Available online at http://scik.org

J. Math. Comput. Sci. 2022, 12:50

https://doi.org/10.28919/jmcs/6647

ISSN: 1927-5307

IMPROVING THE ACCURACY OF THE MACHINE LEARNING PREDICTIVE MODELS FOR ANALYZING CHD DATASET

IVELIN GEORGIEV IVANOV*

College - Dobrich, Shumen University Konstantin Preslavski, Shumen, Bulgaria

Abstract: The problem to classify big data is an important one in machine learning. There are multiple ways to classify data, but the support vector machine (SVM) has become a great tool for the data scientist. In this paper we examine several modifications of the support vector machine algorithm that achieve better efficiency in terms of accuracy, F1 precision and CPU time when classifying test observations in comparison to the standard SVM algorithm. To make the modifications faster than standard SVM we use a special methodology which splits the input dataset into n folds and combine it with input data transformations. Each time we execute the process, one of the folds is saved as a test subset and the rest of the folds are applied for training. The process is executed n times. In the proposed methodology we are looking for the pair of subsets which produces the highest accuracy result. This pair is saved as an output SVM model.

Keywords: big data analysis; machine learning algorithms; classifications; support vector machines (SVM).

2010 AMS Subject Classification: 93A30, 68P01.

1. INTRODUCTION

In the field of the classification problems in the big data analysis, one of the most popular [11] and effective techniques is the support vector machines (SVM). The method has been proposed by Cortes, Vapnik [5], and it has been widely used in classification and regression problems in both application and research fields. Effectiveness of the SVM method depends on selecting parameters such as the optimal kernels and their parameters in the computer realizations. The robustness of

*Corresponding author

E-mail address: iwelin.ivanow@shu.bg

Received August 14, 2021

this method is not always satisfactory in this sense. Business applications need fast and as reliable as possible algorithms to solve the task of classifying big data sets.

SVM is a popular technique for classifying medical data sets [1,2,4,7,8,16,17]. Miyaki and coauthors [12] applied the regression trees to identify the best predictors of diabetes mellitus.

Kumari and coauthors [7] have investigated the standard SVM technique as a classification method on the data of Pima Indian diabetic patients available on http://networkrepository.com/pima-indians-diabetes.php

Kumar et al. [8] have proposed using a GA-SVM hybrid algorithm to optimize the parameters in the SVM in order to find the optimal subset of features.

Yilmaz et al. [17] have used the K-means clustering algorithm to pre-process data to remove noise and, then the SVM was applied as a classification method further.

The four bi-objective algorithms are employed to choose the least number of significant features with the highest classification accuracy using support vector machines in [2]. The proposed algorithms have been compared with other SVM classifiers for medical data sets (Table 10, in [2]).

In this paper we propose some algorithm modifications of the support vector machine method for accessing high-quality SVM model that achieves better efficiency in terms of accuracy, F1 precision and CPU time when classifying test observations in comparison to the standard SVM algorithm. A suggested approach for improving accuracy results of SVM classifiers is creating synthetic attributes which can be utilized in the modelling stage. Synthetic attributes can be created using other different features and data on subjects [3]. This innovation step in our modifications consists of the principal component analysis (PCA) transformation on the full given data set.

To validate the results obtained from the proposed SVM modifications and to assess the performance, the predictions of them are tested on the datasets available on the Internet and compared with the standard SVM algorithm. In this investigation we apply the Python information technology.

The method of support vector machines is used to solve the task of classifying big data sets. The purpose of this method [4] is to construct a dividing line between the two classes (when the observations in a set form two classes against a certain criterion) and two additional lines parallel to the dividing line. The extra lines separate the two classes so that observations remain on both sides of the extra lines. The support vector machine method seeks to maximize the distance between the dividing line and the closest observations of the two classes.

We consider the pairs (x_1, y_1), (x2, y2), ….., (xn, yn) in which xi is an n-size vector and yi takes values +1 or -1. In this way xi is interpreted with specific observations, and yi determines the belonging of observations to one of the two classes. We assume that the two classes can be separated by a straight line or a hyper line.

