Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Region based Support Vector Machine Algorithm for Medical Diagnosis on Pima Indian Diabetes DataSet

Savvas Karatsiolis
Computer Science Department
University of Cyprus
Nicosia, Cyprus

Christos N. Schizas
Computer Science Department
University of Cyprus
Nicosia, Cyprus

***Abstract*—The problem of diagnosing Pima Indian Diabetes from data obtained from the UCI Repository of Machine Learning Databases[6] is handled with a modified Support Vector Machine strategy. Performance comparison with previous studies is presented in order to demonstrate the proposed algorithm's advantages over various classification methods. The goal of the paper is to provide the grasp of a methodology that can be efficiently used to raise classification success rates obtained by the use of conventional approaches such as Neural Networks, RBF networks and K-nearest neighbors. The suggested algorithm divides the training set into two subsets: one that arises from the joining of coherent data regions and one that comprises of the data portion that is difficult to be clustered. Consequently, the first subset is used to train a Support Vector Machine with a RBF kernel and the second subset is used to train another Support Vector Machine with a polynomial kernel. During classification the algorithm is capable of identifying which of the two Support Vector Machine models to use. The intuition behind the suggested algorithm relies on the expectation that the RBF Support Vector Machine model is more appropriate to use on data sets of different characteristics than the polynomial kernel. In the specific study case the suggested algorithm raised average classification success rate to 82.2% while the best performance obtained by previous studies was 81% given by a fine tuned highly complex ARTMAP-IC model.**

***Index Terms*—*Support Vector Machine, Pima Indian Diabetes, Clustering, Support Vector Machine Kernel*

I. Introduction

Medical applications have been pushing Computational Intelligence advancement for some decades now mainly because of the need to improve the accuracy of diagnosis and the need to reduce the accompanied cost. At the same time the medical sector provides plenty of structured information for the researchers to experiment upon. It is an indicative fact that the data sets belonging to "life sciences" category currently available on UCI Repository are more than 28% of the total data sets available. In turn, this amplifies the need to constantly improve available prediction algorithms and come up with new efficient models of data manipulation.

The Pima Indian Diabetes data set includes information gathered from females that are at least 21 years old of the Pima Indian heritage. It's a relatively popular set probably because of the difficulty it opposes towards achieving good classification results, a challenge that seems to be responsible for the data set's quite few citations. As a result of the challenging nature of the specific dataset, a variety of simple and complex models have been tried to achieve improved classification results [1,2,3,4,5], but still the success rates remain only around 80%. Smith et al. [4] used the PID data set to evaluate the perceptron-like ADAPtive learning routine (ADAP). This study had 576 cases in the training set and 192 cases in the test set. Using 576 training instances, the sensitivity and specificity of their algorithm was 76% on the remaining 192 instances. The same number of random training and test sets was used to compare the simulation results. On the Pima Indian Diabetes (PID) database fuzzy ARTMAP test set performance was similar to that of the ADAP algorithm [4] but with far fewer rules and faster training. An ARTMAP pruning algorithm [2] further reduces the number of rules by an order of magnitude and also boosts test set accuracy to 79%.

Methodology	Classification success rates(%)
Logistic Regression	77
MLP(Levenberg-Marquardt)	77
ADAP	76
RBF	68.23
General Regression NN	80.21
ARTMAP-IC	**81**
KNN	77
ARTMAP	66

Table 1. Performance results of various studies on the PID dataset

An instance counting algorithm ARTMAP-IC [1] improves accuracy to 81%, at the expense of added complexity to the model used. Comparison of test set performance of the referenced studies is presented in Table 1 [1,2,3,4,5].

