The Truth is in There - Rule Extraction from Opaque Models Using Genetic Programming

Ulf Johansson
Department of Business and Informatics
University of Borås
Sweden
ulf.johansson@hb.se

Rikard König
Department of Business and Informatics
University of Borås
Sweden
rikard.konig@hb.se

Lars Niklasson
Department of Computer Science
University of Skövde
Sweden
lars.niklasson@ida.his.se

Abstract

A common problem when using complicated models for prediction and classification is that the complexity of the model entails that it is hard, or impossible, to interpret. For some scenarios this might not be a limitation, since the priority is the accuracy of the model. In other situations the limitations might be severe, since additional aspects are important to consider; e.g. comprehensibility or scalability of the model. In this study we show how the gap between accuracy and other aspects can be bridged by using a rule extraction method (termed G-REX) based on genetic programming. The extraction method is evaluated against the five criteria accuracy, comprehensibility, fidelity, scalability and generality. It is also shown how G-REX can create novel representation languages; here regression trees and fuzzy rules. The problem used is a data-mining problem from the marketing domain where the impact of advertising is predicted from investment plans. Several experiments, covering both regression and classification tasks, are evaluated. Results show that G-REX in general is capable of extracting both accurate and comprehensible representations, thus allowing high performance also in domains where comprehensibility is of essence.

Introduction

For the data-mining domain the lack of explanation facilities seems to be a very serious drawback for techniques producing opaque models, for example neural networks. Experience from the field of Expert System has shown that an explanation capability is a vital function provided by symbolic AI systems. In particular the ability to generate even limited explanations is absolutely crucial for the user acceptance of such systems (Davis, Buchanan and Shortliffe, 1977). Since the purpose of most data mining systems is to support decision making the need for explanation facilities in these systems is apparent. Nevertheless many systems (especially those using neural network techniques but also ensemble methods like boosting) are normally regarded as black boxes; i.e. they are opaque to the user.

Background

Andrews, Diederich and Tickle (1995) highlight the deficiency of artificial neural networks (ANNs), and argue for rule extraction; i.e. to create more transparent representations from trained ANNs:

> It is becoming increasingly apparent that the absence of an explanation capability in ANN systems limits the realizations of the full potential of such systems, and it is this precise deficiency that the rule extraction process seeks to reduce. (page 374)

It should be noted that an explanation facility also offers a way to determine data quality, since it makes it possible to examine and interpret the relationships found. If the discovered relationships are found doubtful when inspected, they are less likely to actually add value. The task for the data miner is thus to identify the complex but general relationships that are likely to carry over to the production set, and the explanation facility makes this easier.

Rule extraction from trained neural networks

The knowledge acquired by an ANN during training is encoded as the architecture and the weights. The task to extract explanations from the network is therefore to interpret, in a comprehensible form, the knowledge represented by the architecture and the weights.

Craven and Shavlik (1997) coined the term *representation language* for the language used to describe the model learned by the network. Craven and Shavlik also used the expression *extraction strategy* for the process of transforming the trained network into the new representation language. Representation languages used include (if-then) inference rules, M-of-N rules, fuzzy rules, decision trees and finite-state automata.

There are basically two fundamentally different approaches to rule extraction; decompositional (open box or white box) and pedagogical (black box).

Decompositional approaches focus on extracting rules at the level of individual units within the trained ANN; i.e. the view of the underlying ANN is one of transparency.

Pedagogical approaches treat the trained ANN as a black box; i.e. the view of the underlying ANN is opaque. The core idea in the pedagogical approach is to treat the ANN as an oracle and view the rule extraction as a learning task where the target concept is the function learnt by the ANN. Hence the rules extracted map inputs to outputs. Black-box techniques typically use some symbolic learning algorithm where the ANN is used to generate the training examples.

Evaluation of rule extraction algorithms. Craven and Shavlik (1999) list five criteria to evaluate rule extraction algorithms:

- Comprehensibility: The extent to which extracted representations are humanly comprehensible.
- Fidelity: The extent to which extracted representations accurately model the networks from which they were extracted.
- Accuracy: The ability of extracted representations to accurately predict unseen examples.
- Scalability: The ability of the method to scale to networks with large input spaces and large numbers of weighted connections.
- Generality: The extent to which the method requires special training regimes or restrictions on network architectures.