In its essence, the SVM method is an optimization approach for finding the equation of the separating object. The result of the optimization task are the two additional lines that maximize the distance between the two classes. This approach provides the only solution and leads to a very good predictive power of the built model. The equation of the separating object is searched by the type $w^T x + b$ with unknown w and b, where w, x are n-size vectors and b is an unknown number. There are many hyperplanes through which the two classes can be separated.

At the same time, the optimization model finds the only hyperline that sets the maximum distance to the two additional hyperplanes, which in turn are the boundaries between the two classes. The distance requested is expressed by $2 / \| w \|$. The optimization task is formulated as follows [9,15]:

$$\max_{w,b} J(w) = \frac{1}{2}\, w\, w^T = \frac{1}{2}\left|\left|w\right|\right|^2 \qquad (1)$$

subject to : $y_i\, [\, w^T\, x_i\, + b] - 1 \geq 0$, i=0,1,…,N .

The defined problem (1) is equivalent to the following optimization task involving the Lagrange multiplier $\alpha_i > 0$:

$$\max_{\alpha_i} L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i y_i \alpha_j y_j \; < x_i,\; x_j >$$

subject to: $\sum_{i,j=1}^{n} \alpha_i y_i \; = 0,\; \alpha_i \geq 0, i = 1,2, \ldots, n$

Therefore, the desired hyperplane is represented by:

$\sum_{i \in F}\, \alpha_i\, y_i < x_i\, , x > \; + b$,

where F contains the observations of the training subset, respectively Lagrange's non-zero multipliers. The vectors defining the separating hyperplanes are called support vectors.

The more general type of problem (1) is the following:

$$\max_{w,b} C \qquad (2)$$

subject to : $\frac{1}{\|w\|} y_i\, [\, w^T\, x\, + b] - C \geq 0$, i=0,1,…,N.

Taking C = $1 / \| w \|$ the problem (2) is equivalent to (1).

In the case where the two classes in the set of observations are not linearly separable, the partition hyperplane is searched as:

$$\sum_{i \in T} \alpha_i \, y_i \, K(x_i, x) \; + b \, ,$$

where $K(x_i, x)$ is called a kernel function. The four types of kernels are used from SVM (γ, r and ν are parameters) :

Linear Kernel : : $K(x_i, x_j) = x_i^T \, x_j$;

Polynomial Kernel : $K(x_i, x_j) = (\gamma \, x_i^T \, x_j + \, r \,)^{\nu} \, , \gamma > 0$;

Radial Basis Function (RBF) Kernel :

$K(x_i, x_j) = \, \exp\left(-\gamma \, ||x - y||^2\right) \, , \gamma > 0$;

Sigmoid Kernel:

$K(x_i, x_j) = \tanh(\gamma \, x_i^T \, x_j + \, r \,)$.

2. DATA AND METHODOLOGY

2.1. Several Modifications of the SVM method

The general approach to realize the SVM method via computer realization is as follows - the set of observations is divided into two subsets: a training subset (X_train, y_train) and a testing subset (X_test, y_test) in a random way, where the test subset includes 20% of the total number of observations in the respective dataset. The model of support vector machines is defined and built on the training subset and its adequacy checked on the test subset.

The general approach to realize the SVM method via computer realization is as follows - the set of observations is divided into two subsets: a training subset (X_train, y_train) and a testing subset (X_test, y_test) in a random way, where the test subset includes 20% of the total number of observations in the respective dataset. The model of support vector machines is defined and built on the training subset and its adequacy checked on the test subset.

The application of the support vectors method in the classification of a given set includes the selection of several parameters - the type of the kernel and the value of the constant C used in (2) [4,15]. In this paper, we will use models of SVM with both kernels - linear and a kernel, where the distance between observations is calculated by exponential function (rbf). Then, the built model on the training subset (model=SVC(X_train, y_train)) is evaluated on the test set using the

model.score() command. We run the algorithm 10 times to get the average of model.score(). We program this approach and call it a standard algorithm, we mean Algorithm AS.

We will also experiment with another algorithmic approach in which the observations of the set X and the dependent variable y are divided into n equal parts with the command KFold (len (X), n_folds = n). The operation of this command and its application in the analysis of observations and the role of this division to build an effective model are discussed in (Ivanov et al., 2018). The algorithm for realizing this approach will be called the modified support vector machines (MSVM) and it works for both types of the kernel. An important property of this algorithm is that it can work in a deterministic or stochastic mode, depending on the value of the logical variable. In this paper we apply the deterministic mode.