The relatively low success achieved by different angles of attack is frustratingly opposing claims that the specific data set is complete and correct. As a matter of fact, this is the general case when a bunch of machine learning algorithms all fail to come up with a model that achieves satisfactory recall results. Their results seem to approach a solid upper bound that seems hard to overcome and at the same time theoretically weaker models (linear classifiers, KNN) achieve similar success rates. Analyzing this fact, one may come to the conclusion that a combination of two disappointing omissions constraint machine learning algorithms from generalizing well over the dataset: Appropriate attributes are not used or the variation of attributes over time is highly contributive to the information basis of the problem and is not examined or is not correctly embedded in any form in the dataset. These two problematic phenomena are explained in the following paragraphs.

In order to gain a better understanding of the first constraint let's consider a diagnostician who is able to analyze a person's genome and consequently to directly check the integrity of a gene (or genes) responsible for a specific medical condition. This approach would cancel out the need for a machine learning algorithm to recognize a disease, since an observation of a directly linked “quantity” (responsible gene's structure) provides the answer with zero or negligible uncertainty. Adding levels of abstraction by observing the phenotype of a gene's expression in proteins raises the need to combine measurements of phenotypes and uncertain indicators to generalize over these observations. The higher the level of abstraction, the more carefully the attributes must be selected in order to encapsulate the required information necessary that in turn will drive the learning process. Usually, a higher level of abstraction contributes in the lower expense of data collection but at the same time it fades away the strength of the immediate link and the causality between the observation-conclusion pair. This leads to the use of less appropriate but cheaper to obtain attributes.

The reasoning behind the second omission (not using time varying data relations) when referring to medical problems relies partly on the diversity of DNA and mainly on the interrelations of living organisms body chemistry. A measurement of a physical quantity is affected by factors that do not have to do with its immediate effectors. For example, blood sugar may be increased by factors uncorrelated to diet or pathology, like stress or infections. Taking various measures over time averaging out some unwanted conditions may reveal some information that would normally be overseen by one-time measurements or make it possible to avoid unwanted measurement noise. Also a side effect of the dataset attributes' abstraction may be the delayed effect of a condition on the values of the attributes selected, which could be misleading. For example, the stage of a disease may have different degrees of influence on the attributes, especially on early and later stages of manifestation.

When a dataset is used to train a classification model there is little to do about the second omission regarding time varying data relations. This should probably be considered when conducting data collection. As far as the first problem is concerned, regarding attributes' appropriateness, the algorithm proves that mapping a carefully selected subset of the available data set to a constrained VC dimensioned feature space (through the polynomial kernel SVM) could reveal some relations that would be otherwise unseen when mapping the whole data set to a high or infinite dimensional feature space at once (through a RBF SVM). This fact suggests that some data set cases have large projections on a group of features that overwhelm weaker projections on other features that can provide the means to improve classification rates. The goal is to separate the data set into subsets in a way that this overwhelming phenomenon is reduced or even minimized.

II. The reasoning behind clustering the dataset

A data set is considered to be of some value provided its information basis is maintained and is not malformed by noise to a grade of no separability and the distribution of data is more or less imprinted in collected measurements. Given that theses prerequisites are met, most of the time one can identify two categories of the data in respect to their feature space separability: the category of data that can be classified correctly in an easy or moderate-easy way and a category of cases that is more challenging or very hard to separate with the use of a single model. When dealing with classification problems it is usual to try out several generalizing models towards an improved and approved solution, mainly including neural networks with different architectures and gradient-based search algorithms, support vector machines with various kernels, KNN classifiers, clustering algorithms like k-means, discriminant models etc. It is common to observe that specific test cases in the validation set are always classified correctly and easily. By easy classification i mean at a safe distance from the separation boundary. A more accurate expression would be “classified correctly with high confidence” but the sloppier definition is preferred for the sake of the expressing argument. On the other hand, specific cases may almost always be misclassified. The above scenario is mainly true when high successful classification rates over the validation set cannot be achieved in contrast to a significantly lower training error without this phenomenon appearing as a consequence of over-fitting. But it may be an effect of erroneous or noisy measurements and in some cases could be caused by the use of inappropriate or incomplete attributes set. When the latter possibility concentrates the researcher's suspicions, a principal component analysis of the data set may shed some light to the investigation.

