Eclectic Rule-Extraction from Support Vector Machines

Nahla Barakat, and Joachim Diederich

Abstract**—** Support vector machines (SVMs) have shown superior performance compared to other machine learning techniques, especially in classification problems. Yet one limitation of SVMs is the lack of an explanation capability which is crucial in some applications, e.g. in the medical and security domains. In this paper, a novel approach for *eclectic* rule-extraction from support vector machines is presented. This approach utilizes the knowledge acquired by the SVM and represented in its support vectors as well as the parameters associated with them. The approach includes three stages; training, propositional rule-extraction and rule quality evaluation. Results from four different experiments have demonstrated the value of the approach for extracting comprehensible rules of high accuracy and fidelity.

Keywords**—** Data mining, hybrid rule-extraction algorithms, medical diagnosis, SVMs

I. Introduction

In recent years, support vector machines (SVMs) have shown good performance in a number of application areas. However the learning capability of SVMs comes at a cost: an inherent inability to explain the process by which a learning result was reached. Hence, the situation is similar to artificial neural networks (ANNs) [1],[2], where the apparent lack of an explanation capability has led to various approaches aiming at extracting symbolic rules from neural networks. For SVMs to gain acceptance in areas such as medical diagnosis, it is desirable to offer an "explanation" capability.

A. The Importance of Rule-Extraction Algorithms

The ability of symbolic AI systems to provide a declarative representation of knowledge about the problem domain offers a natural explanation capability for the decisions made by the system. Reference [3] argues that even limited explanation can positively influence the system's acceptance by the user. This capability is important, especially in the case of medical applications. An explanation capability can also provide a check on the internal logic of the system as well as being able to give a novice insight into the problem [4]. In addition, the explanations given by rule-extraction algorithms significantly enhance the capabilities of AI systems to explore data and support the induction and generation of new theories [5].

ANN's & SVMs have no such declarative knowledge structures, and hence, are limited in providing explanations.

B. Background:The Classification of Rule-Extraction Algorithms

One potential method for classifying rule-extraction algorithms is in terms of the "translucency" of the view taken within the rule-extraction method of the underlying classifier. This motif yields two basic categories of rule-extraction techniques: "transparent" and "pedagogical" [1], [2].

The distinguishing characteristic of the "transparent" (or "decompositional") approach is that the focus is on extracting rules at the level of individual components of the underlying machine learning method. In feedforward neural networks, these are hidden and output units.

The classification "pedagogical" or “learning-based” is given to those rule-extraction techniques that treat the underlying classifier as a "black box". Such techniques typically are used in conjunction with a learning algorithm that provides rule-based explanations and the basic motif is to use the trained classifier to generate examples for a second learning algorithm that generates rules as output [6],[7],[8]. A third group in this classification scheme are composites that incorporate elements of both the "transparent" and "pedagogical" rule-extraction techniques. This is the "hybrid" or “eclectic” group [1], [2], [9].

Clearly, this classification scheme, originally developed for rule-extraction from neural networks, is applicable to support vector machines as well. Decompositional approaches can be based on the analysis of support vectors generated by the SVM while learning-based approaches learn what the SVM has learned. An example for learning-based rule-extraction from SVMs is [10].

C. Evaluation of Extracted Rules' Quality

The quality of the extracted rules is very important for rule-extraction techniques. This aspect is different from the other dimensions as it evaluates rule-extraction algorithms at the level of the rules themselves, rather than the level of the rule-extraction algorithm, and is a direct indication on how successful the extraction process is. Four rule-extraction quality criteria were suggested in [1], [2]: rule accuracy,

Manuscript received March 12, 2005.

N. Barakat is a Lecturer in the Faculty of Applied Sciences, Sohar University, Sohar, Oman and a PhD student at the University of Queensland, Brisbane, Australia. (e-mail: n.barakat@ soharuni.edu.om).

J. Diederich, is Dean, Faculty of Applied Sciences, Sohar University, Oman and an Honorary Professor at the University of Queensland, Brisbane, Australia. (e-mail: j.diederich@soharuni.edu.om).

fidelity, consistency, and comprehensibility.

In this context, a rule set is considered to be *accurate* if it can correctly classify previously unseen examples. Similarly a rule set is considered to display a high level of *fidelity* if it can mimic the behavior of the machine learning technique from which it was extracted. An extracted rule set is deemed to be consistent if, under different training sessions, the machine learning technique generates rule sets which produce the same classifications of unseen examples. Finally the *comprehensibility* of a rule set is determined by measuring the size of the rule set (in terms of the number of rules) and the number of antecedents per rule.