The values of the independent variables are loaded in the variable X, and the dependent variable is recorded as y. In advance, the set X (along with the variable y) is divided into n equal parts with the KFold command. By dividing the set X, we form the training set of (n-1) / n parts and the test set of 1/n part. This distribution is performed in n different ways, following the ascending order of the natural numbers.

Each time we build the pattern on the training set (modsvm=SVC(X_train, y_train)) and then test it on the test set. We calculate the percentage of match of the predicted values with the real values in each case by modsvm.score (X_test, y_test) and we select the highest percentage. The training set that is the most consistent match is used to train the method of SVM. For this method we calculate the percentage of match for the test set. The algorithm that implements this approach means Algorithm MSVM.

We will process the data set in advance to obtain more effective algorithms using two approaches. We will standardize the data set (scale the data in short) via the Python package sklearn.preprocessing. In the second approach we apply the principal component analysis method to convert the full data set. Note that all data is converted - the independent variables and the dependent variable.

Thus, in our experiments we apply the above two algorithms (with two types of the kernel) and the treatment in advance the given data set - see Table 1. The standard SVM algorithm is described by Algorithms AS applied on the original data. SVM algorithm modifications depend on algorithms (AS, MSVM), kernels (linear, rbf) and type of data (Original data, Standardized data, PCA transformed data).

Table 1 – SVM algorithm modifications

Algorithms	Kernels	Given Data Set
Algorithm AS	linear kernel	Original data
Algorithm MSVM	rbf kernel	Standardized data
		PCA transformed data

2.2. Performance criteria

We provide experiments with some data sets, which can be found on the Internet. To estimate the effectiveness of the considered algorithms we compare them in two aspects - the success rate of each, presented via score coefficient received by the command score and F1 score estimate, noted as "sc" and "F1sc" in the tables. The second aspect to compare is the CPU time (CPU) of the algorithm execution.

Important aspect for selecting the algorithm which performance best is using metrics and tests to compare how well the method performs with different algorithms. Furthermore metrics and quantitative criteria are also useful to compare performance among different algorithm modifications of SVM. Moreover, for different datasets, different algorithm modifications may give the best results.

In the below table, we visualize the dimensions of a confusion matrix which will help us define performance measures used for classification problems [14].

Table 2. The structure of the Confusion Matrix

	Predicted classes	
Actual classes	True Positives (TP)	False Positives (FP)
	False Negatives (FN)	True Negatives (TN)

The actual observations are presented in the first row of Table 2 and negatives refer to misclassified observations. In fact, True Positives (TP) gives us the number of observations which are observed positive and are actually positive, True Negatives (TN) are observed negative and actual negative, False Positives (FP) are observed negatives and actual positives, and finally, False Negatives are observed positives and actual negatives. This information is correct for each class of the data. Some of the criteria explaining the performance measure of a model are described in [14].

Accuracy as a performance measure looks at the sum of all correctly classified observations divided by the total number of observations in the subset:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}.$$

Sensitivity provides information on the accuracy of only positive predictions:

$$Sensitivity = \frac{TP}{TP+FN}.$$

Specificity represents the accuracy only of negative predictions:

$$Specificity = \frac{TN}{TN+FP}.$$

F1-score is a statistical test that takes into consideration both sensitivity and specificity. In other words, it is the weighted average of the sensitivity and specificity:

$$F1 = 2 * \frac{Sensitivity*Specificity}{Sensitivity+Specificity}.$$

Computational time can make a huge difference to choose the most effective solution for decision making involving enormous databases and complex computations. When using SVM classifier, it is expected the time needed for training and testing to depend on the values of the parameters. In our algorithms, we introduce the rule to execute the pre-processing of data set, including the independent variables.

3. Experimental Results. Classification the Cleveland Heart Disease Dataset

We apply the described algorithms to analyze medical dataset. Similar investigations are introduced in [6,10,13]. We use accuracy score, F1-score and time to evaluate all considered algorithm modifications of the SVM method. Confusion matrix is used as well to present the results from Example 1, where there are more than 2 classes.

