ratio = \exp \left\{ \frac{M}{p \sum_{i=1}^q (n_i - 1)} \right\} \quad \text{where } M = \gamma \sum_{i=1}^q (n_i - 1) \log |S_i^{-1} S|$$

and

$$\gamma = 1 - \frac{2p^2 + 3p - 1}{6(p+1)(q-1)} \left\{ \sum_{i=1}^q \frac{1}{n_i - 1} - \frac{1}{n - q} \right\}$$

and S_i and S are the unbiased estimators of the i -th sample covariance matrix and pooled covariance matrix respectively. n_i is the number of instances in a particular class i .

Mean correlation between variables (correl)

$$correl = \frac{\sum_{i,j} abs(\rho_{ij})}{\text{Total no. of correlation coefficients}}$$

where ρ_{ij} is the correlation between variables i and j .

Max % and Min% in a class (MaxC, MinC)

Measures the percentage of the number of instances belonging to the least common class (MinC) and the most common class (MaxC).

Mean Absolute Skewness (γ)

$$skew = \frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \bar{x}}{s} \right)^3$$

Mean Absolute Kurtosis (β)

$$\beta_i = \left\{ \frac{(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{s} \right)^4 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$$

First and Second canonical correlation (cancor1, cancor2)

Measures the correlation of two canonical (latent) variables, one representing a set of independent variables, the other a set of dependent variables.

First eigenvalue and Second eigenvalue (fract1, fract2)

The eigenvalues are approximately equal to the squares of the canonical correlations. They reflect the proportion of variance explained by each canonical correlation relating two sets of variables, when there is more than one extracted canonical correlation.

Information theoretical measures

Mean entropy of variables

Entropy is a measure of randomness in a variable. The entropy $H(X)$ of a discrete random variable X is calculated in terms of q_i (the probability that X takes on the i -th value). We average the entropy over all the variables and take this as a global measure of entropy of the variables:

$$H(X) = -\sum q_i \log q_i \quad \bar{H}(X) = p^{-1} \sum H(X_i)$$

Entropy of classes

$$H(C) = -\sum_i \pi_i \log \pi_i$$

where π_i is the prior probability for class A_i

Mean mutual entropy of class and attributes

A measure of common information or entropy shared between the two variables. If p_{ij} denotes the joint probability of observing class A_i and the j -th value of variable X , if the marginal probability of class A_i is π_i , and if the marginal probability of variable X taking on its j -th value is q_j , then the mutual information and its mean over all variables are defined as:

Equivalent number of variables (EN)

$$M(C, X) = \sum_{ij} p_{ij} \log \left(\frac{p_{ij}}{\pi_i q_j} \right) \quad \text{and} \quad \bar{M}(C, X) = p^{-1} \sum_i M(C, X_i)$$
$$EN = \frac{\bar{H}(C)}{\bar{M}(C, X)}$$

This is the ratio between the class entropy and the average mutual information.

Noise-signal ratio (NS)

A large NS ratio implies a data set contains much irrelevant information.

$$NS.ratio = \frac{\bar{H}(X) - \bar{M}(C, X)}{\bar{M}(C, X)}$$

MODELLING ALGORITHM PERFORMANCE

For each of the 57 data sets, we thus have a total of 21 measures that describe the characteristics of the data. The performance of each of the six data mining algorithms has been measured on these 57 data sets using the java program *Weka*, a data mining environment developed at the University of Waikato, New Zealand (<http://www.cs.waikato.ac.nz>). The average error on the test set (randomly extracted to be 20% of the original data) is used as the performance measure. We then post-process the

errors of each algorithm for a given problem, to measure their relative performance, with the best performing algorithm measuring 1, and the worst measuring 0. Thus the result of the j th algorithm on the i th data set is calculated as:

$$R_{ij} = 1 - \frac{(e_{ij} - \min(\mathbf{e}_i))}{\max(\mathbf{e}_i) - \min(\mathbf{e}_i)}$$

where e_{ij} is the test set error for the j th algorithm on data set i , and \mathbf{e}_i is a vector of errors for data set i .