D. Problem Overview

Having identified the importance of rule-extraction algorithms, an overview of the problem to be addressed is given here. In artificial neural networks, knowledge acquired during the training phase is encoded in the network architecture, the activation function associated with each (hidden and output) unit of the ANN and a set of weights [2].

Hence, the task of extracting explanations (or rules) from a trained ANN is interpreting in a comprehensible form the collective effect of this encoded knowledge.

In case of support vectors machines, knowledge acquired during the training phase is represented by the model support vectors, and the parameters associated with them. The task of rule-extraction is finding a way to express this knowledge in a comprehensible form.

II. ANN$_S$ and SVM$_S$: Similarities and Differences

A. Internal Representation

Linear ANNs are single-layer networks. Units in an ANN calculate two functions: The input function, normally the weighted sum of inputs with a threshold that can be represented as an additional weight (1), and the output function which calculates the signal sent to other units.

In case of the SVM, linearly separable data should satisfy the constraints in (2) and (3), [12].

$$w_1x_1 + w_2x_2 + ...+ w_ix_i - \mathrm{T} \quad (1)$$

$$x_i.w + b \geq +1 \;\; for \;\; y_i = +1 \qquad (2)$$

$$x_i.w + b \leq -1 \;\; for \;\; y_i = -1 \qquad (3)$$

Where, $\{x_i, y_i\}, \;\; i = 1,...,l, \;\; y_i \in \{-1,1\}, \;\; x_i \in \Re^d$ represent the training data. For the points where (2) holds, data patterns lie on the hyperplane $x_i.w+b=1$, similarly, for the points where (3) holds, patterns lie on the hyperplane $x_i.w + b = -1$ with normal w. Equations (1), (2) and (3) can be interpreted as equations defining a line or a linear hyperplane.

In ANNs non linearity is introduced by hidden layers and non linear activation functions such as sigmoid, radial, or gaussian [11]. In case of, SVMs non linearity is introduced by mapping input vectors $x_i \in \Re^n$ into Z vector of a higher dimensional feature space $\mathrm{F}\,(z = \Phi(x))$, where Φ represents a mapping $\Re^n \rightarrow \Re^F$, (where *F >> n),* to solve a linear classification problem in the feature space. Hence the linear hyperplane in a feature space has an equivalent non- linear decision boundary in input space.

We can conclude that both SVMs *viz* multilayer and RBF ANNs resemble each other in the way they deal with non-linearity [13],[14].

B. Learning

From the computational learning perspective, the problem is to find the best hypothesis given the data set. In case of ANNs, the best hypothesis is the one that minimize the error over a training set, by finding an optimal set of weights for a given number of hidden and output units. Without proper regularisation, overfitting the data set is possible. In contrast, SVMs try to find hypotheses that minimize the true error, hence better approximate the target classification function and overcome the overfitting problem.

From the previous discussion, we can conclude that the rule-extraction classification proposed by [2] can be applied to rule-extraction from SVMs.

III. Related Work: Rule-Extraction from SVMs

A. Decompositional Rule-Extraction from SVMs

Reference [15] introduces an approach for rule-extraction from SVMs: the SVM+ prototype method. The basic idea of this method is to use the output decision function from an SVM and then use K-means clustering to determine prototype vectors for each class. These vectors are combined with support vectors to define an ellipsoid in the input space which are then mapped to if-then rules. This approach does not scale well: in case of a large number of patterns and an overlap between different attributes, the explanation capability suffers.

B. Learning-based Rule-Extraction from SVMs

References [16], [17] suggest a learning-based approach to extract rules from SVMs using two different data sets:

1) A labelled data set is used for SVM learning purposes, i.e. to build a model with acceptable accuracy.

2) A second data set is generated with the same attributes but different values to explore the generalisation behaviour of the SVM. That is, the SVM is used to get the class labels for this data set. Hence a synthetic data set is obtained.

3) The synthetic set is then used to train a machine learning technique with explanation capability. Thereby, rules are generated that represent the generalisation behaviour of the SVM.

IV. The Approach: Eclectic Rule-Extraction from SVMs

Our approach makes use of the information provided by the learned model support vectors, which define the separating hyperplane and the parameters associated with them. The approach handles the rule-extraction task in three basic steps, which proceed as follows:

A. Learning Stage

Use labeled patterns to train an SVM and get an SVM model (classifier) with acceptable accuracy, precision, and recall.

