Induction of Decision Trees from Trained SVM Models using a TREPAN Based Approach

Douglas E. Torres D.

Decanato de Investigación y Postgrado DIP-UNEFA, Venezuela
Facultad de Ingeniería, Universidad Central de Venezuela
douglastd@cantv.net

Abstract. This paper describes the application of an Hybrid Intelligent System (HIS) to extract decision tress from a trained Support Vector Machine (SVM) model based on the TREPAN algorithm. TREPAN, a well-known technique developed originally to extract linguistic rules from a trained Artificial Neural Network, is modified to cope with SVM models. The proposed approach is tested on five data sets related to the medical domain, with excellent performance results.

1 Introduction

Support Vector Machines are efficient computing models that have shown excellent generalization performance in a variety of applications areas. However they have difficulty in explaining the results due to the lack of explanatory power and therefore are considered as "black-box " models. The same situation occurs with Neural Networks.

An important area of investigation that has been used in diverse application domains to develop interpretable expressions is based on the combination of different intelligent techniques such as neural networks, decision trees, systems based on fuzzy rules, reasoning based on cases, among others [1]. This operative synergy, called Hybrid Intelligent Systems (HIS) [2], seeks to improve the efficiency, reasoning power and comprehensibility of the integrand systems.

This paper presents, under the integrative perspective of HIS, an approach for the extraction of knowledge from SVM models. Several approaches have been proposed recently to obtain human interpretable expression, usually through rule extraction procedures from a trained SVM; see for example the works of Nuñez et al. [3], Fung et al. [4], Barakat and Diederich [5] and Zhanh et al [6]. In our study this knowledge is expressed through decision trees derived using a modification of TREPAN [7], an algorithm originally developed by Craven [8] to extract decision trees from a trained Artificial Neural Network (ANN).

The paper is organized as follows: Sec. 2 describes the SVM classifier. In Sec. 3 the TREPAN approach and its modifications are described, while Sec. 4 compares the results obtained by SVM and the modified TREPAN, on five publicly available data

Received 16/03/09
Accepted 27/04/09
Final version 08/05/09

sets and, finally Sec. 5 presents the conclusions.

2 Support Vector Machine

Support Vector Machines provide a novel approach to the two-category classification problem [9]. The methods have been successfully applied to a number of applications ranging from particle identification, face identification and text categorization to engine detection, bioinformatics and data base marketing. The approach is systematic and properly motivated by statistical learning theory [10].

Suppose we have N training data points $\{(X_1, Y_1),\ldots, (X_N,Y_N)\}$, where X_i, i=1,..., N, is a vector of input variables and Y_i is the corresponding participation decision. Denote with S^{+} (resp. S^{-}) the convex hull of the points X_i with output +1 (resp. output −1). Thus, if S^{+} and S^{-} are linearly separable, we can think of constructing the optimal hyperplane $w \cdot X+b = 0$, which has maximum distance from these two convex hulls. The problem can be mathematically formulated as:

$$\begin{aligned} &\underset{w,b}{\text{Min}}\ \tfrac{1}{2}\, w^{T} w \\ &\text{s.t.}\quad y_i(w \cdot X + b) \geq 1 \end{aligned} \qquad (1)$$

where the quantities w and b are usually referred to as weight vector and bias [11]. This is a convex, quadratic programming problem in the unknowns (w, b). It can be equivalently solved by searching for the values of the Lagrange multipliers α_i in the Wolfe dual problem. In this case we have $w = \Sigma_i\, \alpha_i y_i X_i$.

Only those points, which lie closest to the hyperplane, have $\alpha_i > 0$ and contribute to the above sum. These points are called support vectors and represent the essential information about the training set at hand.

Once we have found the optimal hyperplane, we simply determine on which side of the decision boundary a given test pattern X^* lies and assign the corresponding class label, using the function sgn $(w \cdot X^* + b)$.

If the two convex hulls S^{+} and S^{-} are not linearly separable the optimal hyperplane can still be found by accepting a small number of misclassified points in the training set. A regularization factor C accounts for the trade off between training error and distance from S^{+} and S^{-}.

