Extracting Diagnostic Rules from Support Vector Machine

G.Suganya [a,*], D.Dhivya [b,1]

Abstract - **Several data mining and machine learning methods have been used for the diagnosis, prognosis, and management of diabetes. This paper presents a rule extraction approach to obtain maximum accuracy for the prediction of diabetes. Support vector machines (SVMs) utilized for the diagnosis of diabetes. In particular, an additional explanation module, which turns the black box model of an SVM into an intelligible representation of the SVM's diagnostic (classification) decision. Results on a real-life diabetes dataset show that intelligible SVMs provide a promising tool for the prediction of diabetes, where a comprehensible rule set have been generated. Rule extraction describes a pattern or relationships between input features and output class labels directly from the data. This paper utilizes two different techniques for rule extraction. Sequential covering Rule Extraction and the eclectic methods are used to turn the SVM black box into a more intelligible model.**

Index Terms - **K-means clustering, Support vector machine, Rule Extraction, diabetes.**

I. INTRODUCTION

Learning technique based on the theory of statistical learning and implements the principle of structural risk minimization [1]. SVM constructs a hyper plane or set of hyper planes in high dimensional space without any assumptions on the data distribution, which can be used for Classification and Regression tasks. Accordingly SVM is called by the name Support Vector Classification and Support Vector Regression.

The classification problem can be restricted to consideration of the two-class problem without loss of generality. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples. The linear classifier [9] is termed the optimal separating hyper plane. Intuitively, we would expect this boundary to generalize well as opposed to the other possible boundaries. Larger and more complex classification problems have been attacked with SVC.

The advantages of classical Support Vector Classification (SVC) are: a global minimum solution; relatively fast training speed for large-scale learning tasks; and sparseness in solution representation. Notably, has applied SVC to the exacting problem of face recognition, with encouraging results. In conclusion, SVC provides a robust method for pattern classification by minimizing over fitting problems. Use of a kernel function enables the curse of dimensionality to be addressed, and the solution implicitly contains support vectors that provide a description of the significant data for classification.

SVMs can also be applied to regression problems by the introduction of an alternative loss function.

The loss function must be modified to include a distance measure. Similarly to classification problems, a non-linear model is usually required to adequately model data. In the same manner as the non-linear SVC approach [2], a non-linear mapping can be used to map the data into a high dimensional feature space where linear regression is performed. The kernel approach is again employed to address the curse of dimensionality. Support Vector Regression (SVR) has been applied to some time series modeling problems. Notably, has achieved excellent results in applying SVR to one of the data sets. Support vector machines (SVM) have generated a great interest in the community of machine learning due to its excellent generalization performance in a wide variety of learning problems.

SVM training [2] is relatively easy. Because it has no local optimal, unlike in neural networks. SVM scales relatively well to high dimensional data. The tradeoff between classifier complexity and error can be controlled explicitly. SVM is also used for nontraditional data like strings and trees can be used as input to SVM, instead of feature vectors. SVM also has some disadvantages like, it needs a good kernel function[7]. SVM is a black box classifier, the decisions they make are not always easily explainable. The model produced does not naturally provide any useful reasons about why a particular point is classified in one class rather than another. In medical diagnosis, there is a clear need for an explanation component to be associated with classification decisions. So black box model is not suitable for medical diagnosis purposes.

Rule extraction [3] is used to overcome the demerits of black box model. Rule extraction is considered to be understandable and that allow an explicit control of their complexity to meet end user requirements. Rule extraction is also used to maximize the accuracy of the resulting rule set. Rule extraction is one of expressing the knowledge acquired by the SVM during training process in a comprehensible form easily understood by the end user.

The most popular classification techniques are based on the construction of propositional if-then rules from prelabeled training data[6].

These methods, in principle, provide a completely transparent classification decision, but, in practice, their performance and comprehensibility often suffer in cases of high-dimensional data and continuously valued attributes. Another popular family of classifiers exemplified by support vector machines (SVMs) and artificial neural networks (ANNs) builds a mathematical model of the data that often performs significantly better in these situations. Direct rule learners extract rules that describe a pattern or relationships between input features and output class labels directly from the data. Therefore, the task of rule extraction is to devise rules from the model rather than directly from the data. We utilize two different techniques for rule extraction in our proposed approach. The techniques are sequential covering rule extraction and eclectic rule extraction. The performance of the proposed approach is tested using a real life diabetes dataset[14].

