Rule extraction from trained neural networks & connectionist knowledge representation for the determination of pesticide mixtures

Ubbo Visser*
Richi Nayak, Man To Wong†

ABSTRACT

The Department of Computer Science in Agriculture at the University of Munster, Germany (DCSA-UMG) has developed and implemented the plant protection advisory system PRO_PLANT . This has been carried out in cooperation with the Department for Plant Protection, Seed testing and Agriculture Research (DPSAR) (a government and farmer aided institution in Westfalia). The system was financially supported by the Ministry for Environment, Regional Planing and Agriculture of North Rhine Westfalia. It runs offline on a farmers PC on the Windows platform. It is a knowledge-based system, which places fungicide- and growth regulator-consultations for cereals as well as insecticide- and herbicide-consultations in canola and corn at the farmer's disposal (Visser & Voges, 1994). A multilayer-feedforward network is part of this system and is used to determine herbicide mixtures in an actual situation.

The methods and models concerning Artificial Neural Networks (ANN) have been continously developed and improved. Among other things, several rule extraction algorithms have been applied to real-world-problems. Recently, we have completed a study which is described in Visser et al. (1996). We have investigated different rule extraction techniques such as RuleNeg, RULEX (Andrews et al., 1995) and LAP (Hayward et al., 1996) for extracting the knowledge embedded in the trained ANN as a set of symbolic rules. We compared the results with the symbolic induction algorithm OSL (Orlowski, 1993).

The current paper describes techniques, which are able to map the outcome of decompositional rule extraction algorithms to a higher abstraction layer. We generate a knowledge base that consists of generic predicate rules based on the outcome of rule extraction algorithms. This enables the user to interact with the knowledge base.

As the well known machine learning technique C4.5 (Quinlan, 1993) is a de facto standard, we apply this algorithm directly to the data to compare the investigations we have made in the previous study. This process is independent from the above mentioned investigations.

The overall objective is to determine whether all these techniques are able to support and improve the development of systems which solves real-world-problems.

1. Introduction

A high input of pesticides in farming has led to a growing contamination of ground and surface water in Europe. To improve this situation, the farmers have to reduce the input of pesticides to a minimum, which gives as good economic returns as well as high input routine sprays if the application of pesticides will be optimised. Important conditions are crop management to avoid diseases and country wide plant disease control. Therefore, the DSCA-UMG has developed and implemented the plant protection advisory system PRO_PLANT in cooperation with the DPSAR. The system was financially supported by the Ministry for Environment, Regional Planing and Agriculture of North Rhine Westfalia. The system runs offline on the farmer' s PC on the Windows platform. The decision support system PRO_PLANT based on scientific discoveries in the areas phytomedicine and phytopathology as well as practical experiences from plant-protection advisors and farmers. The system is build for pest control against fungal deseases and growth regulators in cereals and for pest control against autumn- and springpests in winter canola as well as for a selection of herbicides in corn (Voges & Visser, 1995).

The amount of herbicides in corn, available on

*Centre of Computing Technology, University of Bremen, Universitatsallee 22, 28344 Bremen, Germany, Email: visser@informatik.uni-bremen.de

†Neurocomputing Research Centre, Queensland University of Technology, GPO Box 2434 , Brisbane 4001, Qld, Australia, Email: {nayak|mwong}@fit.qut.edu.au

the market, has dramatically increased after the ban of Atrazin. Spectrum of effectivness, mechanism of effectivness, amount of active substance, application conditions and the possibility to produce different mixtures of the herbicides are not uniform and quite special. Additionally, the number and sort of corn-weeds has an enormous variety in terms of density, soil type and soil sort as well as the weather conditions. The existing domain knowledge has been collected for the advisory system and has placed to the customers disposal with the help from modern computer science methods. A part of the systems 'herbicide in corn' has been implemented with the help of a neural network. Weeds which have frequent occurrence on light soils (sand) and heavy soils (loam) are gathered to so called 'weed-societies' and are presented to a multilayer-feedforward network. The outcome of the network is a list of applicable herbicide-mixtures. This system has been intensively tested in practical use in the vegetation period 93/94 and 94/95 by plant protection advisors and farmers. As problem solver in this case we use a kind of competition method (Puppe, 1990) in contrast to the rest of the system. In order to get the selection of herbicides the system first is seeking for a known case which fits to the current illness situation. If the fitness of the current situation is not significant, a complex database inquiry will be startet (Visser & Voges, 1994).