To adopt non-linear separating surfaces between the two classes, we can project the input vectors X_i into another high dimensional feature space through a proper mapping $\Phi(\cdot)$. If we employ the Wolfe dual problem to retrieve the optimal hyperplane in the projected space, it is not necessary to know the explicit form of the mapping Φ. We only need the inner product $K(X,X') = \Phi(X) \cdot \Phi(X')$, which is usually called kernel function [12]. Different choices for the kernel function have been suggested; they must verify the Mercer's condition [13]; for example, the Gaussian radial basis function kernel: $K(X,X') = \exp(-\gamma\|X-X'\|^2)$.

The need of properly choosing the kernel is a limitation of the support vector approach. In general, the SVM with lower complexity should be selected.

Several studies have recently reported on the application of Support Vector Machines (SVM) applied to medical databases [14,15,16].

3 Hybrid Intelligent Systems Models: The TREPAN algorithm

Hybrid Intelligent Systems are computational systems, which are based mainly on the integration of soft-computing techniques. This integration allows exploring their advantages in order to increase the overall system performance for a given task or to generate comprehensible representation of the knowledge [17]. With regard to the medical applications handled with hybrid intelligent systems, several studies reported in the literature concerning integration of soft-computing techniques as neural network and decision tress [18], evolutionary artificial neural networks [19], hierarchical soft computing [20].

In this paper the Extraction of Knowledge from an SVM model, which allows its validation and refinement, as well as the integration of connectionist and symbolic systems, is based on the TREPAN algorithm. TREPAN, originally developed to extract decision trees from a trained neural network, differs from other algorithms that extract information from neural networks in several ways [8]:

1. **The Oracle**. It is used to determine the class of each instance that is presented as a query. The Oracle is used for three different purposes: to determine the class labels for the network's training examples; to determine the class labels for the tree's leaves; and to select the splits that create each of the tree's internal nodes.
2. **Split types**. That is, the way the input space is partitioned. TREPAN forms trees that use M-of-N expressions for its splits, that is a Boolean expression specified by an integer threshold, m, and a set of n Boolean conditions. An M-of-N expression is satisfied when at least m of its n conditions are satisfied.
3. **Split Selection**. Split selection involves deciding how to partition the input space at a given internal node in the tree. TREPAN uses a special heuristic search process to build its splitting test.
4. **Tree expansion**. TREPAN grows trees using a best-first expansion that chooses the node where there is the greatest potential to increase the fidelity of the extracted tree to the network.
5. **Stopping Criteria**. Trepan uses local and global stopping criteria. A local criterion considers the state of only a single node to decide whether or not it should be made a leaf, and a global criterion considers the state of the entire tree to decide if the tree-growing process should stop.

As in any decision tree based approach, for example C4.5 [21], the amount of training data reaching each node decreases with the depth of the tree. TREPAN creates new training cases by sampling the distributions of the training data and uses the trained ANN as an oracle to answer queries during the learning process. TREPAN requires as input the weights and biases of the trained neural network and a training data set. As output it produces a decision tree that provides an approximation to the function represented by the ANN.

In this study, we used the TREPAN system developed by the Centre for Molecular Design at the University of Portsmouth [22] as part of the project: "Biological Data Mining: A comparison of neural networks and symbolic techniques". The program is a Matlab [23] implementation of the original TREPAN for classification problems and was successfully applied to neural networks in a variety of bioinformatics and chemoinformatics models. [24]. We replaced the original oracle based on an ANN trained model by an SVM model, trained with the Matlab Support Vector Machine Toolbox [25]. The modified TREPAN requires as input Lagrange multipliers, bias, a training data set, as well as kernel function and supports vector of the trained SVM. Figure 1 presents the data flow for the extraction of knowledge from an SVM trained model.