This paper concentrates on the event that nothing of the above problematic cases is present but rather a more disturbing scenario is at hand: The dataset cases are by nature difficult to separate by a single generalizing machine learning model. It may be the case that the majority of the examples in the validation set are separated correctly by a number of generalizing models in a highly successful rate but a small or medium-sized minority of test cases is terribly processed. The end result is, of course, the degradation of the overall classification performance. An anticipated side effect of this phenomenon is the fact that a linear separation model of the data set gives similar (or even better) results to a non-linear separation model to the surprise of the researcher. This surprise is based on the strong belief that the attribute space of the data set is not linearly separated. The intuition behind this situation is that while the algorithm is trying to minimize the training error while constructing a classification boundary, it faces the problem that following a generalization concept that services the majority of the cases of the problem it raises the training error of a respectable (in terms of size) minority. It will eventually favor the largest of the training cases group because minimizing the objective function is apparently only feasible through the favoring of the majority. At the same time the separation hyper surface is probably not very smooth, a quality necessary for good generalization. In turn, this makes a linear separation of the data perform quite satisfactory relatively to a non smooth hyper surface with large linear regions that may look like a very noisy linear boundary.

The observation that the use of a linear kernel when applying the SVM model to construct a classification system or the use of an architecture that has no hidden layers for a neural network classifier can present results that are similar or even better to a well tuned non linear model of the appropriate modeling scheme, is a key point that it worths a little more examination. By avoiding the technique of feature mapping in a higher dimensional space but instead working in input space and still getting better results on a classification problem can only mean one disappointing situation: feature mapping is done poorly. On the other hand, when dealing with the RBF kernel in SVMs, feature space can be proven to be of infinite dimension and does not constraint the production of any appropriate feature. So feature dimensionality is not an issue. The only possibility left is the situation that some cases are discriminated by some features that are not really helpful for discriminating among other cases spanning different subspaces in feature space. So it would be rational to detect the cases that get a discrimination advantage from one large feature space (like the feature space of a SVM with RBF kernel) and discriminate the rest of the cases in a very different but constrained feature space. By following this methodology, confusing cases are constrained from preventing a smooth generalization boundary to be formed out of reliable data while at the meantime these less reliable data cases are processed explicitly by polynomial features formation. The concept is illustrated in diagram 1 while the details of the strategy will be more obvious in the third part that describes the algorithm.

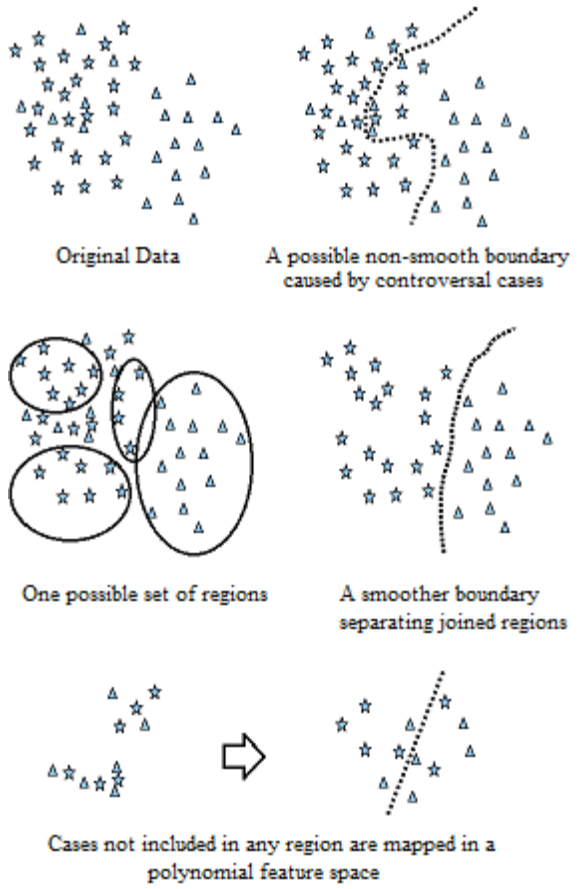

Fig. 1. Controversial cases decrease smoothness of the classification boundary which in turn reduces generalization. A boundary calculated based on highly reliable regions of classes tends to be smoother. The cases that are not included in any region are mapped to a polynomial feature space for a separate classification model.

