STATISTICS IN MEDICINE
Statist. Med. **17**, 2501–2508 (1998)

A COMPARISON OF STATISTICAL LEARNING METHODS ON THE GUSTO DATABASE

MARGUERITE ENNIS[1], GEOFFREY HINTON[2], DAVID NAYLOR[3], MIKE REVOW[2] AND ROBERT TIBSHIRANI[4]*

[1] *Department of Statistics, University of Toronto, Canada*
[2] *Department of Computer Science, University of Toronto, Canada*
[3] *Institute for Clinical and Evaluative Sciences, Canada*
[4] *Department of Public Health Sciences, and Department of Statistics, University of Toronto, Canada*

SUMMARY

We apply a battery of modern, adaptive non-linear learning methods to a large real database of cardiac patient data. We use each method to predict 30 day mortality from a large number of potential risk factors, and we compare their performances. We find that none of the methods could outperform a relatively simple logistic regression model previously developed for this problem. © 1998 John Wiley & Sons, Ltd.

1. INTRODUCTION

In this study we apply a variety of modern learning methods to the GUSTO-I database. The GUSTO-1 trial was a four-arm randomized study that compared the efficacy of different intravenous thrombolytic regimens for acute myocardial infarction. Acute myocardial infarction (or 'heart attack') is caused by the formation of a clot in one of the coronary arteries that supply blood to the heart muscle. Thrombolytic drugs work by breaking up the clot and restoring blood flow through the artery. Earlier studies had shown that a new and more expensive thrombolytic drug – tissue plasminogen activator [tPA] – restored blood flow more quickly and more often than alternative drug regimens. The study hypothesis was accordingly that the tPA arm would show a 1 per cent absolute short-term (30-day) mortality advantage over any or all of the other arms. Treatments in the other arms included: streptokinase, an older and less expensive thrombolytic drug, which was given with two different regimens of heparin (a drug that helps keep the coronary artery open after the initial break-up of the clot by a thrombolytic drug), and a combination of tPA and streptokinase. The trial involved 41,021 patients admitted to 1081 hospitals in 15 countries, and the study hypothesis was confirmed (the GUSTO-1 Investigators[1]).

In our analysis, we focus on risk factors for the binary outcome 30-day mortality. Lee *et al.*[2] fit linear logistic regression models to these data and reported some success in accurately predicting mortality. In this paper we apply to this problem some recently developed adaptive methods for

* Correspondence to: Robert Tibshirani, Department of Public Health Sciences, University of Toronto, McMurrich Building, Toronto, Ontario, M5S 1A8 Canada. E-mail: tibs@utstat.toronto.edu

CCC 0277–6715/98/212501–08$17.50

Received May 1997
Accepted January 1998

prediction, specifically neural networks, classification trees, generalized additive models (GAM) and multivariate adaptive regression splines (MARS). Descriptions of these techniques appear in many books, for example, Ripley.[3] Some of these methods are designed for very large data sets and such data sets are still relatively rare in statistical applications. The GUSTO-I trial data provided a valuable opportunity to apply the methods on a large problem and to compare their performances.

2. METHODS

There were a total of 2851 patients (7 per cent) who died within 30 days. Although the complete database contains over 100 predictor variables, we decided to use the same variables that Lee *et al.* included in their final model, after consulting a specialist who confirmed that there are few other variables of clinical interest. The predictors were:

1. DIABETES – yes or no
2. PREVMI – previous myocardial infarction 1 = yes
3. PREVCVD – previous cardiovascular disease 1 = yes
4. PREVCABG – previous coronary artery bypass graft surgery 1 = yes
5. PREVPTCA – previous angioplasty 1 = yes
6. SEX
7. HTN – hypertension
8. TX – treatment four groups: tPA = 1; streptokinase and intravenous heparin = 2; streptokinase and tPA = 3; streptokinase + subcutaneous heparin = 4
9. SMKD – smoking status 1 = current, 2 = former, 3 = never
10. MILOCC – part of the heart muscle affected by the blocked artery, coded3–5 unordered 3 = anterior, 4 = inferior, 5 = other (including posterior, lateral, apical), 6 = no MI, Z = missing
11. KILLIPB – killip class (see definition below) coded I = 3, II = 4, III = 5, IV = 6
12. AGE
13. HEIGHT
14. PULSE – heart rate
15. SYSBP – systolic blood pressure
16. TTTRTTM – time (hrs) from the arrival in the emergency department to treatment with a thrombolytic drug.
17. WEIGHT

Lee *et al.*'s logistic regression model contained linear terms for AGE, SBP, KILLIPB, PULSE, MILOCC, PREVMNI, HEIGHT, TTRTTM, SMKD, DIABETES, WEIGHT, PREVCABG, TX, HTN, and PREVCVD; a piecewise linear spline for PULSE, a cubic spline for HEIGHT, and a product interaction between AGE and KILLIPB.