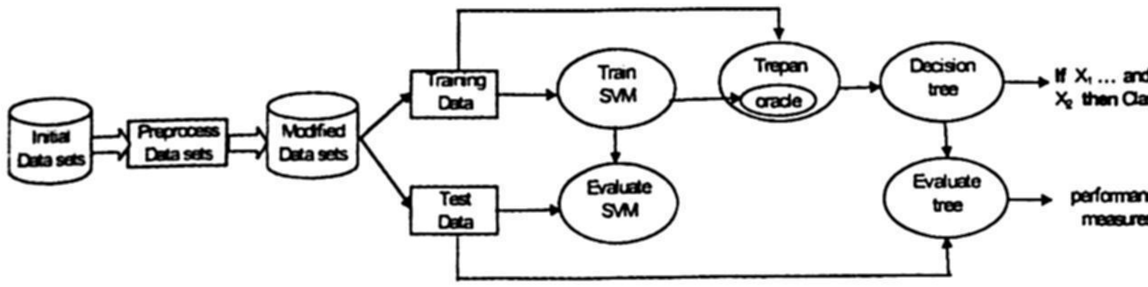

Fig. 1. Knowledge extraction data flow with TREPAN

As an example of the output generated by TREPAN, Figure 2 presents the decision tree extracted from the SVM model of the Haberman Survival data set (to be presented in the next section).

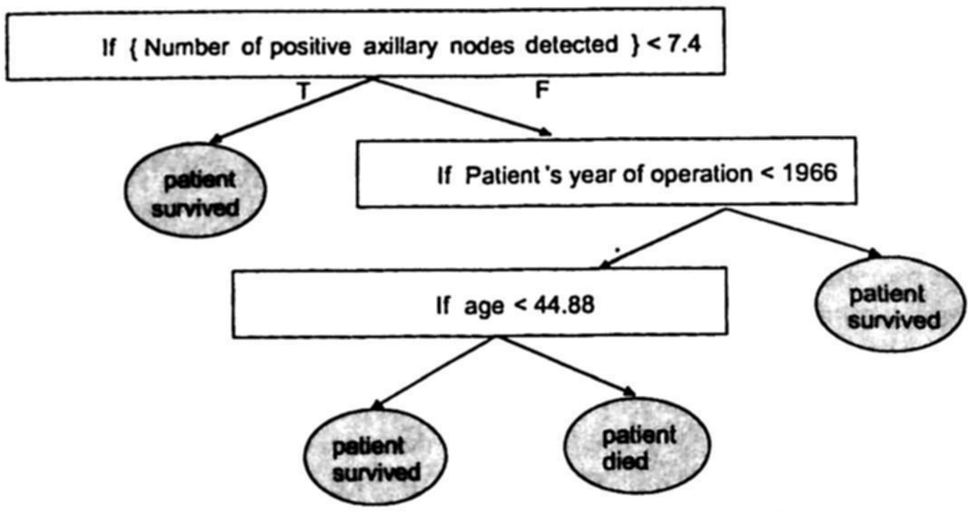
If { Number of positive axillary nodes detected } < 7.4
T
F
patient survived
If Patient 's year of operation < 1966
If age < 44.88
patient survived
patient survived
patient died

Fig. 2. Tree extracted by TREPAN for Haberman data set

Note that from the extracted tree, it is easy to obtain the following comprehensible rules:

- If Number of positive axillary nodes detected < 7.4 then class is patient survived 5 years or longer
- If Number of positive axillary nodes detected ≥ 7.4 and Patient's year of operation < 1966 and age < 44.88 then class is patient survived 5 years or longer

- If Number of positive axillary nodes detected ≥ 7.4 and Patient's year of operation < 1966 and age ≥ 44.88 then class is patient died within 5 year
- If Number of positive axillary nodes detected ≥ 7.4 and Patient's year of operation ≥ 1966 then class is patient survived 5 years or longer.

4 Experimental Results

4.1 Data Collection

We performed experiments on some commonly used data sets from the UCI repository [26] and Statlog project [27]. From the UCI Repository we selected the following data sets: Wisconsin Diagnostic Breast Cancer, Pima Indians Diabetes, Haberman Survival and Thyroid Gland. From the Statlog collection we selected the Heart disease. Table 1 shows the data sets characteristics: one of them correspond to multi-class data sets. A 10-fold cross-validation (CV) was performed. The data sets have been divided into 10 subsets of equal size. Each data sets was trained 10 times, each time leaving out one of the subsets from training and using only this omitted subset to evaluate the obtained model and average values are reported.

4.2 SVM Models

All data sets were trained using the Gaussian kernel and the Matlab Support Vector Machine Toolbox [25]. For all data sets we used the grid-search and cross-validation approach proposed by Hsu et al [28,29] using different kernel parameters in $\gamma = [2^4, 2^3, 2^2, \ldots, 2^{-10}]$ and $C = [2^{12}, 2^{11}, 2^{10}, \ldots, 2^{-2}]$.