We can now model the relationship between problem characteristics (up to 21 inputs) and relative algorithm performance (6 outputs) using a neural network trained with the 57 data sets. A number of different neural network architectures, choice of inputs, and parameter combinations were experimented with in the Neuroshell2 environment to arrive at the best model, which was a probabilistic neural network. 20% of the data was randomly extracted into a test set to determine the performance of each model. The inputs that resulted in the best model were:

p, q, Bin, Con, %_missing, MinC, MaxC, SDratio, correl, cancel1, cancel2, fract2, M(C,X), H(C), EnAtr, NSRatio.

Table 2 shows the classification accuracy obtained by the neural network model. The overall accuracy when predicting which technique will be best for a given problem is 77% (100% accuracy on the randomly extracted test set which has 11 examples). The accuracy increases to 95% if the top two techniques are inspected, and 98% if the top three techniques are inspected for the best performing technique. Figure 1 shows these relationships.

	IBK	C4.5	PART	NB	OneR	KD
Actual winners:	9	15	3	18	5	7
Classified winners:	2	7	9	21	5	13
% accuracy	22%	47%	100%	100%	100%	100%

Table 2: Accuracy of neural network in predicting best algorithm

CONCLUSIONS

In this paper we have demonstrated the merits of training a neural network to predict data mining algorithm performance. For a given problem, the data characteristics can be fed to the neural network as inputs, and the output is a ranked list of techniques predicting their likely performance on the problem. We have shown that when the top two predicted techniques are inspected, the accuracy is 95% in identifying the best performing algorithm. The performance of the (small) test set was 100%. Thus the user need only test one or two techniques to find a suitable model, rather than the large number of techniques available to the data mining community that the predictive model is trained on.

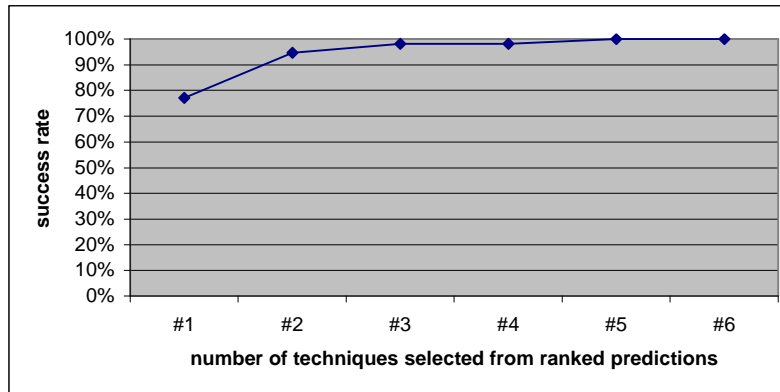


Figure 1: Success rate in finding best performing algorithm based on neural network predictions

REFERENCES

- [1] D. Aha, Machine Learning Database, University of California, Irvine, <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>
- [2] P. B. Brazdil and R. J. Henery, "Analysis of Results", in D. Michie, D. J. Spiegelhalter and C.C. Taylor (eds.), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Limited, Chapter 10, 1994.
- [3] R. J. Henery, "Methods for Comparison", in in D. Michie, D. J. Spiegelhalter and C.C. Taylor (eds.), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood Limited, Chapter 7, 1994.
- [4] D. Aha, and D. Kibler, "Instance-based learning algorithms", *Machine Learning*, vol.6, pp. 37-66, 1991.
- [5] R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [6] E. Frank and I. H. Witten, "Generating Accurate Rule Sets Without Global Optimization". In Shavlik, J., ed., *Machine Learning: Proceedings of the Fifteenth International Conference*, Morgan Kaufmann Publishers, San Francisco, CA, 1998.
- [7] G. H. John and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers". *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. pp. 338-345. Morgan Kaufmann, San Mateo, 1995.
- [8] R.C. Holte, "Very simple classification rules perform well on most commonly used datasets". *Machine Learning*, Vol. 11, pp. 63-91, 1993.
- [9] Silverman, B. W., *Density estimation for statistics and data analysis*, Chapman and Hall, New York, 1986.
- [10] D. Aha, "Generalizing case studies: a case study", *Proceedings of the 9th International Conference on Machine Learning*, pp. 1-10, 1992.