2. Applied Techniques

2.1. Symbolic induction

C4.5 (Quinlan, 1993) derives the production rules from the decision tree generated from a set of training cases. A set of training cases is refined into subsets of cases that are, or seem to be heading towards, single-class collections of cases. A test is chosen, based on a single attribute, that has one or more mutually exclusive outcomes. The decision tree consists of a decision node identifying the test, and one branch for each possible outcome. The same tree building machinery is applied recursively to each subset of training cases. The original tree classifier (ID3) used a criterion called gain criterion. This criterion has a deficiency - it has a strong bias in favor of tests with many outcomes. To avoid this gain ratio criterion is used. To generate production rules from decision trees, the following steps are used:

- Every path from the root of an unpruned tree to a leaf gives one initial rule. The left hand side of the rule contains all the conditions established by the path, and the right hand side specifies the class at the leaf.
- Each such rule is simplified by removing conditions that do not seem helpful for discriminating the nominated class from the other classes, using a pessimistic estimate of the accuracy of the rule.
- For each class in turn, all the simplified rules for that class are sifted to remove rules that do not contribute to the accuracy of the set of rules as a whole.
- The set of rules for the classes are then ordered to minimize false positive errors and a default class is chosen.

The above process leads to a production rule classifier that is usually about as accurate as a pruned tree, but more easily understood by people.

2.2. Rule extraction

In most applications build with ANN it is absolutely crucial to explain how and why a decision was determined. There are several cases when there is a statutory requirement (e.g. credit applications) and in other cases the explanation of the result is part of the application (e.g. diagnosis in medicine and environmental technique). Additionally, it is highly desirable that internal states are interpretable, e.g. to prove the reliability of the system (transparency). These along with other considerations such as the topic 'explanation of connectionist systems', have discovered the interest on rule extraction algorithms which extract symbolic rules out of trained neural networks.

A universal demand to a rule extraction technique is efficiency. In order to pass the combinatoric problems, which can arise due to the evaluation of all possible combinations of income links to a node, most of the techniques are using heuristic methods. Transparent approaches e.g. are using bias nodes in order to filter significant inputs. An equally important evaluation criterium is the "quality" of the rules (Towell & Shavlik, 1993). Rule-quality comprise

- accuracy,
- fidelity and
- comprehensibility.

A set of rules created by a rule extraction algorithm is *accurate* if unknown patterns are classified correctly. *Fidelity* is present if the set of rules imitates the behaviour of the ANN perfectly. *Comprehensibility* can be measured with the help of three attributes; *number of rules*, *number of antecedents per rule* and *consistency*. A set of rules is consistent if a network with different parameters creates a set of rules which classifies unknown pattern correctly and consistently.

Andrews and Geva (1995) have developed a rule extraction technique called RULEX. This technique is based on a particular type of multilayer perceptron, the constrained error back-propagation network which is, regarding classification and func-

Table: 1: Experimental results

	RULEX	RuleNeg	OSL	C4.5
Learning-Error	5%	0%	-	-
No of rules	49	60	67	15
No of antecedents	323	278	414	65
Av. no of antec.	6.59	4.63	6.18	4.3
Accuracy	95%	100%	100%	90%
Fidelity	high	100%	-	-
Consistency	Yes	Yes	-	-

tion approximation, similiar in concept to radial basis function networks, but with local functions constructed from sigmoids not from gaussians.

The outcome of this technique are described in more detail in Andrews et al. (1995). The described method is limited to the creation of conjunctive and propositional rules.

2.3. Rule translation process

To provide a much more detailed explanation of why a particular instance is classified as a target concept, we need to convert the propositional symbolic rules extracted from the trained neural network to quantified rules in form of generic predicates. The generated knowledge base that includes rules and facts and type-hierarchy can be used as input to any inference engine that allows user interaction and enables the greater explanation capability.

The converter module takes the extracted rules from trained neural network as inputs and yields the quantified and generic rules and facts. The rule translation process is explained with the rules extracted using RULEX. RULEX performs the rule-extraction process by the direct interpretation of weight parameters as rules so there is no intermediate concepts, only a set of attributes and a set of output classifiers. The rules format extracted from RULEX decompositional rule extraction technique is as demonstrated in table 3 on the next page.

One of the direct interpretation of weight parameters gained by RULEX for this particular dataset, when the output unit or node have an activation of one is:
Solni(yes) And Steme(yes) And Polco(yes) And Galap(yes) And windh(no)
Let replace the set of attributes by variables like, P denote the set of all values of attribute Solni, Q the set of Steme values, X the set of Polco values, Y the set of Galap values and Z the set of windh values. The ancillary predicate for the above rule inferring that the output unit will fire can be written as:
$\{P, Q, X, Y, Z herb\_mixt\_1\_pred\_0(P, Q, X, Y, Z)\}$ with its associated facts $\{herb\_mixt\_1\_pred\_0(yes, yes, yes, yes, no)\}$ where the name of the predicate is the – name of the target class_predicate_no of fact –. If any rule has more than one value for attribute then it would be written as another predicate with the possible combination of attribute-values.