For multi-class data sets we used the one-against-all approach [30]. We built *k* SVM models, with *k* equal to the number of classes. The *i*-th model is trained with all of the examples in the *i*-th class with positive labels, and all other examples with negative labels. Average values of accuracy are reported.

Table 2 shows the performance of the trained SVM models during the testing phase along with the best parameters (γ,C) and the average number of support vectors derived. The performance is measured using sensitivity, specificity and accuracy indexes [31].

$$\text{sensitivity} = \frac{TP}{TP + FN}; \quad \text{specificity} = \frac{TN}{TN + FP} \qquad \textbf{(2)}$$
$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where:

TP = Number of True Positive classified cases (the method correctly classifies)

TN = Number of True Negative classified cases (the method correctly classifies)
FP = Number of False Positive classified cases (the method labels a case as positive while it is a negative)
FN = Number of False Negative classified cases (the method labels a case as negative while it is a positive).

4.3 TREPAN Results

As previously described the trained SVM models were evaluated by TREPAN. Table 3 presents the average performance results for the testing phase for the data sets considered along with the average number of rules for the induced trees and the relative variation with respect to SVM accuracy. Table 3 shows that the Fidelity index (that is the percentage of predictions made by the tree extracted by TREPAN that agree with the predictions made by SVM), is over 90 % for all data sets, except Heart disease data set. The accuracy of the induced trees is very similar to the one obtained by SVM, with an average error of only 3.59 %. In others study the extracted rule from set has better generalization performance than the trained model, has been also reported in [24,33] for other data sets.

Table 1. Data sets and their characteristics

Data set	Number instances	Number Attributes	Attributes Type	Classes
Heart disease	270	13	Real, Nominal	2
Wisconsin Diagnostic Breast Cancer	569	32	Real	2
Pima Indians Diabetes	768	8	Real	2
Haberman Survival	306	4	Real	2
Thyroid Gland	215	5	Real	3

Table 2. Average performance results for SVM (Testing phase)

Data set	**(γ, C)**	**Average Support Vectors**	**Sensitivity %**	**Specificity %**	**Accuracy %**
Heart disease	$(2^{-6}, 2^{6})$	93.6	88.31	87.93	88.15
Wisconsin Diagnostic Breast Cancer	$(2^{-2}, 2^{3})$	69.1	98.09	98.06	98.07
Pima Indians Diabetes	$(2^{-2}, 2^{1})$	373.2	74.15	79.4	77.99
Haberman Survival	$(2^{-3}, 2^{3})$	147.3	60.61	77.66	75.82
Thyroid Gland	$(2^{-0}, 2^{4})$	19.3	98.36	95.41	97.36

Table 3. Average performance results for TREPAN (Testing phase)

Dataset	**Fidelity %**	**Average Number of rules**	**Sensitivity %**	**Specificity %**	**Accuracy %**	**Relative Variation %**
Heart disease	86.67	7.0	84.21	81.36	82.96	-5.89
Wisconsin Diagnostic Breast Cancer	94.55	4.5	92.31	94.46	93.67	-4.49
Pima Indians Diabetes	91.66	12.2	70.14	78.46	76.17	-2.33
Haberman Survival	94.6	3.8	62.07	77.26	75.82	-0.16
Thyroid Gland	93.87	3.9	94.19	88.37	92.25	-5.25

Figures 3-6 show the trees extracted by TREPAN for the datasets: Heart Disease, Wisconsin Diagnostic Breast Cancer, Pima Indians Diabetes and Thyroid Gland, respectively. The trees extracted correspond to the fold with the highest value of accuracy. These examples are related to a single class. For example, Fig. 3 shows the tree extracted by TREPAN for data set Heart disease. From here, a set of rules can be generated to classify absence or presence of heart disease.