Most researchers have evaluated their rule extraction methods using the first three criteria but, according to Craven and Shavlik, scalability and generality have often been overlooked. In the paper they define scalability as:

> Scalability refers to how the running time of a rule extraction algorithm and the comprehensibility of its extracted models vary as a function of such factors as network, feature-set and training-set size. (page 2)

Craven and Shavlik reason that models that scale well in terms of running time, but not in terms of comprehensibility will be of little use. It should be noted that scaling is an inherent problem, regarding both running time and comprehensibility, for decompositional methods. The potential size of a rule for a unit with n inputs each having k possible values is k^n, meaning that a straightforward search for rules is impossible for larger networks.

Craven and Shavlik proposed that rule extraction researchers should pursue new directions to overcome the problem of scalability, e.g.:

- Methods for controlling the comprehensibility/fidelity trade-off; i.e. the possibility to improve the comprehensibility of an extracted rule set by compromising on its fidelity and accuracy.
- Methods for anytime rule extraction; i.e. the ability to interrupt the rule extraction at any time and then get the best solution found up to that point.

Regarding generality, Craven and Shavlik argue that rule extraction algorithms must exhibit a high level of generality to become widely accepted. In particular, algorithms requiring specific training regimes or algorithms limited to narrow architectural classes are deemed less interesting. Ultimately rule extraction algorithms should be so general that the models they extract from need not even be neural networks. Obviously there is also a need to explain complex models like ensembles or classifiers using boosting, so it is natural to extend the task of rule extraction to operate on these models.

Predicting the impact of advertising

The ability to predict the effects of investments in advertising is important for all companies using advertising to attract customers.

In the media analysis domain the focus traditionally has been to explain the effect of previous investments. The methods are often based on linear models and have low predictive power. However, it is also important to identify differences between expected outcome and actual outcome. In cases where there is a substantial difference, efforts have to be made to identify the cause. This is the reason why it is important to generate models, which show good predictive performance on typical data. It is thus assumed that historical data for a product contain information about its individual situation (e.g., how well its marketing campaigns are perceived) and that this could be used to build a predictive model.

The domain. Every week a number of individuals are interviewed to find out if they have seen and remember adverts in different areas (in this case car adverts). From these interviews the following percentages (among others) are produced for each make:

- Top Of Mind (TOM). The make is the first mentioned by the interviewee.
- In Mind (IM). The interviewee mentions the make.

The overall task is to supply a company with useful information about the outcome of its planned media investment strategy. This task is normally divided into two sub-tasks: a monthly prediction (with updates every week) and a long-term forecast, covering approximately one year.

Related work

Johansson and Niklasson (2001) showed, for the car domain, that the performance of the neural network approach clearly surpasses the linear approaches traditionally used, and that it is the temporal ability rather than the non-linearity that increases the performance.

The fact that the results for the ANNs were significantly better than the standard method actually used made the neural network approach interesting enough to exploit further. At the same time the ability to present the model learned by the network in a more transparent notation was identified as a key property for the technique to be used as a tool for decision-making.

Johansson and Niklasson (2002) used the trained ANNs as a basis for finding a model transparent enough to enable decision-making. More specifically, the rule extraction method TREPAN (Craven and Shavlik, 1996) was used to create decision trees from the trained ANNs. Since TREPAN performs classification only, the original problem had to be reformulated into predicting if the effect for a certain week exceeded a specific limit. The limit chosen (with the motivation that it represents a "good week") was the 66-percentile of the training set.

The main result was that the decision trees extracted had higher performance on unseen data than the trees created directly from the data set, by the standard tool "See5" (Quinlan, 1998). The complexity of the extracted representations was comparable to that of the trees generated by See5.

Nevertheless the trees created by TREPAN were still rather complicated. Since smaller (less complex) trees would make it easier for decision-makers to grasp the underlying relationships in the data, Johansson, König and Niklasson (2003) suggested a novel method called G-REX[1], for rule extraction. G-REX is based on genetic programming and was tested on both well-known classification problems and the "impact of advertising" problem. The extracted rules from G-REX generally outperformed both TREPAN and See5 regarding both accuracy and comprehensibility.