The data set has no missing values since Lee *et al.* used an imputation procedure to fill in any missing explanatory variables. Categorical variables were changed into dummy variables for use in logistic regression, MARS and GAM. As in the paper, systolic blood pressure was upper truncated at 120 and time to treatment lower truncated at two hours.

We randomly divided the data set into two parts: two-thirds of the data form the training set ($n = 27220$), and the rest the test set ($n = 13610$). We used the training set for model development. Where we had to choose parameters in model development, they were determined by dividing the

training set into a smaller training set ($n = 18147$) and a validation set ($n = 9073$). We determined all these subsets once by random draw and all procedures used the same training and test sets thereafter. The test set was used only for the final assessment of the model. After fitting a model via the training set, we found the predicted probabilities p_i of death for the test set observations from that model. Comparisons were made in terms of the log-likelihood for the test set $\sum \ln (p_i^{y_i}/(1 - p_i)^{1 - y_i})$ where $y_i \in \{0, 1\}$ indicates the observed outcomes. We also looked at the area under the ROC curve obtained for the test set.

2.1. Logistic regression

We fitted four different logistic regression models to the training set. Model 1 has covariates for AGE and KILLIPB only while model 2 has covariates for age, Killip class and interactions between age and Killip class. Model 3 has all the covariates in Lee *et al.*'s model, but no interactions and no non-linear (spline) terms. Model 4 has all the covariates, interactions and spline terms exactly as in Lee's model. As above, we estimated the parameters in these models using just the training set.

2.2. Generalized additive models

We built a generalized additive logistic regression model using the S-plus routine 'gam'. This model is a non-linear generalization of the usual linear logistic model that uses smooth spline functions in place of linear risk terms (see Hastie and Tibshirani[5]). The model we used contained linear terms for all the dummy variables, while we entered the variables for age, height, weight, pulse rate, systolic blood pressure and time to treatment as smoothing splines with 4 degrees of freedom. (The value 4 was fixed *a priori*). We did not use any interaction terms.

2.3. Classification trees

We grew a classification tree using the S-plus routine 'tree' on the smaller training set. Classification trees stratify the population in a binary tree form, and are especially good at finding interactions between risk factors (Breiman *et al.*[6]). We found the cost-complexity parameter that minimized the misclassification rate on the validation set. We then pruned the tree using this cost-complexity parameter, reducing the number of terminal nodes from 168 to 31. We obtained predictions for the test set by running the test set covariates down the pruned tree.

2.4 MARS

We fit MARS models of degree 1 (additive) and 17 (all interactions allowed) to the smaller training set with different penalty settings ranging between 0 and 15. MARS stands for 'multivariate additive regression splines', and is a kind of hybrid between generalized additive models and classification tree (Friedman[7]). It is designed to find low-order additive structure as well as interactions between risk factors. Since MARS is designed for regression rather than classification, we first fit MARS regression models with a 0 –1 response. Then we converted each model into a logistic model by using the model matrix produced by the MARS procedure as the design matrix for logistic regression. From these logistic MARS models corresponding to the different penalty values, we picked the one that maximized the validation set log-likelihood. The degree 1 model was maximized at a penalty setting of 1 and the full degree model at 16.

Table I. Test results for various models. Model 1 has AGE and KILLIPB; model 2 has AGE, KILLIPB and interactions between AGE and KILLIPB; model 3 has all the covariates in Lee *et al.*'s model, but no interactions and no non-linear (spline) terms; model 4 has all the covariates, interactions and spline terms exactly as in Lee *et al.*'s model; models 5, 6, 7 have access to all of the predictors; model 5 allows additive terms, while model 6 permits interactions of any order

Model	Log-likelihood	ROC curve area
1 Logistic	− 2939·5	0·787
2 Logistic	− 2930·5	0·788
3 Logistic	− 2791·8	0·818
4 Logistic	− 2785·0	0·820
5 MARS-1	− 2797·1	0·817
6 MARS-full	− 2872·3	0·810
7 Tree	− 3028·9	0·752
8 GAM	− 2789·6	0·819

3. RESULTS

We evaluated the models developed by means of the training set on the test set. Summary results appear in Table I. None of the methods outperformed Lee *et al.*'s model (4), either in log-likelihood or area under the ROC curve. The no-interaction models 3, 5 and 8 come very close to the best performance. In view of this, it is not surprising that the least additive, model 7, performed worst.

