Automatic Weight Estimation Method for Multiple SVMs in Software Sensors Systems

Plamena Ts. Andreeva[1] and Svetla I. Vassileva[2]

***Abstract* – An automatic weight estimation method for multiple support vector machines (SVM) is considered, which takes advantage from the discriminative classification based on ROC analysis. The software sensors systems support the hardware systems and take benefit of the modelling process estimating the important variables. The achieved results are presented and future trends are given.**

***Keywords* – Weight estimation, Prediction accuracy, Support Vector Machines, Software sensors systems.**

I. INTRODUCTION

The machine learning (ML) technology provides systems with new tools able to improve process supervision. Thanks to Internet resources, a remote model can efficiently run a database diagnosis. An useful tool for processing information and analyzing feature relationships is data mining (DM) technique [1, 2]. The aim is to achieve fast and simple learning models that result in small rule bases, which can be interpreted easily. The wide used support vector [3, 4] machines (SVM) are effective tool for optimal classifier capacity tailored on the given task problem. In this particular study SVM are explored and evaluated by the test accuracy based on receiver operational characteristic (ROC) curve analysis. The automatic weight estimation method for multiple SVM is considered in order to conduct feasible classification and acquire accurate prediction in software sensors systems. For training the model easily the "one vs. all" strategy is used. This leads to improved initial accuracy.

Software sensor systems are applicable to linear and non-linear systems, when uncertainty or incomplete information is available. The main idea is that the efficiency of hardware sensors is complemented by software sensors [5] which combine the information from the sensor network with a process model in order to predict some key-process variables which are generally not available on-line.

The support vector machines [6] have been a promising tool for classification and regression. Its success depends on the tuning of several parameters which affect the generalization error. A popular approach is to approximate the error by a bound that is a function of parameters. Then, we search for parameters so that this bound is minimized. In the context of medical diagnosis, the extraction of statistically independent components has been proposed as an objective of early sensory processing, finding the important attributes and primary dependencies.

II. PROBLEM STATEMENT

The automated acquisition of knowledge by machine learning approach is an active area of current research in Artificial Intelligence [7]. ML studies automatic techniques to make accurate predictions based on past observations. There are several multi-class classification techniques: Support Vector Machines (SVMs), decision trees, etc. [8, 9]. Nevertheless, building a highly accurate multi-class prediction is certainly a difficult task. Various systems of multiple classifiers have been proposed in the literature [10].

There is a scope of many diseases with the same symptoms, and also the degree or level of the symptoms is absent from the knowledge. To overcome the first problem, the knowledge engineer should design the knowledge base with more specific rules. For rules with identical symptoms, some sort of measures of coupling between the antecedent and the consequent clauses are to be devised. This measure may represent the likelihood of the disease among its competitive disease space. For example, if the diseases are seasonal, then the disease associated with the most appropriate season may be given a higher weight, which, in some ways should be reflected in the measure. The second problem, however, is more complex because the setting of the threshold level at the symptoms to represent their strength is difficult even for expert doctors. In fact, the doctors generally diagnose a disease from the relative strength of the symptoms but quantification of the relative levels remains unsolved.

The level or strength of the facts submitted may not conform to their actual strength, either due to media noise of the communicating sources of data or incapability of the sources to judge the correct level/ strength of the facts. Several mathematical models were developed in order to optimize the information provided by the sensor network for the studied process.

III. THEORETICAL BACKGROUND

The SVM can be used to learn highly accurate models from data. It is based on quadratic optimization [11] of convex function. This is realized by nonlinear mapping using so-

[1]Plamena Ts. Andreeva is with the Department of Knowledge Based Systems at the Institute of Control and System Research, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria, E-mail: plamena@icsr.bas.bg