The G-REX algorithm. The extraction strategy adopted by G-REX includes the use of GP on trained ANNs. This approach incorporates the demands on the extracted representation into the strategy itself, which is a key concept.

When using G-REX on a specific problem fitness function, function set and terminal set must be chosen. The function and terminal sets determine the representation language, while the fitness function captures what should be optimized in the extracted representation.

Obviously there is a direct connection between the formulation of the fitness and the evolved programs. This is a nice property for the task of rule extraction since the exact choice of what to optimize in the rule set is transferred into the formulation of the fitness function. This function could for example include how faithful the rules are to the ANN (fidelity), how compact the rules are (comprehensibility) and how well they perform on a validation set (accuracy).

[1] Genetic-RuleEXtraction

Method

The overall purpose of this study is to evaluate G-REX on new tasks and using new representation languages. More specifically G-REX will be extended to handle:

- Regression problems producing regression trees.
- Classification problems producing fuzzy rules.

In addition G-REX will use not only ANNs to extract from, but also another opaque model; i.e. boosted decision trees.

The study is a comparative one where the results from G-REX are compared both to the original results (from the opaque model) and to the results from standard techniques. The standard techniques are the default selections for the respective problem category in the data-mining tool Clementine[2]. For classification tasks this is (boosted) decision trees using the C5.0[3] algorithm. For regression tasks the technique is C&R-T.

The problems and data used

Two variations of the "impact of advertising" problem are used. In both experiments TOM and IM are predicted from investments in different media categories. 100 weeks are used for training and the test set consists of 50 weeks. To reduce the number of input variables only four aggregate variables are used:

- TV: money spent on TV-commercials.
- MP: money spent on advertising in morning press.
- OP: money spent on advertising in other press; i.e. evening press, popular press and special interest press.
- OI: money spent in other media; i.e. radio, outdoor, movie.

The two main experiments are:

- A long-term (one year) regression forecast. This is very similar to the original experiments used by Johansson and Niklasson (2001). The main difference is the aggregation of input variables.
- A short-term (one month) prediction using classification. This is similar to the experiments conducted by Johansson et. al. (2003), but now the horizon is one month instead of just one week. This is an important difference since some variables, shown to be very important (e.g. *share-of-voice*) will not be available.

[2] www.spss.com/spssbi/clementine

[3] C 5.0 is called "See 5" on the Windows platform.

Only four car brands (Volvo, Ford, Hyundai and Toyota) are used in the experiments. Previous studies have produced good results on these data sets.

Long-term regression forecast

The purpose of this experiment is to produce a long-term forecast covering approximately one year. Each input tuple consists of investments during the current week and also from four lagged weeks. The overall problem is thus to predict effects of advertising from sequences of investments. Three approaches are evaluated:

ANNs. The ANNs are standard multi-layered perceptions (MLPs) with one hidden layer. Initial experimentation using a validation set found 8 hidden neurons to be sufficient. For each effect (e.g. TOM for Ford) five ANNs are trained and the prediction is the average of those nets.

C&R-Trees in Clementine. Here the standard method for producing regression trees in Clementine is invoked. It should be noted that the technique termed C&R-Trees, according to the documentation, is a comprehensive implementation of the methods described as CART© (Breiman et. al., 1984).

G-REX. To enable a fair comparison with C&R-Trees G-REX uses a functional set consisting only of relational operators and an *if-statement*. The terminal set consists of the input variables and random constants in a suitable range. Using these function and terminal sets the feasible expressions are exactly the same for G-REX and C&R-Trees. G-REX uses the results of the trained ANN as fitness cases; i.e. the fitness is based on fidelity. In addition a penalty term is applied to longer representations, thus enforcing more compact trees.

Short-term prediction using classification

The purpose of this experiment is to produce a short-term prediction on a horizon of four weeks.