4. BACK-PROPAGATION NEURAL NETS

We experimented with fitting multi-layer neural networks to the training data using the back-propagation algorithm of Rumelhart *et al.*,[8] in which one fits the unknown weight parameters using a gradient search method. In this setting, a neural network is a non-linear generalization of the linear logistic model; it essentially replaces the raw risk factors by adaptively chosen non-linear functions of linear combinations of risk factors.

The networks we used had one or more layers of sigmoidal units with a single sigmoidal output unit. The networks were all feedforward, so that there were no backward connections from higher layers to lower layers. We interpret the output as estimating the posterior probability of death, and so the loss function minimized during training was log probability of the data.

The first issue to resolve when using a neural network is to choose the complexity, that is, the number of weights in the network. Once one has fixed the architecture, means must be sought to control the bias-variance trade-off during training. To obtain an estimate of an upper bound on the network complexity we noted that each of the 18, 147 training examples has about 0·35 bits of information.* Assuming it takes about 3 bits per weight, this means that we can afford to have

* There is about a 7 per cent mortality rate in the data. Thus the entropy of the output is $0{\cdot}07 \log 0{\cdot}07 + (1 - 0{\cdot}07) \times \log(1 - 0{\cdot}07) = 0{\cdot}25$ nats or about 0·35 bits per training case

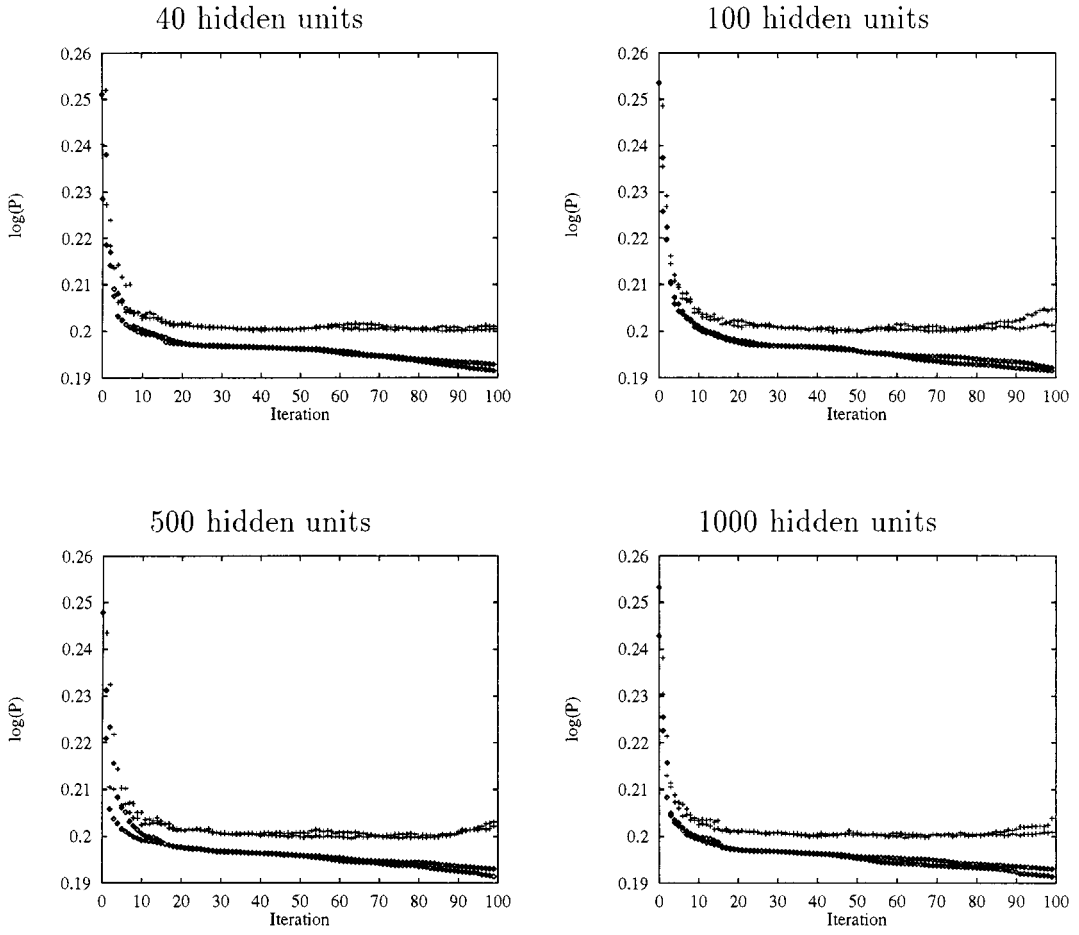

Figure 1. Performance (log P) of neural networks with a single layer of hidden units on the training (lower data points) and validation sets (upper data points) as a function of conjugate gradient epoch number. Results starting from two different points in weight space are shown for each architecture. Performance is similar for all networks. All networks show evidence of overfitting