[2]Svetla I. Vassileva is the head of the Department of Knowledge Based Systems at the Institute of Control and System Research, Bulgarian Academy of Sciences, 1113 Sofia, Bulgaria, E-mail: vasileva@icsr.bas.bg

called kernel functions. Via normalizing any uniformly separating hyperplane can be defined by the property:

$$\mathbf{w}^T\mathbf{x}+b \geq 1, \text{ when } \mathbf{x} \text{ is in } C_1 \text{ and}$$
$$\mathbf{w}^T\mathbf{x}+b \leq -1 \text{ for } \mathbf{x} \in C_2 \quad (1)$$

Given a separating hyperplane α which satisfies conditions (1), for the distances between α and the points $\mathbf{x}_i$, we have:

$$dist(\mathbf{x},\alpha) = \frac{\left|\mathbf{w}^T\mathbf{x}+b\right|}{\|\mathbf{w}\|} \geq \frac{1}{\|\mathbf{w}\|} \quad (2)$$

and the optimal hyperplane is the one with maximal 1/**w** that guaranties an optimal margin of width 2/**w** because the equality in (2) is reached at least for one point in C_1 and at least one point in C_2. By maximizing that distance we minimize the following dot product $\frac{1}{2}\mathbf{w}^T\mathbf{w}$. This represents the quadratic constrained optimization problem.

A trained SVM can be used for classifying an unknown vector by applying the following criterion which depends on the sign of the expression:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \quad (3)$$

In the sums above it holds $\alpha_i > 0$ only for the support vectors hence they play the main role during the usage phase. After calculating the α_i we can find the weight vector **w** and the bias *b* we need by the formula:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \quad (4)$$

The loss functions can be incorporated into the SVM training problem. Unlike the naturally occurring high-dimensional measurements (as visual images), medical sensor's measurements are not easily interpretable by human observers. For a learning machine there is no fundamental difference between these types of data. In this domain, machine intelligence will prove to make sense out of the data.

Support vector machines are not preferred in applications requiring great classification speed, due to the large number of support vectors. To overcome this problem we devise a primal method with the following properties: a) the basis functions are not connected with the concept of support vectors; b) set of kernel basis functions with specified maximum size are found to approximate the SVM primal cost function well; c) it is efficient and roughly scales as $O(n.d_{max}^2)$ where *n* is the number of training examples; and d) the number of basis functions for an accuracy close to the SVM accuracy is usually far less than the number of SVM support vectors.

A. Accurate prediction

We consider two measures for accurate prediction: test error numbers — the number of misclassifications on independent test samples, and the error numbers of 10-fold cross validation. The inference of diagnosis based on kernel function is equivalent to assigning a particular loss function to the errors: there are two types of error possible: of commission and errors of omission. The costs of making these two types of error are not necessarily equal, and depend on the relative cost (used as weights) of an erroneous prediction compared to missing a interesting correct prediction. Generally we wish to make the decision which minimises the expected loss.

Classification can be tuned to find rules with different loss functions by changing the selection criteria in the validation set. An approach to improve accuracy involve fold prediction. When seeking to solve the constrained optimization problem, asking for small $\mathbf{w}^T.\mathbf{w}$ is like "weight decay" in Neural Nets and like Ridge Regression parameters in Linear regression and like the use of Priors in Bayesian Regression-all are signed to smooth the function and reduce overfitting.

B. Precision measure

Nowadays it is accepted that there are no algorithms able to make exact classifications in a general domain. Local information is extracted from features associated to each local structure and provides information about its type of structure. Contextual information is incorporated by taking into account the relative spatial distribution of these local features. Finally global information is obtained as a result of considering the local features over wide neighborhoods embracing the whole feature set.

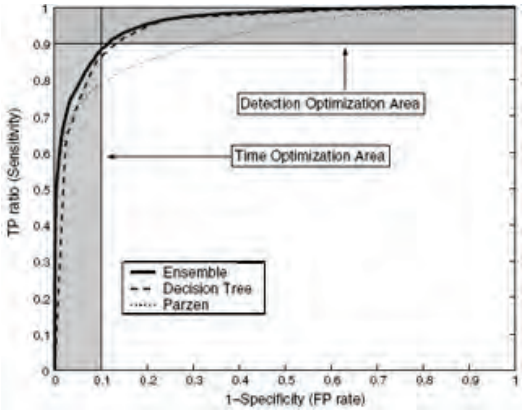

Figure 1. An illustrative example of specificity versus sensitivity plotted for the three different classifiers models.