The original regression problem is transformed into a binary classification problem where the task is to predict whether the effect (TOM or IM) will be higher than the 66-percentile representing a "good week". In addition to the input variables used in the long-term forecast the variable *previous effect* (PE) is introduced. PE is the targeted effect for previous weeks. Obviously this would be available when predicting on short horizons. PE is an important indicator for trends; i.e. detecting when the ratio between investments and effects changes. The task here is to predict an effect four weeks ahead using the investments between now and that week, together with previous effects from between the current week and two weeks back. In this experiment five different approaches are evaluated:

ANNs. The ANNs are standard MLPs with one hidden layer. Initial experimentation using a validation set found 5 hidden units to be sufficient. There is just one output unit and the two classes are coded as –1 and +1. An output over 0 from the ANN represents a predicted class of HIGH (good week). For each effect eleven ANNs are trained and the prediction is the average of those nets.

C5.0. Both single decision trees and boosted trees created by C5.0 are evaluated.

G-REX extracting Boolean rules from ANNs. The function set consists of relational operators and logical operators (AND, OR). The terminal set contains the input variables and random constants. An extracted representation is a Boolean rule. The fitness function is based on fidelity towards the ANN and a penalty term to enforce short rules.

G-REX extracting Boolean rules from boosted decision trees. The only difference from the previous experiment is that the fitness uses fidelity towards the boosted trees.

G-REX extracting fuzzy rules from ANNs. In this experiment the extracted rule is a fuzzy rule. Each input variable has been manually fuzzified and has two possible fuzzy values, labeled Low and High. Fig.1 shows how the fuzzification was performed. The constants *a* and *b* were, for each variable, chosen as the 20-percentile and the 80-percentile of the training data.

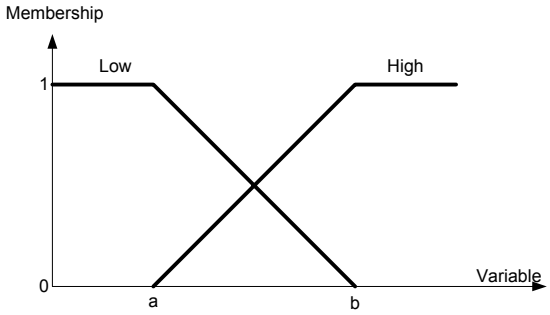

Fig. 1: Fuzzification.

The terminal set contains the input variables and the names of the fuzzy sets. The function set now contains logical operators, *hedges (very* and *rather*) and the function *is*. If μ_A is the membership mapping function for the fuzzy set A and μ_B is the membership mapping function for the fuzzy set B, then the logical operators, working on fuzzy variables, are defined like:

$$\mu_{A \text{ AND } B}(x) = \mu_A(x) \wedge \mu_B(x) = \min \{\mu_A(x), \mu_B(x)\}$$
$$\mu_{A \text{ OR } B}(x) = \mu_A(x) \vee \mu_B(x) = \max \{\mu_A(x), \mu_B(x)\}$$

Hedges serve as modifiers of fuzzy values. In this experiment the two hedges *very* and *rather*, as defined below, are used.

$$\textbf{\textit{very:}}\ \mu_A'(x) = \mu_A(x)^2 \qquad \textbf{\textit{rather:}}\ \mu_A'(x) = \sqrt{\mu_A(x)}$$

To produce a prediction the output from the fuzzy rule is compared to a threshold value, which is also evolved for each candidate rule.

Results

Table 1 shows the results for the regression task. The results are given as coefficient of determination (R^2), between predicted values and target values on the test set.

	TOM			IM		
	ANN	C&R-T	G-REX	ANN	C&R-T	G-REX
Volvo	0.78	0.37	0.60	0.81	0.60	0.80
Ford	0.75	0.48	0.60	0.62	0.44	0.61
Toyota	0.73	0.35	0.58	0.75	0.44	0.61
Hyundai	0.64	0.66	0.65	0.67	0.64	0.67
MEAN	**0.73**	**0.47**	**0.61**	**0.71**	**0.53**	**0.67**

Table 1: Results for the regression task.

Fig. 2 and Fig. 3. show predictions from the ANN and G-REX, plotted against the target values.