B. Rule Generation

The objective of this stage, which proceeds in two steps, is to express the concepts learned by the model in a comprehensible form. The first step utilizes the knowledge offered by support vectors and parameters associated with them, while the second step aims to express that knowledge in a comprehensible form. The steps are:

1) From the training data set, select the patterns that become support vectors, but discard their class label.
2) Use the SVM model to predict the class label of those patterns (support vectors), hence a special synthetic data set is generated.
3) Use the synthetic data set to train a machine learning technique with explanation capability (in this case the C5 decision tree learner is used) [18], hence symbolic rules that represent the concepts learned by the SVM model are generated.

C. Evaluating the Quality of the Extracted Rules

A second (previously unseen) data set is used at this stage to test the quality of the extracted rules in terms of the aspects mentioned in I-*C*. The algorithm used for testing fidelity is shown in Table I

A similar algorithm is used to measure the accuracy of the extracted rules, but the class as predicted by rules ($rule_{class}$) is compared to the target class of the patterns (instead of the class as predicted by the SVMs model (SVM_{class}).

V. The Experiments

Four experiments with four different benchmark data sets (available from the UCI ML repository) [19] have been performed to test the validity of the rule-extraction approach. The details are given in the following paragraphs.

A. Pima Indians Diabetes

A sample of 458 patterns extracted from the original data set is used, after removing all patterns with zero value for the attributes "2-hour OGTT plasma glucose", "diastolic blood pressure" and "triceps skin fold thickness" which are clinically insignificant. Each pattern has 9 attributes; risk factors are described in terms of 8 attributes plus a binary output class (tested positive or negative for diabetes). 247 patterns are used for training (88 positive and 159 negative). The remaining patterns are used for rule quality testing

B. Heart Diseases:

The reduced Cleveland heart diseases data set is used. This data set has 13 risk factors, plus the class, which indicates the diagnosis of heart disease. This data set was also used by a group of previous researchers. 223 patterns are used for training (103 positive patterns and 120 negative patterns). The remaining patterns are used for rule quality testing. All patterns with missing values are discarded.

TABLE I
FIDELITY MEASUREMENT ALGORITHM

```
f = 0
n=1
$r_n \in R$
 $SVM_{class} \in \{-1,1\}$,    $R_{class} \in \{-1,1\}$
     /*$R$ is set of rules. 1< = n < = N the total number of rules*/
For each pattern $i$ in a data set of size I
     Find the class $SVM_{class}$ for $i$ by use of SVM
           repeat for  n >=1 to n<=N
           Find class $R_{class}$ for $i$  by use of $r_n$
           if  $R_{class}$ =  $SVM_{class}$
           then  f++
           else
            n++
          /* end of every rule*/
    /* end of every pattern*/
fidelity= f/I
```

C. Breast Cancer

The Wisconsin breast cancer data set is used. The data set has 9 variables as risk factors, in addition to the class label (0 for benign, 1 for malignant). A random sample of 98 positive examples and 110 negative examples was selected for training. The remaining patterns are used for rule quality testing. All repeated patterns were discarded to avoid the bias resulting from the boosting effect of those patterns.

D. Hepatitis

A data set of 59 patterns is used (11 positive and 48 negative) for training purposes. The remaining patterns are used for rule quality testing. The data set has 20 attributes, starting with the class (no survival 1, survival 0) and 19 risk factors. The attribute No 19 was discarded as it has 67 missing values.

In all experiments the SVM^{light} software [20] is used and the linear SVMs give the best possible results.

VI. Results

Table II demonstrates that the four experiments are consistent in their results. The experiments generated rules of high quality and accuracy. It can also be noted that rules are of good comprehensibility and the number of rules or antecedents does not increase proportionally with the number of support vectors or the number of attributes in the data sets.

Results also show that some rules have better generalization than the SVM model from which they were extracted. This raises again the open question of fidelity versus the accuracy of the rules [21], [22].

TABLE II
EXPERIMENTAL RESULTS

	HEPATITIS	DIABETES	HEART DISEASES	BREAST CANCER
No of Training patterns	59	247	223	208
SVM LOO accuracy	85%	86%	72%	95%
No of model support vectors	15	85	87	19
No of rules/antecedents	1/1	3/3	2/2	3/2
Test set patterns	73	211	77	208
SVM classification accuracy on test set	83%	95%	74%	83%
Rule accuracy	85%	93%	82.5%	82%
Rule fidelity	96%	92%	88%	91%

VII. CONCLUSION

From this work we can conclude that the knowledge acquired by the SVM model can be extracted by making use of the model support vectors and a machine learning technique with explanation capability. It can also be concluded that this approach fits well into the eclectic rule-extraction algorithm family, as it elects the patterns that have influence in defining the separating hyperplane and it has also a pedagogical component. The taxonomy that considers the translucency dimension for rule-extraction originally developed for artificial neural networks is also valid for SVMs (learning-based and eclectic approaches).