One of the direct interpretation of weight parameters gained by RULEX for the output unit or node having low output is:
Echog(yes) And Vioar(yes) And Polco(no) And windh(yes)
This is the rule 34 in the Rulex rule-set then this can be represented as $\neg\{P, X, Y, Z herb\_mixt\_1\_pred\_34(P, X, Y, Z)\}$.

A general predicate for goal concept of herb_mixt_type1 can be expressed as $\forall P, \ldots, Z\ herb\_mixt_1(P, \ldots, Z)$. In the similar way general predicate for herb_mixt_type2 can be expressed as $\forall P, \ldots, Z herb\_mixt\_2(P, \ldots, Z)$. To complete the definition of each target class, we need to introduce rules utilizing the definition for each predicate resulting in the rules:
$P, \ldots, Z herb\_mixt\_1\_pred\_0(P, Q, X, Y, Z) \Rightarrow herb\_mixt\_1(P, \ldots, Z)$,
$\neg\{P, \ldots, Z herb\_mixt\_1\_pred\_34(P, X, Y, Z) \Rightarrow herb\_mixt\_1(P, \ldots, Z)\}$

See Nayak et al. (1997) or Hayward et al. (1998) for a more detailed description of the process.

3. Results

The problem is the classification of a data set with 255 records, eight attributes with two features each and 38 output classes. With regards to complex benchmark tests, it is not supposed to be difficult to create a set of rules. The inputs are different combinations of weeds and each output class represents several herbicide mixtures which can be applied against these weeds.

In order to demonstrate the input to the rule translation process that is discussed later in this paper we will give an example. It shows a brief extract of the results from the 1996 study for output class 1 of the rule extraction algorithm RULEX:

Table: 2: Knowledge Base

```
/* IS_A RELATIONSHIPS */
...
is_a(echog_yes, echog)
is_a(echog_no, echog)
is_a(vioar_yes, vioar)
is_a(polco_no, polco)
is_a(windh_yes, windh)
...
/* FACTS */
...
¬herb_mixture_1_pred_34(echog_yes, vioar_yes, polco_no, windh_yes)
¬herb_mixture_2_pred_2(solni_yes, steme_yes, polco_yes, galap_no, windh_no)
...
/* BACKWARD REASONING PREDICATES */
...
¬herb_mixture_1(AA < echog, AB < atxss, AC < solni, AD < steme, AE < vioar, AF < polco, AG < galap,
AH < windh) :-
herb_mixture_1_pred_34(AA, AE, AF, AH)
¬herb_mixture_2(AA < echog, AB < atxss, AC < solni, AD < steme, AE < vioar, AF < polco, AG < galap,
AH < windh) :-
herb_mixture_2_pred_2(AC, AD, AF, AG, AH)
```

Table: 3: Rule example of RULEX

IF	vioar	IS NO
AND	galap	IS YES
AND	solni	IS NO
AND	echog	IS NO
AND	atxss	IS YES
THEN	APPLY	Mixture 1

In other words:

IF	Klettenlabkraut *(Galium aparine)*
AND	Melde *(Atriplex patula L.)*
THEN	herbicide class 1

The output class 1 represents several herbicide mixtures which can be applied against the weeds *Galium aparine* and *Atriplex patula L.* in corn. In particular:

Stentan 4.00l	Zintan Pack 1/2
Gardobuc 1.50l	Lido Pack fluid 1/2
Pendimox 3.00l	Extoll 2.00l
Buctril/Certrol B 1.00l	Duogranol 1.50kg
Banvel 4S 0.50l + Cato 20g	Lentagran 1.70kg

3.1. Comparison C4.5 with previous results

The results determined with the help of C4.5 are discussed with concern to the above described rule quality. The results are shown in table 1.

C4.5 does not need a network training session before creating rules. The algorithm classifies most of the patterns correctly and creates 15 rules with an accuracy of 90%. The number of rules and antecedents per rule is significantely less compare to the applied algorithms carried out in the previous paper. This can be an important issue if the data set is large. In this particular case we may assume that the symbolic induction methods are the best because we do not have to train a network. Also, there is one step less than with the other algorithms.

3.2. Connectionist knowledge representation

The process starts with with training of a supervised ANN. Once the training process is over, the rule extraction method RULEX is used to extract propositional rules. The extracted rules are transformed in a connectionist knowledge base representation that includes a is-a hierarchy, type-token distinction and type-restrictions.