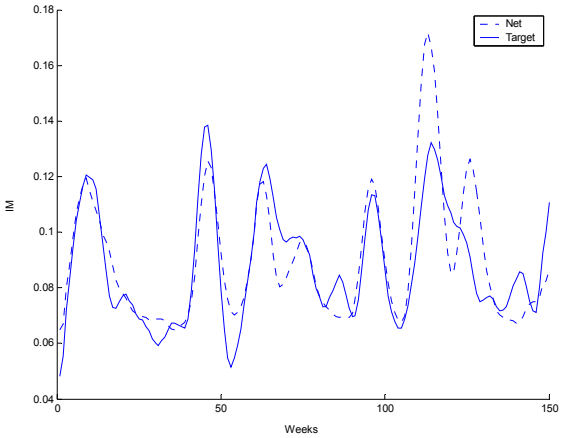

Fig. 2: ANN prediction for Ford IM. Training and test set.

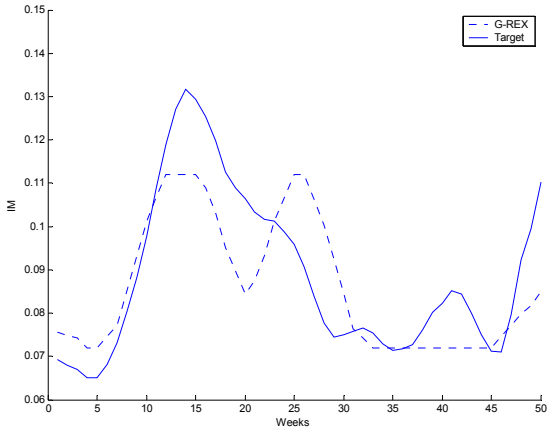

Fig. 3: G-REX prediction for Ford IM. Test set only.

A sample evolved S-expression is shown in Fig. 4 below:

```
(if  (< TV0 17 )
     (if (< TV2 36 )
       (if (> OI1 40 ) 82 72)
       (if (< TV2 97 ) 83 97 ) )
     (if (> OP1 216 )
       (if (< TV2 97 ) 85 112 )
       (if (< TV2 40 ) 85 112 ) ) )
```

Fig. 4: Evolved regression tree for Ford IM.

Table 2 and Table 3 show the results from the classification experiments as percent correct on the test set.

	ANN	C5.0	C5.0 boost	G-REX ANN	G-REX C5.0 boost	G-REX fuzzy
Volvo	92%	66%	72%	92%	72%	92%
Ford	80%	82%	78%	80%	78%	82%
Toyota	80%	66%	72%	72%	72%	76%
Hyundai	74%	34%	46%	94%	50%	90%
MEAN	**82%**	**62%**	**67%**	**85%**	**68%**	**85%**

Table 2: Results for the classification task (TOM).

	ANN	C5.0	C5.0 boost	G-REX ANN	G-REX C5.0 boost	G-REX fuzzy
Volvo	90%	74%	74%	90%	72%	88%
Ford	78%	72%	76%	80%	70%	82%
Toyota	84%	72%	82%	80%	78%	82%
Hyundai	84%	62%	74%	84%	72%	84%
MEAN	**84%**	**70%**	**77%**	**84%**	**73%**	**84%**

Table 3: Results for the classification task (IM).

Most of the extracted rules are both accurate and very compact. Fig. 5 and Fig. 6 show sample Boolean and fuzzy rules extracted by G-REX.

```
(AND(OR ( > Prev0 10558)( > TV0 10596))
    (AND( > TV1 9320   )( > TV0 933  )) )
```

Fig. 5: Evolved Boolean rule for Toyota IM (good week).

```
(AND(TV0 is rather high PE0 is very high))
```

Fig. 6: Evolved fuzzy rule for Ford IM (good week).

Discussion

In this section G-REX is evaluated against the criteria proposed by Craven and Shavlik.

Accuracy. G-REX performs well in this study. Most importantly the accuracy on test sets is normally almost as good as that of the underlying ANN. Regarding accuracy G-REX outperforms standard tools like C 5.0 and C&R-T.

Comprehensibility. Craven and Shavlik specifically stress "methods for controlling the comprehensibility/fidelity trade-off" as an important part of rule extraction algorithms. The possibility to dictate this tradeoff by the choice of fitness function consequently is a key property of G-REX.