The receiver operational characteristic (ROC) curves are used to optimize the trade-off between false positive and false negative rates. On Fig.1 are shown three classification curves with fault positive (FP) rates versus true positive (TP). We apply the use of ROC curves to evaluate several classifier models, including classifier ensembles. By simple classifiers examination our results shows that classifier ensemble methods have substantial advantages over simple classifiers. The ROC curves for all classifiers were calculated on the testing set. The ensemble with the largest area is plotted bold.

The curves plot on Fig. 1 shows the error tolerance on the *X* axis versus the percentage of points predicted within the tolerance on the *Y*-axis. The resulting curve estimates the cumulative distribution function of the error. In our medical experiments these curves provide a much more compelling presentation of regression results than tables of squared errors.

The shaded vertical stripe shows an example of a desirable time-optimization area. Its width denotes the maximum FP rate we are prepared to accept. Its height measures the amount

of accuracy we are prepared to sacrifice. A classifier performs well if the error curve climbs rapidly towards the upper left-hand corner. In our case, SVM was hand tuned based on testing information to estimate the best possible performance. Model parameters were selected using cross-validation.

IV. Automated Weight Estimation

The mechanism developed for the medical system (sensor system) diagnostic is based on the principles of multi-criteria decision making and takes into consideration input from training cases. Multiple criteria are used to describe the previous quality. In this paper we explore multiple SVMs in order to achieve optimal precision. Here the goal is to produce a model, which predicts target value of data instances in the testing set which are given only the attributes. The key idea in our approach is not primarily to define more complex functions, but to deal with more complex output spaces by extracting combined features over inputs and outputs. For a large class of structured models, we propose an automatic weight estimation to learn mappings involving complex structures in polynomial time despite an exponential number of possible output values. We empirically evaluate our approach for a specific problem: prediction of heart disease.

We select features with high F-scores and then apply SVM for training/prediction. The procedure is summarized below:

1. Calculate F-score of every feature.
2. Pick thresholds by human eye to cut low and high F-scores.
3. For each threshold, do the following:
   a) Drop features with F-score below this threshold.
   b) Randomly split the training data Xtrain / Xvalid.
   c) Let Xtrain be the new training data. Use the SVM in b) to obtain a predictor, then predict X valid.
   d) Repeat the steps above five times, and then calculate the average validation error.
4. Choose the threshold with the *min* average validation error.
5. Drop features with F-score below the selected threshold.

In the above procedure, possible thresholds are identified by human eye. For the examined data set, there is a quite clear gap between high and lower scores (see Figure 3). We can automate this step by gradually adding high-F-score features, until the validation accuracy decreases.

To test the above described method we conduct several experiments in Weka [12]. The distribution of experiments cuts down the time the experiments take. The setup of a selected classifier can be loaded and saved from and to XML. The multidimensional models [13] are also created and their associations are used in specific applications. This is rather useful for transferring a classifier setup from the Weka Explorer over to the Experimenter without having to setup the classifier from scratch. A robust alternative to the binary format is the XML format for Internet access [14].

V. Experimental Results

In diagnosis applications the outcome may be the prediction of disease vs. normal or in prognosis applications. The input features may include clinical variables from medical examinations, laboratory test results, or other measurements. In this study we have used a set of very well-known databases from UCI repository. Separating the classes with a large margin minimizes the bound on the expected generalization error. In the case of non-separable classes, it minimizes the number of misclassifications whilst maximizing the margin with respect to the correctly classified examples. On Fig. 2 the errors from the default SVM classifier with parameter C=1 are depicted. For the positive diagnosis there are 31 mis-classifications out from the whole dataset (270 examples) which gives the 11,48% rate. After the automatic weight estimation the optimal classifier is obtained and the accuracy rises to 24 from the first class and 15 misclassification from the second. Unlike other algorithms, the method makes no assumptions about the relationships between a set of features (attributes) in a feature space. This allows us to identify and determine the most relevant features used in a model and their dependencies.