VIII. REFERENCES

[1] A.B. Tickle, R.Andrews, M.Golea, and J.Diederich, “The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural network”, *IEEE Trans. Neural Networks*, vol. 9(6), pp. 1057-1068, 1998.

[2] R. Andrews, J. Diederich, and A.B. Tickle, “A Survey and Critique of Techniques For Extracting Rules From Trained Artificial Neural Networks”, *Knowledge Based Systems*, vol. 8, pp. 373-389, 1995.

[3] R. Davis, B.G. Buchanan, and E. Shortcliff, “Production Rules as a Representation for a Knowledge Based Consultation Progra”, *J. Artificial Intelligence*, vol. 8(1), pp.15-45, 1977.

[4] S. Gallant, “Connectionist Expert System”, *Communications of the ACM*, vol. 31 (2), pp. 152-169, 1988.

[5] S. Sestito and T. Dillon, “Automated Knowledge Acquisition of Rules With Continuously Valued Attributes”, *in Proc.12th International Conference on Expert Systems and their Applications (AVIGNON'92)*, Avignon -France, 1992, pp. 645-656.

[6] M.W. Craven, and J.W. Shavlik, “Using Sampling and Queries to Extract Rules From Trained Neural Networks”, *in Proc. of the 11th International Conference on Machine learning*, NJ, 1994, pp.37-45.

[7] G. Towell, and J. Shavlik. "The Extraction of Refined Rules From Knowledge Based Neural Networks”, *J. Machine Learning,* vol. 131, pp.71-101, 1993.

[8] M.W. Craven, and J.W. Shavlik, “Extracting Tree–Structured Representation of Trained Networks”, *Advances in Neural Information Processing Systems*, vol. 8, pp.24-30, 1996.

[9] A. Tickle, A, M. Orlowski, M, J. Diederich, “DEDEC: A Methodology for Extracting Rules from Trained Artificial Neural Networks. ”In: Andrews, R.; Diederich, J. (Eds.): Rules and Networks. Brisbane, Qld.: QUT Publication 1996, 90-102.

[10] R. Mitsdorffer, J. Diederich, and C. Tan, “Rule-extraction from Technology IPOs in the US Stock Market”, presented at ICONIP02, Singapore, 2002.

[11] H. Khuu, H.K. Lee, J-L, Tsai. “ Machine learning with Neural Networks and support vector machines”, University of Wisconsin, unpublished, 2004

[12] C. Burges, *A tutorial on support vector machines for pattern recognition. data mining and knowledge discovery*, Boston, Kluwer Academic publishers, 1998.

[13] V. Kecman, *Learning and Soft Computing*. Cambridge, MA: MIT Press, 2001

[14] V. Kecman, “Learning by Support Vector Machines from Huge Data Sets”, presented at KES 2004, Eighth international conference on knowledge-based intelligent information & engineering systems, 20-24 September, 2004, Wellington, New Zeland.

[15] H. Núñez, C. Angulo, and A.Catala, “Rule-extraction from Support Vector Machines”, *in Proc. of European Symposium on Artificial Neural Networks*, Burges, 2002, pp.107-112.

[16] N. Barakat , and J. Diederich, “Learning-based rule-extraction from support vector machines: Performance on benchmark data sets”: Kasabov, N., Chan, Z.S.H. (Eds.), *in Proc. of the conference on Neuro-Computing and Evolving Intelligence*, Auckland, New Zealand, Auckland. Knowledge Engineering and Discovery Research Institute (KEDRI) (2004).

[17] J. Diederich , and N. Barakat, “Hybrid rule-extraction from support vector machines” *in Proc. of IEEE conference on cybernetics and intelligent systems*, Singapore, 2004, pp. 1270-1275.

[18] http://www.rulequest.com

[19] http://www.ics.uci.edu/~mlearn/MLRepository.html

[20] http://svmlight.joachims.org/

[21] M. Craven and J. Shavlik, **“**Rule Extraction: Where Do We Go from Here?”, Department of Computer Sciences, Machine Learning Research Group Working Paper 99-1, 1999.

[22] A.Tickel, F. Maire, G. Bologna, J. Diederich.” Lessons from past, current issues and future research directions in extracting the knowledge embedded in Artificial Neural Networks”. Lecture notes in computer science*, Hybrid Neural Systems*, vol. 1778, revised papers from a workshop 1998, pp. 226 - 239