II. PROPOSED MODEL FOR RULE EXTRACTION

The proposed methodology for Rule Extraction is used to obtain maximum accuracy of rules generated by support vector machines. Rule extraction is to devise rules from the model rather than directly from the data. Extracting diagnostic rules have following modules:

A. Data Collection
B. Data Preprocessing
C. SVM Training
D. Rule Extraction

Figure 1 shows the schematic representation of the issues to be addressed while developing a Rule extraction model.

A. Data Collection

The data collected[10] about each subject include Number of times pregnant, Plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg),Triceps skin fold thickness (mm),2-Hour serum insulin (mu U/ml),Body mass index (weight in kg/(height in m)^2),Diabetes pedigree function, Age(years).
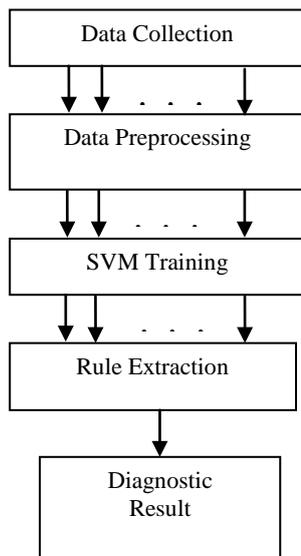

Figure 1. Rule Extraction model

B. Data Preprocessing

The clustering algorithm was employed to ensure the selection of a representative training set, especially for the negative (majority) class. This was done because selecting training examples at random may result in important patterns in the original dataset, impeding the quality of the learned model.

The K-means clustering algorithm [13] was given below.
1) Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2) Assign each object to the group that has the closest centroid.
3) When all objects have been assigned, recalculate the positions of the K centroids.
4) Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Five clusters were devised from the original dataset. Within each cluster, samples (subjects) were sorted based on their Euclidean distance from the cluster center. Samples at the maximum distance from the cluster center were excluded, as they are likely to be outliers. From each of the sorted clusters, every third sample is included in the final dataset, to select representative examples from each cluster, at different Euclidean distances from the center. All positive samples were included in the final dataset. Details of the training and testing sets are shown in Table I.

Training		Testing	
+ve	-ve	+ve	-ve
77	142	30	70

C. SVM Training

SVM training is used to predict the diagnosis [diabetic (class label 1), non-diabetic (class label 0)].Leave one out cross validation is used to find the training parameters.

A fast support vector machine (SVM) training algorithm is proposed under the decomposition framework of SVM's algorithm by effectively integrating kernel caching, digest and shrinking policies and stopping conditions.

Complex problem is first decomposed into a series of sub-problems which can be easily solved with the existing techniques, and then solutions to these sub-problems can be combined to solve the original problem. Still, this principle is suitable for SVM's training algorithm.

Training a support vector machine (SVM) leads to a quadratic optimization problem with bound constraints and one linear equality constraint. Despite the fact that this type of problem is well understood, there are many issues to be considered in designing an SVM learner. In particular, for large learning tasks with many training examples, off-the-shelf optimization techniques for general quadratic programs quickly become intractable in their memory and time requirements.

D. Rule Extraction

The proposed model takes advantage of the information provided by the support vectors. They are used to determine the boundaries of regions defined in the input space (ellipsoids or hyper-rectangles). These regions are obtained from the combination of prototype vectors and support vectors. The prototype vectors [4][5] are computed through a clustering algorithm. Each region defines a rule with its corresponding syntax: equation rules, which correspond to mathematical equations of the ellipsoids and interval rules, associated with hyper-rectangles defined from parallel clusters to the coordinate axes.

The clusters are defined by the prototype, which will be the center, and by a support vector within the partition. The chosen support vector will be the farthest to the prototype. The straight line defined by these two points is the first axis of the cluster. By simple geometry, the rest of the axes and the associate vertices are determined. There are three possibilities to define these vertices: with the support vector itself, derived from a support vector or with the farthest point to the prototype [8]. To construct hyper-rectangles a similar procedure is followed. The only difference is that lines parallel to the coordinate axes are used to define the axes of the associated clusters.

For each iteration, there are m regions with a positive partition test and p regions with a negative partition test. These last ones are transferred to rules. In the next iteration, the data of the m regions are used to determine m+1 new prototypes and to generate m+1 new clusters. If the maximum number of iterations is reached, all the regions (independently of the results of the partition test) are transferred to rules.