about 2000 weights in the network. After coding the categorical variables, there are 21 inputs. This suggests a network with 100 hidden units. We also experimented with much larger networks (up to 1000 hidden units) and smaller nets. We also tried nets with more than one hidden layer. Figure 1 shows the results of this exploratory analysis. From these data, it was clear that the size of the network was not crucial and so we selected nets with a single hidden layer containing 100 units (*net100*) and one with two hidden layers with 20 and 10 units (*net20_10*) for further work. In all cases we allowed only feedforward connections, including direct connection from the input to the output unit. Including bias connections, *net100* has 2322 connections and *net20_10* has 682 connections.

The conventional way of controlling the bias-variance trade-off in neural nets is to use *weight decay* or *early stopping*. Another effective way is *bagging* (Breiman[9]). This method reduces

Table II. Test results using bags on the two neural net architectures. The top row gives the average log probability per test case, the second row is the area under the ROC curve of the net's prediction while the last row gives bagged-ROC areas

Parameter	*net100*	*net20_10*	*netARD*
log (P) (bits)	0·205	0·205	0·205
ROC	0·817	0·816	0·815
bagged-ROC	0·816	0·816	

variance in the output by averaging predictions from multiple networks trained on B bootstrap samples of the training data. In this work we used $B = 15$ bootstrap samples. For each sample, the network weights were initialized randomly from a uniform distribution in the range $[-0{\cdot}3 : 0{\cdot}3]$. After each conjugate gradient weight update, we evaluated the performance on the validation data and we retained the best performance. We attempted a maximum of 50 iterations, but the minimum tended to occur after 25–30 iterations. Table II summarizes the test results.

Weight decay is most conveniently viewed as estimating the posterior probability of the network weights given that the weights have a prior zero mean Gaussian distribution. For training data $\mathbf{T}$ and weights $\mathbf{w}$ this leads to the objective function for network $\mathcal{N}$

$$E = \log P(\mathbf{D}|\mathbf{w}, \mathcal{N}) + \sum_{\mathrm{i}} \lambda_{\mathrm{i}} w_{\mathrm{i}}^2. \tag{1}$$

Conventional weight decay assumes that the prior distribution on the weight is spherical and so uses the very crude assumption $\lambda = \lambda_i$, but has the desired property that one needs to estimate only a single parameter. The obvious drawback is that it is probably a poor model of the weight distribution. For example, a weight connecting an irrelevant input to the output should have a much tighter distribution around zero than a weight that is connected to a highly correlated input to the output. This is the basic idea behind automatic relevance detection (ARD) (Neal[10]), in which weights are grouped and given the same λ parameter. For example, it makes sense to collect all outgoing weights from an input unit to a hidden layer in a single group and those to the output unit in another group. Choosing values for the λ_i is a difficult problem. Neal[10] has suggested a technique based on Monte Carlo sampling, while MacKay[11] has suggested a simpler method based on a Gaussian approximation. The weight decay parameter for group j is estimated from

$$\lambda_j = \gamma_j / \sum_i w_i^2 \tag{2}$$

where γ_j is the number of good parameter measurements (MacKay[11]). We tested MacKay's Gaussian approximation on a network with a single hidden layer with 20 units *netARD*. We grouped together all outgoing weights from an input to the hidden units and gave them a common λ. Similarly we allotted a λ to the hidden to output weights and bias to hidden weights. Thus the network has 482 weights and 23 weight decay groups. We initialized all groups with a small amount of weight decay, $\lambda_j = 0{\cdot}1$. The network was trained by doing λ updates (equation

Table III. Weight decay factors associated with each of the input-hidden weights

Input	Weight decay
KILLIP II	4·67
KILLIP III	1·59
KILLIP IV	1·83
Age	5·30
Previous MI	2·33
Previous cardiac vascular disease	2·10
Previous cardiac arterior bypass grafting	1·60
Hypertension	3·90
Strep + IV hep	3·30
Strep + tPA	2·72
Strep + SQ-hep	2·33
Former smoker	4·48
Never smoked	3·62
MI loc interior	5·42
MI loc other	4·01
Pulse	12·29
Systolic BP	14·22
Time to treatment	22·49
Height	14·40
Weight	11·14
Diabetes	2·69

(2)) after every two conjugate updates of the network weights. We repeated this 25 times. The performance of the resulting network is listed in Table II.