At the same time the experiments show that, for the data sets investigated, G-REX is often capable of coming up with a short *and* accurate rule. As a matter of fact for most problems studied G-REX performs just as well when forced to look for short rules.

Another important aspect of the G-REX algorithm is the possibility to use different representation languages. In this study Boolean rules, fuzzy rules and regression trees were created just by changing the function and terminal sets.

Fidelity. Although this is not the main purpose of the G-REX algorithm the study show that the extracted representations have very similar performance to the ANNs, both on training and test sets. Obviously G-REX, especially when forced to look for short rules, is not capable of representing all the complexity of an ANN. With this in mind, it is a fair presumption that G-REX is capable of finding the general relationship between input and output, represented by the ANN.

Scalability. When it comes to scalability, black-box approaches in general have an advantage compared to open-box methods. Black-box approaches obviously are independent of the exact architecture of the ANN, which is in sharp contrast to open-box methods. Thus the size of the input space and the number of data points are the interesting parameters when considering the scalability of a black box approach.

Still it should be recognized that GP (and therefore G-REX) is computationally expensive. It should also be noted that G-REX has not yet been tested on a really large data set. There is no reason to believe that G-REX will not perform well on larger data sets, but it remains to be verified.

GP also inherently has the ability of "anytime rule extraction" since evolution can be aborted at any time to produce the best rule found up to that point.

Generality. G-REX is very general since it operates on a data set disregarding things like architecture, training regimes etc. As seen in this study G-REX does not even require the underlying application to be a neural network. G-REX can be used equally well on, for instance, boosted decision trees or ensembles combining different classifiers. G-REX also proved feasible not only on classification tasks but also on regression tasks.

Conclusions

The purpose of this study has been to evaluate the versatility of the genetic programming rule extraction algorithm G-REX, against the five criteria identified by Craven and Shavlik (1999). The results show that G-REX not only exhibits a high degree of accuracy, but also that this accuracy does not necessarily come on the expense of comprehensibility. Fidelity and scalability have not been prioritized in this study.

Regarding generality G-REX is very versatile since it acts on data sets and not the actual underlying architectures. G-REX can be applied to many different types of models and generate a multitude of representations. This is demonstrated here by having G-REX produce regression trees and fuzzy rules in addition to Boolean rules and decision trees.

The conclusion is that we might be closer to a general purpose tool for knowledge extraction from opaque models.

References

R. Andrews, J. Diederich and A. B. Tickle 1995. A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. *Knowledge-Based Systems*, 8(6).

L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone 1994. *Classification and Regression Trees*, Wadsworth International Group.

M. Craven and J. Shavlik 1996. Extracting Tree-Structured Representations of Trained Networks. *Advances in Neural Information Processing Systems*, 8, pp. 24-30.

M. Craven and J. Shavlik 1997. Using Neural Networks for Data Mining. *Future Generation Computer Systems: special issue on Data Mining*, pp. 211-229.

M. Craven and J. Shavlik 1999. Rule Extraction: Where Do We Go from Here? *University of Wisconsin Machine Learning Research Group working paper 99-1*.

R. Davis, B. G. Buchanan and E. Shortliffe 1977. Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, Vol 8, No 1, pp 15-45.

U. Johansson and L. Niklasson 2001. Predicting the Impact of Advertising - a Neural Network Approach. *Proc. The International Joint Conference on Neural Networks*, IEEE Press, Washington D.C., pp. 1799-1804.

U. Johansson and L. Niklasson 2002. Neural Networks - from Prediction to Explanation. *Proc. IASTED International Conference Artificial Intelligence and Applications*, IASTED, Malaga, Spain, pp. 93-98.

U. Johansson, R. König and L. Niklasson 2003. Rule Extraction from Trained Neural Networks using Genetic Programming. *13th International Conference on Artificial Neural Networks,* Istanbul, Turkey, 2003, supplementary proceedings pp.13-16.

U. Johansson, C. Sönströd, R. König and L. Niklasson 2003. Neural Networks and Rule Extraction for Prediction and Explanation in the Marketing Domain. *Proc. The International Joint Conference on Neural Networks,* IEEE Press, Portland, OR, 2003, pp. 2866-2871.

J. R. Quinlan, See5 version 1.16, http://www.rulequest.com, 1998.