Let's start by analyzing the Cleveland Heart Disease Dataset from the data repository http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/ (processed.cleveland.data). This is a statistic for patients with heart disease in different regions of the world. The set contains 303 observations with 14 characteristics. Their description can be found in a source described in the literature [9]. The dependent variable y describes the observations in five classes: a healthy patient (value 0) and a patient depending on the degree of disease, the y values can be 1,2,3,4. When classifying the observations of this set of data, we apply the two described algorithms AS and MSVM.

The set of data is split into two subsets - a training subset and a test subset, with 80% and 20% observations from the entire set (test_size = 0.2). To verify the achieved accuracy of the built model, three criteria are applied to the test subset - the value of the score parameter that determines the reliability of the built model (highest value 1); confusion matrix and the value of the F1-score. The results of the application of the two algorithms are presented in Table 3 on the Cleveland Heart Disease (CHD) data set.

Table 3 – The results of the experiments with the CHD data set, in which the classes are 1,2,3,4, respectively, for the patient depending on the degree of the disease and 0 for the healthy patient

	Algorithm AS			Algorithm MSVM		
	(test_ size=0.2, 10 runs)			(n_folds=5)		
	sc	F1sc	CPU	sc	F1sc	CPU
	CHD: linear kernel					
Normal data	0.59	0.56	7.2s	0.68	0.64	4.8s
Scale data	0.55	0.52	0.66s	0.63	0.60	0.04s
With PCA	0.68	0.67	0.03s	0.69	0.66	0.02s
	CHD: rbf kernel					
Normal data	0.53	0.53	0.02s	0.63	0.49	0.14s
Scale data	0.56	0.54	0.005s	0.69	0.65	0.06s
With PCA	0.63	0.62	0.04s	0.77	0.77	0.06s

In addition, we show the classification report, which presents the additional numerical values of the precision, recall, F1 score for the test subset. For example, the result of running the MSVM algorithm in the case of the linear kernel with PCA treatment on the data lead us to the following

classification report of the test subset (Table 3) and the corresponding confusion matrix presented in Table 4.

Table 4 – Results obtained via Algorithm MSVM with the linear kernel and PCA treatment on the data

Class	Precision	Recall	F1 score	Number of observations
0	0.87	0.97	0.92	35
1	0.50	0.18	0.27	11
2	0.40	0.40	0.40	5
3	0.33	0.80	0.47	5
4	0.00	0.00	0.00	4
av. value	0.66	0.70	0.66	60

The results in Table 4 show that this model does not classify patients by morbidity. The highest value in Table 3 for the score coefficient is 0.77, and from Table 4 the F1 score is very low for the classes of the patients, the built model classifies very well the first class (the healthy patients) and not so well the classes of the sick patients. For classes of sick patients (Table 5), misconceptions (located as non-diagonal entries) are more than properly classified observations (located diagonally).

Table 5 – Confusion matrix obtained via Algorithm MSVM with the linear kernel and PCA treatment on the data

	Predicted				
Actual	0	1	2	3	4
0	34	1	0	0	0
1	5	2	1	3	0
2	0	1	2	2	0
3	0	0	1	4	0
4	0	0	0	0	0

These results suggest that it could be more successful in classifying observations by dividing them into two classes - the healthy patient class and the sick patient class, covering the observations with corresponding values of the dependent variable from 1 to 4. We apply the two algorithms on

the Cleveland Heart Disease data set with two classes of observations. The results are presented in Table 6.

The same set (with two classes of observations) has been analyzed by other authors in the literature. Experimental results can be seen in Khanna et al., 2015. The SVM method is implemented with the linear and rbf kernels. The set of data is split into two equal parts - a training subset and a test subset with the procedure train_split (test_size = 0.2). The results obtained by the authors in Khanna et al., 2015 show that the score values on the test subset for both kernels are 84.8% - 87.6% and the F1-score is 0.85-0.88. The operation of the Algorithm MSVM with linear kernel (Table 6) reaches the values of 0.93-0.97 at n_folds = 5.