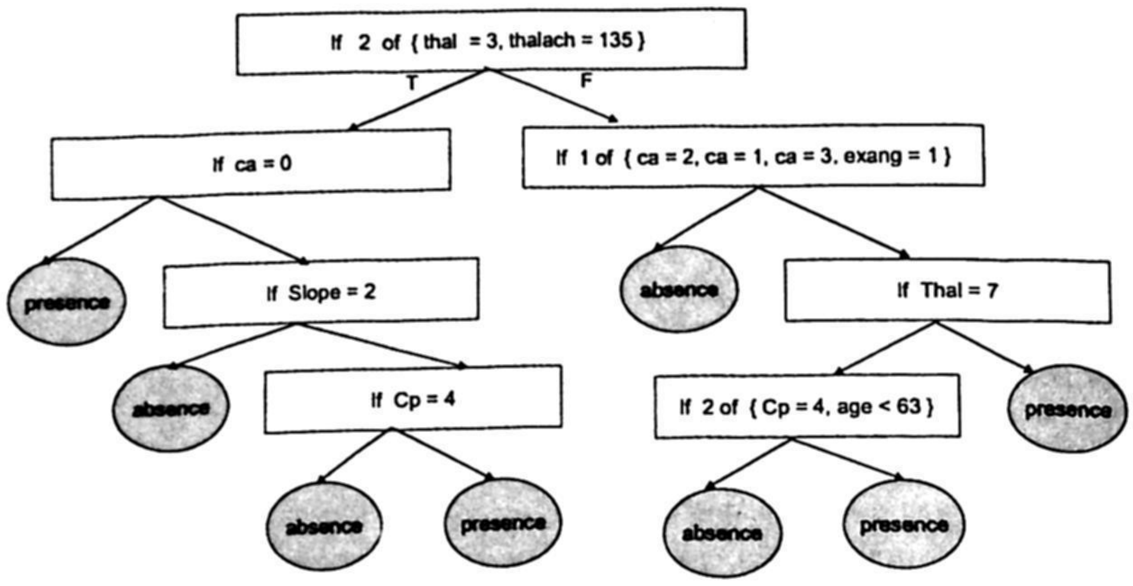
If 2 of { thal = 3, thalach = 135 }
T
F
If ca = 0
If 1 of { ca = 2, ca = 1, ca = 3, exang = 1 }
presence
If Slope = 2
absence
If Thal = 7
absence
If Cp = 4
If 2 of { Cp = 4, age < 63 }
presence
absence
presence
absence
presence