In addition, another characteristic of data distribution is considered by the suggested methodology: data clustering. Data belonging to the same class may be distributed in considerably sparse clusters. As long as these clusters are at least loosely defined, they are discovered by the suggested algorithm because of its natural ability to identify data clusters. Based on the above discussion it may seem rational to cluster the training set in a way that the training examples that require different or even conflicting generalizations are assigned to a different modeling cluster. Classifying a new case is becoming a two stage process, first recognizing the cluster a specific data set belongs to and then applying the correct model for separation.

III. Region based support vector machine algorith

Having a number of machine learning models derived during the training phase that can be used depending on the test case under consideration seems very promising. The selected model for each occasion is based on its suitability to operate on the subspace of the problem that the test case belongs to. However, classifying the cases to problem subspaces and training a model for each subspace is by itself a modified version of the original problem. There are two major problems that one has to overcome in order to be able to deal with this angle of view of the classification problem. Making the clustering according to input space metrics will not help since the problem described in the previous section is laying in feature space mapping, so using input space based clustering is not an option. What is more, as explained above, splitting the feature space into subspaces and maintaining the ability to distinguish what model to use for a specific unknown input case, it is a classification problem of its own merit.

Regional SVM algorithm deals with these two problems by using multiple SVMs decision boundaries to form unified areas (regions) that include example cases belonging only to one class and exactly zero example cases belonging to other class(es). This results in the formation of confident positive regions and confident negative regions since numerically the classes of an SVM dichotomization are either +1 or -1. Positive regions include only a number of positive class cases and respectively negative regions include only a number of negative class cases. To be more specific, every cluster is defined by the union of the spaces that a group of SVM classifiers learn to classify as class +1 (positive class region) or class -1 (negative class region). By detecting one class region after the other and the respective data cases that define them, the algorithm builds up a set of negative and positive class cases that can be used to train an SVM model that uses RBF kernel to construct a generalizing boundary of classification. The rest of the cases that during the calculation of the SVM regions are left orphan, meaning they are not part of a region, are used to construct a second model that will try to discriminate between the classes by using a reduced feature space kernel (polynomial kernel). This reduced feature space kernel will try to create polynomial features of the input space and is different from the spatial concept that an RBF kernel is using through the Euclidean distance from selected support vectors. As anticipated, a neural network with a hidden layer is capable of applying the concept of the creation and fitting of polynomial factors of the input space and so serve the role of replacing the polynomial kernel SVM, although the latter is preferred due to lower execution times when the data set input size is not large. Evaluating regions of positive and negative cases provides the means to solve the problem of recognizing which model is appropriate for an input case: first investigate whether the case falls into any of the regions and if so apply the RBF kernel SVM, else use the polynomial kernel SVM.

In order to calculate positive and negative regions present in the data set, a genetic algorithm is put to work with the objective to evolve a population of representations of binary decisions of whether to include or not the individual cases of the given data set in the training of multiple SVMs. In this specific case (PID dataset) 3 SVMs are used to be trained over evolving data subsets. After the training of the SVMs using the data set represented by an individual from the genetic algorithm population, the union of the space of the positive (or the negative) side of the boundaries of the SVMs is checked to detect whether it includes a satisfactory number of only positive (or negative) examples of the dataset. If it does, the corresponding data cases are removed from the current data set and the region is saved in memory. A region is defined by its sign (positive or negative class), by the set of the SVMs whose unions of boundary areas define the region and by the data cases that are found in the region. It must be noted that all positive regions are calculated first and when it becomes very hard for the genetic algorithm to detect more positive regions the whole process is repeated again so as to calculate the possible negative regions. Every time a region is detected the enclosed data cases are extracted from the search and the algorithm is initialized (initial random population generation) to the new smaller data set. The fitness function of the genetic algorithm has to reflect the necessity to include only single class cases. This is accomplished by constructing a fitness function that depends on the ratio of the defined region's enclosed cases belonging to the targeted class over the enclosed cases belonging to the undesired class.