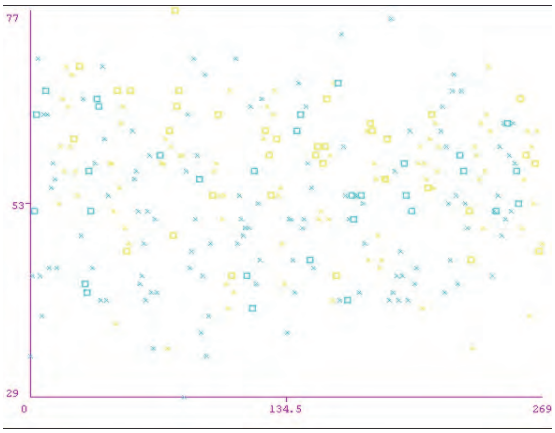

Figure 2. Error misclassifications results from the SVM classifier for the two diagnoses. The number of instances is on the X axis. The correct classified examples are depicted with cross and the misclassifications are given with squares – the yellow for negative diagnosis and the blue are for positive one.

In all our experiments, we train the algorithm, select parameters based on a validation set, and then report performance on an independent test set. SVM learning algorithm is performed on the data, generating a set of classifier models. Beyond the choice of model and loss function our method has only a single parameter to tune, namely the regularization parameter C. We train models with C ranging from 1 to 100, in powers of 2, all to precision 0.01. We then pick the best model based on the performance on the validation set, and report its performance on the test set.

Although the most regularized linear SVM is the best in this example, we notice the threshold 0,5 is determined by the two most extreme points in the two classes (see Fig. 3). For values larger than the initial value 0,1, the endpoint behavior depends on whether the classes are balanced or not. In either case, as α increases, the error converges to the estimated null error rate.

This same objection is often made at the other extreme of the optimal margin. Typically it involves more support points and tends to be more stable. Here the regularization forces more points to overlap the margin. We are able to implement our approach most easily with the “one vs all” strategy.

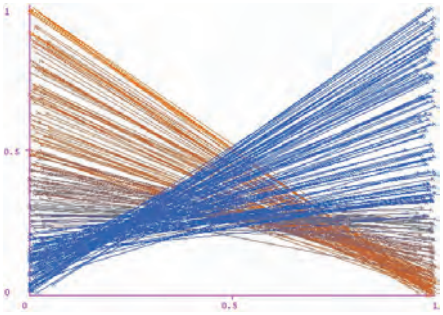

Figure 3. Threshold value around 0,5 and cost function for the tested binary classification. The blue are for the class “Yes”.

The output results from Weka built model are given on Fig. 4. The correct classified instances are 231 (85,56%).

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.87	0.127	0.885	0.83	0.887	“ yes”
0.873	0.17	0.845	0.873	0.859	“ no”

Figure 4. Screen output results from Weka built model for the tested classifier on the heart dataset.

Automated, the optimized SVM classification can generate models based on a number of criteria including precision/ recall, or correlation coefficient. In order to get statistically meaningful results, the default number of iterations is 10. The following results are generated. There are 30 result lines processed. In this experiment, each set of 10 cross-validation folds is averaged, producing one result line for each run. The percentage correct for each of the SVM schemes is shown in Fig. 4. The value at the beginning of the row represents the number of estimates that are used to calculate the standard deviation. The obtained model with weight estimation is given on Fig. 5. The positive diagnoses for class “Yes” are classified with precision of 0,885, and the negative are with lower precision which corresponds to the cost matrices. Our method enables the user to control the training time by choosing the number of basis functions to use.