A purported advantage to ARD is that it can indicate which units in the net are most important and which are irrelevant for predicting the output. Weights with low values of λ_j are more relevant than weights with large values. Table III lists the weight decay factors associated with weights from each of the inputs. The smallest λ_i are associated with Killip class.

Again, none of these methods outperformed the model of Lee *et al.*

5. DISCUSSION

We found it surprising that with such a large data set, none of the adaptive non-linear methods that we tried could outperform the logistic regression model of Lee *et al.* Below we discuss some possible explanations for this.

(i) *Use of the entire data set*. Lee *et al.* developed their model on the entire data set of 40,830 patients, rather than on just a smaller training set. Since they do not give an objective algorithm for how they chose the model, it is impossible to quantify the possible bias that this introduces into our comparisons. However the fact that their model is of relatively low complexity makes it less likely that this bias is significant.

(ii) *Low predictability*. The predictive power of all of the methods here is quite low. Although the ROC area of the best model is fairly high (82 per cent), the per cent deviance (twice log-likelihood) of the constant model it explains is only about 20 per cent. We suspect that

adaptive non-linear methods are most useful in problems with high signal-to-noise ratio, sometimes occurring in engineering and physical science. In human medical studies, the signal-to-noise ratio is often quite low (as it is here), and hence the modern methods may have less to offer.

(iii) *KILLIP class*. This composite health measure is made up of many other predictors used here, and it was the strongest factor in Lee *et al*.'s logistic model. Hence its presence in the model leaves less of an opportunity for adventurous modelling of these predictors.

Generalizations about the predictive performance of adaptive non-linear algorithms versus more standard statistical techniques must be made with caution. Specific data sets will be more or less advantageous for each method. However, it is noteworthy that adaptive non-linear methods offered no advantages in this large and rich data set. Comparatively few clinical data sets will have over 40,000 subjects characterized as extensively as is the case in the GUSTO-1 trial. Thus, our findings add evidence to support those who have suggested that adaptive non-linear algorithms might have limited applicability in clinical settings.

ACKNOWLEDGEMENTS

We thank an editor and referees for helpful comments.

REFERENCES

1. GUSTO-1 Investigators. 'An international randomized trial comparing four thrombolytic strategies for acute myocardial infarction', *New England Journal of Medicine*, 673–682 (1993).
2. Lee, K., Woodleif, L., Topol, E., Weaver, D., Betriu, A., Col, J., Simoons, M., Aylward, P., Van de Werf, F. and Califf, R. 'Predictors of 30-day mortality in the era of reperfusion for acute mycocardial infarction', *Circulation*, 1659–1668 (1995).
3. Ripley, B. D. *Pattern Recognition and Neural Networks – a Statistical Approach*, Cambridge University Press, 1995.
4. Killip, T. and Kimbal, J. T. 'Treatment of myocardial infarction in a coronary care unit. A two year experience with 250 patients', *American Journal of Cardiology*, 457–464 (1967).
5. Hastie, T. and Tibshirani, R. *Generalized Additive Models*, Chapman and Hall, 1990.
6. Breiman, L., Friedman, J. H., Olshen, R. and Stone, C. J. *Classification and Regression Trees*, Wadsworth, 1984.
7. Friedman, J. H. 'Multivariate adaptive regression splines (with discussion)', *Annals of Statistics*, **19**, (1), 1–141 (1991).
8. Rumelhart, D. E., Hinton, G. E. and Williams, R. J. 'Learning internal representations by error propagation', *in* McClelland, J. L., Rumelhart, D. E. and the PDP research group (eds), *Parallel Distributed Processing*: *Explorations in the Microstructure of Cognition*, volume I, Bradford Books, Cambridge, MA, 1986.
9. Breiman, L. 'Bagging predictors', *Machine Learning*, 123–140 (1996).
10. Neal, R. M. *Bayesian Learning for Neural Networks*, Springer-Verlag, 1996.
11. MacKay, D. J. C. 'Bayesian interpolation', *Neural Computation*, **4**, 415–447 (1992).