III. REVIEW OF SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) [12] is a learning machine that plots the training vectors in high dimensional feature space, and labels each vector by its class. SVM classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space .

The SVM is based on the idea of structural risk minimization[11], which minimizes the generalization error, i.e. true error on unseen examples. The number of free parameters used in the SVM depends on the margin that separates the data points to classes but not on the number of input features. Thus SVM does not require a reduction in the number of features in order to avoid over fitting. More formally, a support vector machine constructs a hyper plane or set of hyper planes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks.

Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. In this two class classification problem we assume we are given a training data set, $X \in \mathbb{R}^n$, $d_k \in \{-1,+1\}$, and an indicator function f which is the bipolar signum function which will ultimately permit a mapping from each of the input points X to the appropriate d_k. Class $_1$ points will be indicated by a+1 which we refer to as positive samples: class $_2$ points will be indicated by a-1 and thus called negative samples. Note that linear separability implies that we can find an oriented hyper plane defined by a set of weights W and a bias ω_0 (assuming that the weight vector is not augmented) which separates the positive data points from the negative ones.

IV. RESULT AND DISCUSSION

Cross validation takes the group structure of the data into account. Group is a grouping vector that defines the class for each observation. Group can be a numeric vector, a string array, or a cell array of strings. The partition of the groups depends on the type of cross-validation: For K-fold, each group is divided into K subsets, approximately equal in size. For all others, approximately equal numbers of observations from each group are selected for the evaluation set. In both cases the training set contains at least one observation from each group.

It trains a support vector machine classifier using data training taken from two groups given by group. SVMStruct contains information about the trained classifier, including the support vectors that are used by SVMclassify for classification. Group is a column vector of values of the same length as training that defines two groups. Group can be a numeric vector, a string array, or a cell array of strings. SVMtrain treats empty strings in GROUP as missing values and ignores the corresponding rows of training.

It classifies each row of the data in Sample using the information in a support vector machine classifier structure SVMStruct, created using the svmtrain function. Sample must have the same number of columns as the data used to train the classifier in svmtrain. Group indicates the group to which each row of Sample has been assigned. The performance of SVM is shown in figure 2.
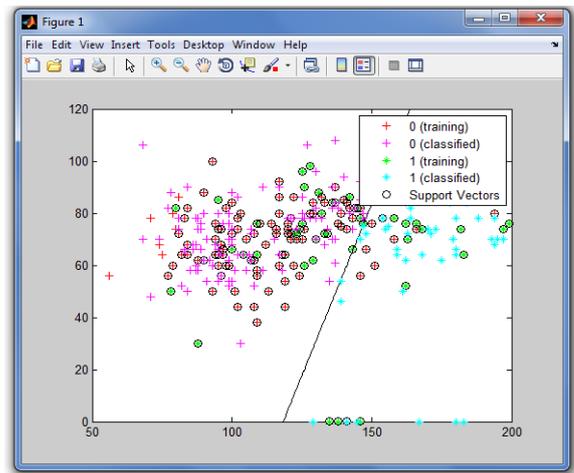

Figure 2. Performance of SVM

The performance of SVM is calculated based on correct rate and error rate during classification process. The result is shown in Table II.

TABLE II
PERFORMANCE OF SVM

SVM	
Accuracy	
Correct Rate	Error Rate
83.02%	16.98%

Rule Extraction extracts rules directly from a subset of the SVM SVs, using a modified sequential covering algorithm and based on an ordered search of the most discriminatory features. These features are then ranked and utilized to form an initial ruleset for the positive class, which is then refined and pruned. A prepruning strategy is adopted to prune rules with performance below a user-defined threshold. In the postpruning step, the algorithm utilizes the area under the receiver operating characteristic (ROC) curve (AUC) to control the tradeoff between the classifier (ruleset) performance. Rule extraction is shown in figure 3.
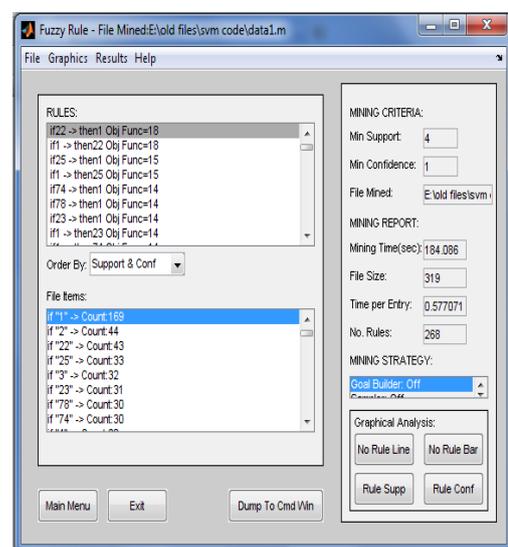