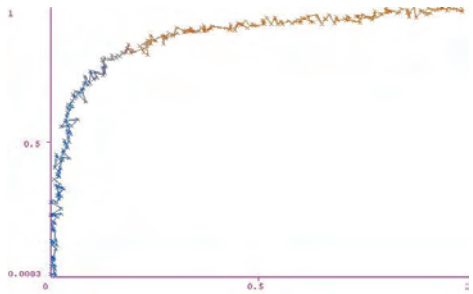

Figure 5. The ROC curve for the obtained optimal model with weight estimation.

The term statistical significance used in the previous section refers to the result of a pair-wise comparison of schemes using either a standard T-Test or the corrected resampled T-Test [2]. As the significance level is decreased, the confidence in the conclusion increases. In our experiment the resulting significance indicates the probability of positive diagnosis with attributes greater than selected. The accepted threshold is a significance level of 0.05. Such level is equivalent to requiring that the disparity would occur in no more than 1 in every 20 cases. The ranking test ranks the schemes according to the total number of significant wins and losses against the other schemes.

VI. Conclusion

The considered method applies multiple classifiers and on this basis makes the semi-automatic data analysis faster and easier. The advantages of this SVM method allow optimizing for different loss functions. The increased accuracy rate based on the automatic weight tuning method is a sure indicator of the importance of implementing DM systems. In our investigation we found that it is possible to build a flexible and accurate final model and present results to non-experts via ROC curves.

Acknowledgement

This research is supported by the NSRF of the Bulgarian Ministry of Education and Science as part of the Project MI - № 1509/2005 and in part by a research grant I - № 1406/2004.

References

[1] P. Andreeva, “Data Modelling and Specific Rule Generation via Data Mining Techniques”, Journal of E-learning and Knowledge Society, http://www.je-lks.it, 2007.

[2] I. Witten, E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2005.

[3] P. Bartlett, J. Sh-Taylor, “Generalization performance of support vector machines and other pattern classifiers”, in: B. Schölkopf, C. Burges, and A. Smola, (Ed.): *Advances in Kernel Methods*, MIT Press, MA 43-54, 1999.

[4] I. Steinwart, “On the influence of the kernel on the consistency of support vector machines”, Journal of Machine Learning Research, 2:67–93, 2002.

[5] S. Vassileva, X. Z. Wang, “Neural Network Systems and Their Applications in Software Sensor Systems for Chemical and Biotechnological Processes”, in Intelligence Systems: Technology and Applications, Ed. C.T. Leondes, CRC Press, USA, pp.291-335, 2002.

[6] T. Joachims, “Training linear SVMs in linear time”, KDD’06, August 20–23, 2006, Philadelphia, Pennsylvania, USA. Results and analysis. ACM SIGKDD Newsletter, 12(2):95–108, 2006

[7] S. Mark, *Introduction to Knowledge Systems*, Morgan Kaufmann, San Mateo, CA, Chapter 5, pp. 433-458, 1995.

[8] S. Hadjitodorov, L. Kuncheva, L. Todorova, “Moderate diversity for better cluster ensembles”, Information Fusion, 2006.

[9] T. Joachims, “A support vector method for multivariate performance measures”, in International Conference on Machine Learning (ICML), 2005.

[10] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley, 2004.

[11] B. Schölkopf, Smola A.J., *Learning with Kernels: Support Vector Machines, Regula-rization, Optimization and Beyond*, MIT Press, Cambridge, 2002.

[12] University of Waikato, ML Program WEKA, www.cs.waikato.ac.nz/ml/weka. 2007.

[13] I. Valova, A. Goknil, “Applicability of the Object-Oriented Paradigm in Multidimensional Models”, Int. Scientific Conference Computer Science’06, Oct. 12 - 15, Istanbul, Turkey, pp.190-195, 2006.

[14] S. Vassileva, G. Georgiev, S. Mileva, “Knowledge-Based Control Systems via Internet”, Part I. Appl. in Biotechnology Bioautomation, ISSN 1312-451X, pp. 37-48, 2005.