A rule such as the one above is translated into the connectionist knowledge representation demonstrated in table 2. As a matter of fact, RULEX extracted 49 rules out of the ANN. The CKB consists of 16 is_a-relations, 1824 facts and 1824 predicates. This is because of eight atributes with two features each and one positive predicate and 37 negative predicates per mixture. One may assume that comprehensibility decreases due to the number of facts and rules in the knowledge base. However, we are now able to query to system on an 'higher' level. The level of interactivity is even higher than using the propositional rules.

4. Conclusions

In this paper we have investigated two subjects

- C4.5 vs. rule extraction techniques and another symbolic algorithm and
- the transformation of propositional rules as an outcome of RULE into a connectionist knowledge representation.

The objective was to determine whether these techniques are able to support and improve the development of systems which solves real world problems.

In this particular study we come to the conclusion that the C4.5 system produces less rules and less antecedents per rule than OSL and the applied rule extraction techniques applied to an ANN. However, C4.5 has less accuracy than the other techniques but is likely to match even strong criteria as the error is only 10%. We may assume that in this particular case C4.5 is the best technique to apply.

With regards to the connectionist knowledge representation we have to outline that the extracted rules can be generalised and generic rules can be processed by a connectionist reasoning system. This is a step forward in terms of interactivity between user and system. The rules support the type constraint on the arguments. Furthermore, the methodology provides a much more detailed explanation of why a particular instance is classified as a member of a gol concept.

References

[1] R. Andrews, J. Diederich, and A. Tickle. A Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks. *Knowledge-Based Systems*, 8(6):373–389, 1995.

[2] R. Andrews and S. Geva. Inserting and extracting knowledge from constrained error back propagation networks. In *Proc. 6th Australian Conference on Neural Networks*, Sydney, NSW, 1995.

[3] R. Hayward, A. Tickle, and J. Diederich. *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, chapter Extracting Rules for Grammar Recognition from Cascade-2 Networks. Springer-Verlag, Berlin, Feb/Mar 1996.

[4] R. Nayak, R. Hayward, and J. Diederich. Connectionist Knowledge Representation By Extracted Generich Rules from Trained Feed-Forward Neural Networks. In J. Diederich and R. Andrews, editors, *Connectionist Systems for Knowledge Representation and Deduction*, Proceedings of the CADE-14 Workshop, Townsville, Australia, pages 87–98, Brisbane, July 1997. QUT.

[5] M. Orlowski. On the Integration of the Knowledge Bases. In G.E. Lasker, editor, *Advances in Computers and Information Engineering*, Canada, 1993. The International Institute For Advanced Studies in Systems Research and Cybernetics.

[6] F. Puppe. *Problemlosungsmethoden in Expertensystemen.* Studienreihe Informatik. Springer, Berlin, 1990.

[7] J. R. Quinlan. *C4.5:programs for machine learning.* Morgan Kaufman, San Mateo CA, 1993.

[8] G. Towell and J. Shavlik. Extracting refined rules from knowledge-based neural networks. *Machine Learning*, 13(1):71–101, 1993.

[9] U. Visser, A. Tickle, R. Hayward, and R. Andrews. Rule-Extraction from trained neural networks: Different techniques for the determination of herbicides for the plant protection advisory system PRO_PLANT. In J. Diederich and R. Andrews, editors, *Rules and Networks*, Proceedings of the Rule Extraction From Trained Artificial Neural Networks Workshop, Brighton, UK, pages 133–139, Brisbane, April 1996. Society for the Study of Artificial Intelligence and Simulation of Behavior, QUT.

[10] U. Visser, U. Voges, and U. Streit. Integration of AI-, Database- and Telecommunication-Techniques for the Plant Protection Expert System PRO_PLANT. In D.F. Anger, R.V. Rodriguez, and M. Ali, editors, *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, volume 7 of *International Conference of IEA/AIE*, pages 367–374, San Antonio, 31. May - 3. June 1994. Gordon and Breach Science Publishers.

[11] Uwe Voges and Ubbo Visser. PRO_PLANT II – Umfassende Pflanzenschutzberatung und -information. In Bernd Bachmann and Frank Maurer, editors, *Expertensysteme 95. Leistungsschau. 3. Deutsche Expertensystemtagung (XPS-95), Universitat Kaiserslautern, 1.-3. Marz 1995*, pages 61–62, Universitat Kaiserslautern, FB Informatik, AG Expertensysteme. 67653 Kaiserslautern., March 1995.