When the positive and negative regions are collected they will be used in a twofold way. First, all data cases included in both positive and negative regions are combined to produce a somehow reliable data set that is used to train a SVM with RBF kernel. What is expected as a result is a model that has smooth boundary and generalizes well over the data that falls into the constructed regions. To determine whether a test data case falls into one of the regions learned is just the result of classification of the case with the SVMs that define the regions one at a time. If a case is classified as positive by all the SVMs defining a positive region then it belongs to that area by definition. The analogy holds for a negative region. All data left out (not belonging to a region) is collected and a data set is constructed that will train a polynomial kernel SVM. A priori we expect this data set to be hard to separate, which means that results are expected to be lower that the results achieved by the RBF model. The important thing is that when the two models are combined to a unified classification system one should get significantly better results compared to the case of training one model for the whole original dataset.

The following steps define the algorithm's major functionalities:

1) Initialize the genetic algorithm with a population of N individuals with random genes and train initial SVMs accordingly. For each population individual there are two sets of information: the training sets that are used by the RBF

SVMs to create combined regions of classification and the actual SVM set.

$$P=\{[TRsets_i, TRsvms_i]\} \quad , \quad i=1,2,3,\ldots, N$$
$$TRsets_i=\{x_1, x_2, x_3, x_4 \ldots x_m\} \quad , \quad m=|Training\ Set|$$
$$0<x_k\le j \qquad j=|SVM\ classifiers|$$
$$TRsvms_i=\{svm_1, svm_2 .. svm_j\}$$

$$x_k=\begin{cases} 0 & TrainingSet[k]\notin p_i \\ 1 & TrainingSet[k]\in TrainingSetSVM_1 \\ 2 & TrainingSet[k]\in TrainingSetSVM_2 \\ . & \\ . & \\ j & TrainingSet[k]\in TrainingSetSVM_j \end{cases}$$

2) Fitness Function reflects the search for solely positive regions. Each individual's fitness function is the number of positive cases that are classified as positive by all of its SVM models over the number of negative cases that are classified as positive by all of its SVM models.

$$f(p_i)=\frac{|Cases^+|}{|Cases^-|+0.001}$$

$$Cases^+\in\{PositiveRegion_{SVM1}\cap \ldots \cap PositiveRegion_{SVMj}\}$$
$$Cases^+\in\{Positive\,Class\}$$
$$Cases^-\in\{PositiveRegion_{SVM1}\cap \ldots \cap PositiveRegion_{SVMj}\}$$
$$Cases^-\in\{Negative\,Class\}$$

3) The population is evolved through crossover and mutation by maximizing fitness function $f(p_i)$. After a genetic operator is applied to an individual its corresponding SVMs must be trained again.

4) If after an epoch an individual has a high fitness function (meaning zero negative class cases) and the number of the positive cases is more than or equal to 5% of the total positive data set examples, then the region is saved and the included cases are removed from the total training set. The process is repeated from step (1).

5) When the algorithm cannot detect any more valid positive regions, search for positive regions is ended and steps 1 to 4 are repeated for the negative regions with the reversal of the fitness function to reflect the goal of searching negative regions.

6) All cases included in all regions (either positive or negative) form a training set that is used to train the main RBF SVM. All left over cases not belonging to a positive or a negative region are used to train a polynomial kernel SVM.