3. CONCLUSION

Big data analytic is moving towards intensively under multidisciplinary collaboration between statistics, optimization modelling and computer science. This leads to strong demand for algorithmic innovations to analyse big data sets. This paper has covered a specific family of the machine learning algorithms. The proposed SVM algorithmic modifications combine simplicity and efficient computer realization. They achieve better efficiency in terms of accuracy, F1 precision and CPU time. Experiments demonstrate that for different types of the data sets the proposed approaches may be more successful.

ACKNOWLEDGEMENTS

The paper was supported by the project RD- 08-92/ 01.02.2021 from the Shumen University, Bulgaria.

CONFLICT OF INTERESTS

The author declares that there is no conflict of interests.

REFERENCES

[1] F. Ahmed, N.A.M. Isa, Z. Hussain, M.K. Osman, Intelligent medical dis- ease diagnosis using improved hybrid genetic algorithm - multilayer perceptron network, J. Med. Syst. 37 (2013), 9934-9937.

[2] M. Alirezaei, S.T.A. Niaki, S.A.A. Niaki, A bi-objective hybrid optimization algorithm to reduce noise and data dimension in diabetes diagnosis using support vector machines, Expert Syst. Appl. 127 (2019), 47–57.

[3] A. Alsaffar, Empirical study on the effect of using synthetic attributes on classification algorithms, Int. J. Intell. Comput. Cybern. 10(2) (2017), 111–129.

[4] J. Cervantes, F.G. Lamont, A. Lopez-Chau, L.R. Razahua, J.S. Ruiz, Data selection based on decision tree for SVM classification on large data sets, Appl. Soft Comput. 37 (2015), 787–798.

[5] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20(2) (1995), 273–297.

[6] C. Gera, K. Joshi, A survey on data mining techniques in the medicative field, Int. J. Computer Appl. 113(13) (2015), 32–35.

[7] V.A. Kumari, R. Chitra, Classification of diabetes disease using support vector machine, Int. J. Eng. Res. Appl. 3(2) (2013), 1797–1801.

[8] G.R. Kumar, D.G. Ramachandra, K. Nagamani, An efficient feature selection system to integrating SVM with genetic algorithm for large medical datasets, Int. J. Adv. Res. Computer Sci. Software Eng. Res. 4 (2014), 272–277.

[9] D. Khanna, R. Sahu, V. Baths, B. Deshpande, Comparative Study of Classification Techniques (SVM, Logistic Regression and Neural Networks) to Predict the Prevalence of Heart Disease, Int. J. Mach. Learn. Comput. 5(5) (2015), 414–419.

[10] S.G. Kulkarni, M.V. Babu, Introspection of various K-Nearest Neighbor Techniques, UACEE Int. J. Adv. Computer Sci. Appl. 3 (2013), 103–106.

[11] W.-C. Lin, S.-W. Ke, C.-F. Tsai, Top 10 data mining techniques in business applications: a brief survey, Kybernetes, 46(7) (2017), 1158–1170.

[12] K. Miyaki, I. Takei, K. Watanabe, H. Nakashima, K. Watanabe, K. Omae, Novel statistical classification model of type 2 diabetes mellitus patients for tailor made prevention using data mining algorithm, J. Epidemiol. 12 (2002), 243–248.

[13] S. Myneni, V.L. Patel, Organization of biomedical data for collaborative scientific research: a research information management system, Int. J. Inform. Manage. 30(3) (2010), 256–264.

[14] D. Tripathi, D.R. Edla, V. Kuppili, A. Bablani, R. Dharavath, Credit Scoring based on Weighted Voting and Cluster based Feature Selection, Procedia Computer Sci. 132 (2018), 22–31.

[15] A. Tzotsos, D. Argialas, Support Vector Machine Classification for Object-Based Image Analysis, In: Blaschke T., Lang S. and Hay G.J. (eds) Object-Based Image Analysis. Lecture Notes in Geoinformation and Cartography. Springer, Berlin. (2008).

[16] B.P. Vrigazova, Detection of Malignant and Benign Breast Cancer Using the ANOVA-BOOTSTRAP-SVM, J. Data Inform. Sci. 5(2) (2020), 62-75.

[17] N. Yilmaz, O. Inan, M.S. Uzer, A new data preparation method based on clustering algorithms for diagnosis systems of heart and diabetes diseases, J. Med. Syst. 38 (2014), 1–12.