Figure 3. Rule Extraction

Rules generated based on the minimum support, minimum confidence. The performance of rule extraction is calculated by mining time and time per entry.

V. CONCLUSION

The hybrid system for medical diagnosis was developed. Data preprocessing was employed for sub sampling to ensure the selection of a representative training set. We have employed SVMs for the diagnosis and prediction of diabetes, where an additional rule-based explanation component is utilized to provide comprehensibility. The SVM and the rules extracted from it, are intended to work as a second opinion for diagnosis of and as a tool to predict diabetes through identifying people at high risk.

REFERENCES

[1] S. Fine and K. Scheinberg. "Efficient SVM training using low-rank kernel representations". In Journal of Machine Learning Research, 243C264, 2001.

[2] S. V. N. Vishwanathan, and M. N. Murty. "SSVM: a simple SVM algorithm". In Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, Vol.3, 2393-2398, Honolulu,HI, USA, 2002.

[3] N. Barakat and J. Diederich, "Eclectic Rule-Extraction from Support Vector Machines," Int'l. Computational Intelligence, vol. 2, no. 1, pp. 59-62, 2005.

[4] A.P. Bradley, "The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms," Pattern Recognition vol. 30, no. 7, pp. 1145-1159, 1997.

[5] N. Barakat and A. P. Bradley, "Rule extraction from support vector machines: A sequential covering approach," IEEE Trans. Knowl. Data Eng.,vol. 19, no. 6, pp. 729–741, Jun. 2007.

[6] N. Barakat and A.P. Bradley, "Rule extraction from support vector machines: Measuring the explanation capability using the area under the ROCcurve," presented at the 18th Int. Conf. Pattern Recognit. (ICPR 2006).

[7] T. Joachims, "Learning to Classify Text Using Support Vector Machines,"Norwell, MA: Kluwer, 2002.

[8] H. Nunez, C. Angulo, and A. Catala, "Rule Extraction from Support Vector Machines," Proc. European Symp. Artificial Neural Networks, pp. 107-112, 2002.

[9] C. Burges. "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, Kluwer Academic, 1998.

[10] C. Merz and P. Murphy, UCI machine learning repository, Irvine, 1992 .

[11] A.Kampouraki, C. Nikou, and G. Manis, "Robustness of support vector machine-based classification of heart rate signals," in Proc. IEEE Conf. Eng.Med. Biol. Soc. (EMBS 2006),NewYork,Aug.–Sep., pp. 2159–2162.

[12] R. Kondor and T. Jebara, "A kernel between sets of vectors," in Proc. 20th Int. Conf. Mach. Learn. (ICML), Washington, DC, 2003, pp. 361–368.

[13] P. N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining", NewYork: Addison-Wesley, 2005.

[14] Barakat, "Rule extraction from support vector machines: Medical diagnosis prediction and explanation," Ph.D. thesis, School Inf. Technol. Electr. Eng. (ITEE), Univ. Queensland, Brisbane, Australia, 2007.

BIOGRAPHY

Ms.G.Suganya is currently working as a Lecturer in the Department of Information Technology in Narasu`s Sarathy Institute of Technology, Salem. She received her M.Tech degree in Mainframe Technology under Anna University of Technology, Coimbatore in May 2011. She received her B.E degree in Computer Science and Engineering, under Anna University Chennai in May 2009. Her research interest is in Data Mining and Bioinformatics.

Ms.D.Dhivya is currently working as a Lecturer in the Department of Computer Science and Engineering in Narasu`s Sarathy Institute of Technology, Salem. She received her M.E degree in Computer Science and Engineering under Anna University of Technology, Coimbatore in May 2011. She received her B.E degree in Computer Science and Engineering, under Anna University Chennai in May 2009. Her research interest is in Data Mining and Bioinformatics.