Fig. 3 Tree extracted by TREPAN for data set Heart disease

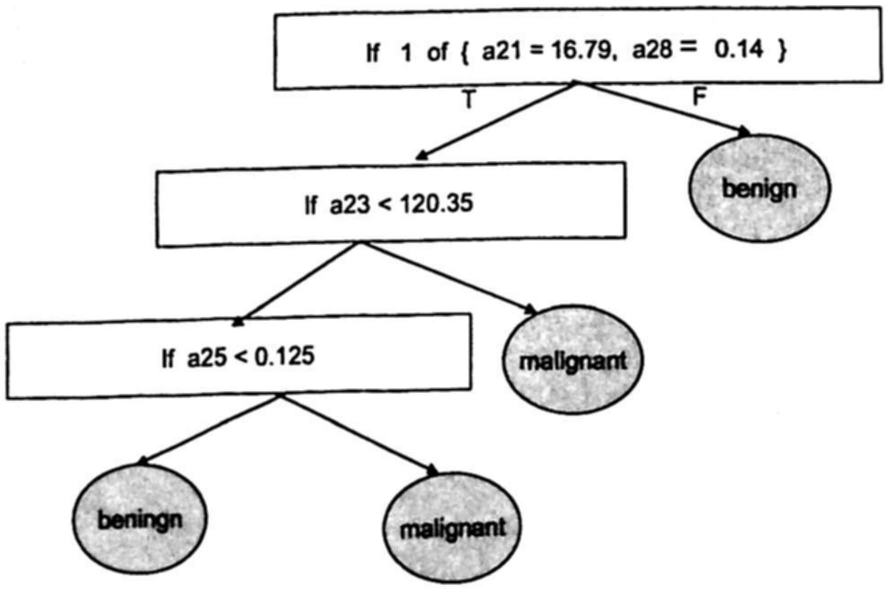

Fig. 4 Tree extracted by TREPAN for data set Wisconsin Diagnostic Breast Cancer

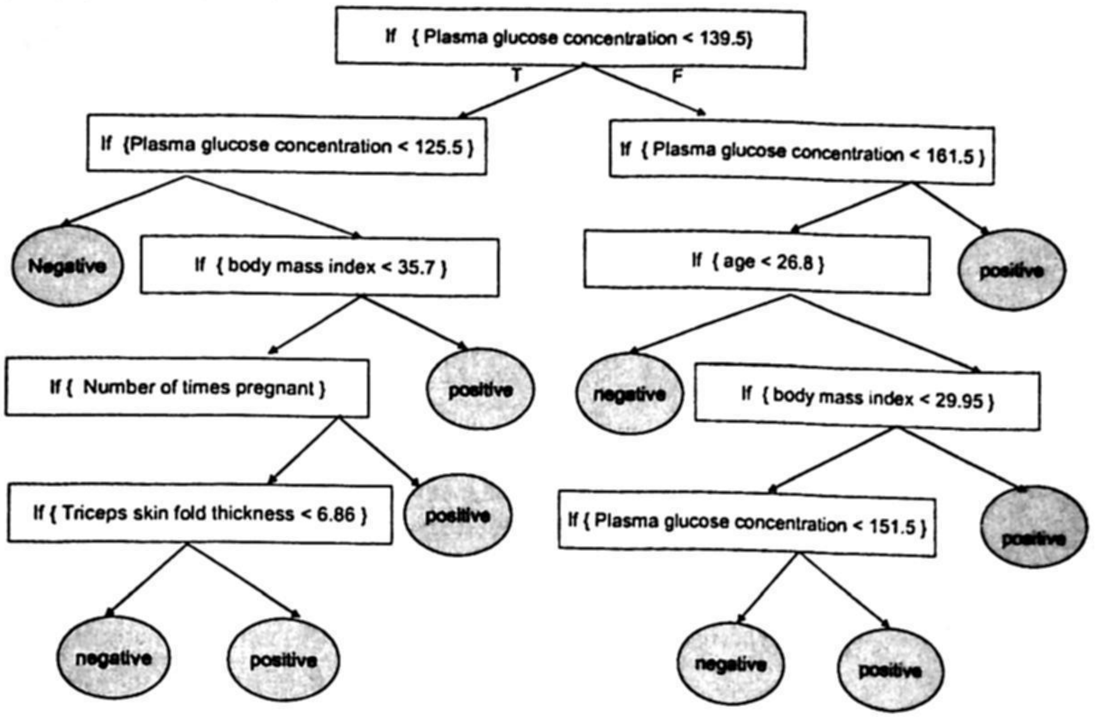
If { Plasma glucose concentration < 139.5}
T
F
If {Plasma glucose concentration < 125.5 }
If { Plasma glucose concentration < 161.5 }
Negative
If { body mass index < 35.7 }
If { age < 26.8 }
positive
If { Number of times pregnant }
positive
negative
If { body mass index < 29.95 }
If { Triceps skin fold thickness < 6.86 }
positive
If { Plasma glucose concentration < 151.5 }
positive
negative
positive
negative
positive

Fig. 5 Tree extracted by TREPAN for data set Pima Indians Diabetes

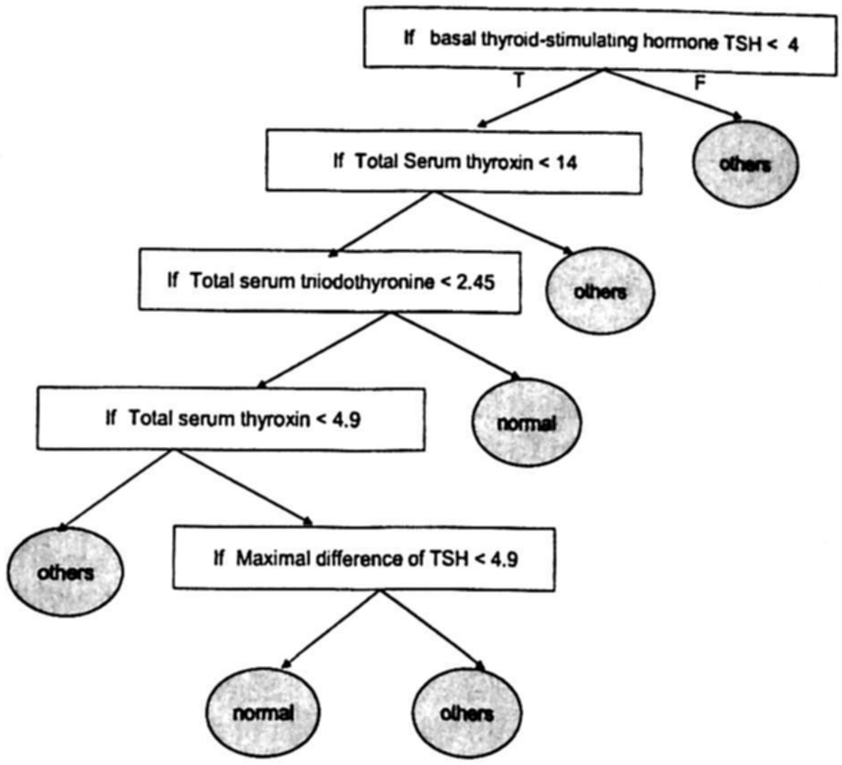
If basal thyroid-stimulating hormone TSH < 4
T
F
If Total Serum thyroxin < 14
others
If Total serum triiodothyronine < 2.45
others
If Total serum thyroxin < 4.9
normal
others
If Maximal difference of TSH < 4.9
normal
others

Fig. 6 Tree extracted by TREPAN for data set New Thyroid Gland

Table 4 shows the average accuracy rates for the data sets considered (for the testing phase), of SVM, TREPAN and C4.5, a well-known decision tree algorithm [21].

Table 4. Average Accuracy Rate (%) for SVM, TREPAN and C4.5

Data set	SVM	TREPAN	C4.5
Heart disease	88.15	82.96	78.52
Wisconsin Diagnostic Breast Cancer	98.07	93.67	94.73
Pima Indians Diabetes	77.99	76.17	75.39
Haberman Survival	75.82	75.82	71.9
Thyroid Gland	97.36	92.25	92.09

In order to empirically evaluate the accuracy of the models, a statistical method, as suggested by Mitchell [32] was used. Using a 95 % confidence level, the statistical test shows that average accuracy for SVM and TREPAN on Pima Indians Diabetes and Haberman Survival data sets are equal, while SVM outperforms TREPAN on the other data sets. However, TREPAN has a better explanation capability.

5 Conclusions

In this paper we have presented an approach for extracting rules from SVM trained models. The proposed approach, based on TREPAN, a modification of the well know TREPAN algorithm, was tested on five commonly data sets publicly available. The results showed that the proposed approach produces a classification system with performance indexes as accurate as the trained SVM models but providing a set of trees, which allow a better understanding of the data set under analysis.

Acknowledgements. The author are grateful to the anonymous referees for their comments and valuable suggestions on the earlier version of the paper.

References

1. Jain L.C, Martin N. M. (1998): Fusion of Neural Networks, Fuzzy Sets, and Genetic Algorithms, Industrial Applications. Ed. CRC Press.
2. Jacobsen H.A. (1998): A Generic Architecture for Hybrid Intelligent Systems. IEEE Fuzzy Systems IEEE Fuzzy Systems. Anchorage, Alaska.
3. Núñez H., Angulo C., Català A. (2002): Rule-extraction from Support Vector Machines, ESANN'2002 Proceedings – The European Symposium on Artificial Neural Networks, Bruges (Belgium), ISBN 2-930307-02-1, pp. 107-112.
4. Fung G., Sandilya S., Rao B.(2004): Rule extraction from Linear Support Vector Machines, Computer Aided Diagnosis & Therapy Solutions, Siemens Medical Solutions, Submitted.
5. Barakat N., Diederich J.(2004): Learning-based Rule-Extraction from support Vector Machines Performance On Benchmark Data Sets. In Kasabov, N. and chan, Z. s. H. Eds. Conference on Neuro-computing and Evolving Intelligence (KEDRI 2004), Auckland, New Zealand.
6. Zhang Y., Su Y., Jia T. and Chu J. (2005): Rule Extraction from Trained Support Vector Machines. PAKDD, LNAI 3518, Springer-Verlarg, Berlin
7. http://www.biostat.wisc.edu/~craven/
8. Craven M.W. (1996): Extracting Comprehensible Models from Trained Neural Networks. Ph.D. Thesis. University of Wisconsin-Madison.
9. Cristianini N., Shawe-Taylor J. (2000): An Introduction to Support Vector Machines. Cambridge University Press, Cambridge.
10. Vapnik V. (1998): Statistical Learning Theory, John Wiley & Sons.
11. Cortes C., Vapnik V. (1995): support-vector network. Machine Learning, 20:273-297
12. Guyon I, Stork G (2000) Linear Discriminant and Support Vector Classifiers In: Advances in Large Margin Classifiers ed. By Smola A.J. et al. The MIT Press Cambridge, MA.
13. Campbell C. (2000): An Introduction to Kernel Methods, In R.J. Howlett and L.C. Jain, editors, Radial Basic Function Networks: Design and Applications, Springer Verlag, Berlin, 2000, 31.
14. Valentini G., Muselli M., Ruffino F. (2004): Cancer recognition with bagged ensembles of support vector machines. Neurocomputing, 56: 461-466.
15. Bertoni A., Folgieri R., Valentini G. (2005): Bio-molecular cancer prediction with random subspace ensembles of support vector machines. Neurocomputing, 63: 535-539.
16. Cardoso J., Pinto J., Cardoso M. (2005): Modelling ordinal relations with SVMs: An application to objective aesthetic evaluation of breast cancer conservative treatment. Neural Networks, 18: 808-817.
17. http://www.comp.nus.edu.sg/~pris/HybridSystems/DescriptionDetailed1.html
18. Jerez J., Gómez J., Ramos G., Muñoz J., Alba E. (2003): A combined neural network and decision trees model for prognosis of breast cancer relapse. Artificial Intelligence in Medicine, 27: 45-63.
19. Kim K-J, Cho S-B. (2004): Prediction of colon cancer using an evolutionary neural network, 61: 361-379.

20. Yeh J-S., Cheng C-H. (2005): Using hierarchical soft method to discriminate microcyte anemia. Expert Systems with Applications, 29: 515-524.
21. Quinlan J.R. (1993): Programs for Machine Learning, Morgan Kaufmann Publishers.
22. Trepan – Matlab. Available at http:///www.cmdmport.ac.uk/biomine
23. MathWorks (2002) *Matlab 6.5* R13.
24. Hudson B., Whitley D., Ford M., Browne A. (2003): Biological Data Mining: A comparison of Neural Network and Symbolic Techniques. Technical Report. Centre for Molecular Design, University of Portsmouth. Available at: http://www.cmd.port.ac.uk/biomine
25. Cawley G. C. MATLAB Support Vector Machine Toolbox v. 0.54. University of East Anglia School of Information Systems, Norwich, U.K. Available at: http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox
26. C.L.Blake and C.J.Merz: UCI Repository of Machine Learning Databases. University of California, Department of Information and computers Science, Irvine, CA. available at: http://www.ics.edu/~mlearn/MLRepository.html
27. Statlog. http://www.liacc.up.pt/ML/statlog/databases.html
28. Hsu C.-W, Lin C.J. (2002): A Comparison of Methods for Multi-class Support Vector Machines. IEEE Transaction on Neural Networks, 13: 415-425.
29. Hsu C.-W., Chang C.-C., Lin C.-J. (2004): A Practical Guide to Support Vector Classification. Department of Computer Science and Information Engineering, National Taiwan University. Available at http://www.csie.ntu.edu.tw/ ~cjlin/papers/guide/guide.pdf.
30. Anguita D., Ridella S., Sterpi D. (2004): A new Method for Multiclass Support Vector Machines Proc. of the IEEE Int. Joint Conf. on Neural Networks, IJCNN 2004, Budapest, Hungary, July 2004
31. Veropoulos K., Campbell C., Cristianini N. (199): Controlling the Sensitivity of Support Vector Machines, Proceedings of the IJCAI99.
32. Mitchell T. M. (1997): Machine Learning , McGraw Hill, New York.
33. Torres D., Rocco C. (2005): Extracting Trees from Trained SVM Models using TREPAN Based Approach. Hybrid Intelligent System, IEEE Computer Society, RJ, Brasil.