7) During the classification phase, the unknown case is passed through the RBF SVMs of the detected region (not the main RBF SVM) and if it is found to belong to any of these regions it is classified with the main RBF SVM. Otherwise it is classified by the polynomial kernel SVM.

IV. Results

The suggested algorithm was used to divide the data set to two subsets: the one that is used to train an RBF support vector machine and the one that is used to train a polynomial support machines. The complete data set consists of 500 normal cases and 268 abnormal cases. Region based SVM algorithm results in the creation of a RBF training set of 376 normal and 177 abnormal cases and of a polynomial training consisting of 124 normal and 91 abnormal cases. In turn, these training sets are used to train both SVMs by a 5-fold cross validation technique. Each validation set pair is shown in Table 2 and the final results are shown in Table 3.

RBF SVM				**POLYNOMIAL SVM**			
Training Set Norm	*Training Set Abnorm*	*Test Set Norm*	*Test Set Abnorm*	*Training Set Norm*	*Training Set Abnorm*	*Test Set Norm*	*Test Set Abnor*
131	123	245	54	62	54	62	37

Table 2. Cross Validation Data Sets' sizes

	RBF SVM		**POLYNOMIAL SVM**		**Overall Final Test Results**	
Val/ion Set no	***Normal Cases Success Rate (%)***	***Abnorm Cases Success Rate (%)***	***Normal Cases Success Rate (%)***	***Abnorm Cases Success Rate (%)***	***Normal Cases Success Rate (%)***	***Abnorm Cases Success Rate (%)***
1	98.38	86.94	67.75	64.87	**83.06**	**82.41**
2	83.68	92.6	64.52	67.57	**79.98**	**82.41**
3	85.3	92.6	70.97	64.87	**82.41**	**82.41**
4	86.94	90.74	64.52	70.27	**82.41**	**81.31**
5	88.57	88.89	64.52	72.97	**83.71**	**82.41**

Table 3. Final results on various validation test sets. The overall success rate is calculated using the success rates of both the RBF and the polynomial SVMs.

V. Conclusions

The overall results are compared with the ones of several previous studies shown in Table 1 and a small improvement on the average performance is achieved with the suggested algorithm. It is important to note that the suggested algorithm is able to outperform much more complex algorithms like

ARTMAP-IC and achieves satisfactory performance while avoiding excess tuning. Another important attribute of region based SVM algorithm is that it can be applied without further considerations or modifications to any "hard" classification problem that seems difficult to solve with high successful classification rates.

On the other hand there are some limitations in applying the suggested algorithm. The most obvious one is the limitation raised by the data set size. A small dataset is not eligible for solving with the presented algorithm because of the algorithm's natural approach to divide the dataset to clusters which in turn reduces the size of the test sets making the test phase unreliable, prone to over fitting or even unfeasible when just a bunch of test examples are available. Consequently the available dataset must comprise of at least some hundred examples. Another issue that region based SVM must deal with is the execution time of the genetic algorithm when the dataset is large. Special programming skills must be used to improve performance with the use of parallel execution being the most efficient approach to follow. Finally the algorithm's performance was not tested on multi class problems.

References

[1] Carpenter, G.A., Markuzon, N., "ARTMAP-IC and medical d0iagnosis: Instance counting and inconsistent cases", Neural Networks, 11:323-336, 1998.

[2] Carpenter, G.A., Tan, A.H., "Rule extraction: From neural architecture to symbolic representation", Connection Sci. **7** 3–27, 1995.

[3] Deng, D., Kasabov, N., "On-line pattern analysis by evolving self-organizing maps", Proc. of the 5th Biannual Conference on Artificial Neural Networks and Expert Systems (ANNES), Dunedin, November, pp.46-51, 2001.

[4] Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., Johannes, R. S., "Using the ADAP learning"

[5] Kamer Kayaer, Tulay Yildirim , "Medical Diagnosis on Pima Indian Diabetes Using General Regression Neural Networks", 2003.

[6] http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes