Comparative analysis of optimization models for the binary classification problem by the SVM method

A A Andrianova

Department of System Analysis and Information Technologies, Kazan (Volga Region) Federal University, Kazan, Kremlevskaya st. 35, Russia

E-mail: aandr78@mail.ru

Abstract. In the article several optimization models are considered for solving the binary classification problem by the support vector machine method. Mathematical statements of models differ in the form of the objective function and variables responsible for the magnitude of the classification error for each example of the training set. The main goal is to change the general principle of the formulation of the objective function, making the main criterion for minimizing the error function. Multicriteria statements of the optimization model are also considered. The article presents a conceptual and computational comparison of the suggested models of optimization problems with a computational experiment that estimates the time of the solution of the problem, the number of iterations, the received accuracy of classification. The comparison is performed on known data sets formed for the support vector machine method.

1. Introduction

The classification problem is one of the most popular big data problem. One of the ways to solve it, both in the case of binary classification and in the case of multiclass classification, is the Support Vector Machine method (SVM) ([1-3]). In this article for simplicity the binary classification problem will be considered.

The main idea of the SVM method consists in constructing the classes separating hyperplane (linear separation case) or the nonlinear surface (for kernel approach). To find it, the optimization problem is used, which is based on maximizing the width of the classes separating "strip". The constraint system of this problem includes inequalities for all training samples, which provides the conditions for belong to the "correct side" of the separation, taking into account the samples class labels and the possible classification error. In order to take into account classification errors, which is especially important when classes are not separable by the using surface, the objective function is complemented by a penalty term, which depends on the errors for each of the samples of the training dataset. Thus, we obtain an optimization problem of large dimension. In this paper other ways of formulating the objective function of SVM optimization problem will be considered. The main attention will be paid to the minimization of the different types of the classification error functions including multicriteria models. Several optimizing classification models will be formulated. For them, the results of an experimental study are given to obtain conclusions on the possibility of improving the accuracy of classification and reducing the computational complexity of the optimization problem. The experiments were performed on known datasets ([4]).
2. The classical C-SVM model and its modifications

Let us formulate the SVM optimization problem, which is commonly called as C-SVM model.

Suppose we have a training dataset of \( K > 0 \) samples \( \{x_i, y_i\} \) \( i = 1..K \) of two known classes. There \( x_i \in R_L \) - \( L \)-dimensional vector characteristics, \( y_i \in \{-1, 1\} \) is a class label. The binary classifier is a function \( F(x) = \text{sign}(w^T \phi(x) + b) \). The function \( \phi(x) : R_L \rightarrow R_M \) is a means of transition to another space of vector characteristics and is used for definition the kernel function \( K(x_1, x_2) = \phi(x_1)^T \phi(x_2) \) for samples \( x_1, x_2 \in R_L \). For the simplest case \( L = M \), \( \phi(x) = x \) and \( w \in R_L \). So, the problem is to find coefficients \( w \in R_L \) and \( b \in R_1 \).

Traditionally ([1,2]) the classical C-SVC method determines the separating hyperplane as a solution to the following optimization problem:

\[
\min_{w,b,\xi} \rightarrow 0.5 \|w\|^2 + C \sum_{i=1}^{K} \xi_i \tag{1}
\]

with constraints

\[
y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1..K
\]

\[
\xi_i \geq 0, \quad i = 1..K. \tag{2}
\]

The variables \( \xi_i \), \( i = 1..K \) allows to determine the error in determining the well-known \( i \)-th sample to another class, \( C > 0 \) - fixed penalty constant. The values of the classification error \( \xi_i \) can be interpreted as follows:

1) \( \xi_i = 0 \) if the class belonging is correct;
2) \( 0 < \xi_i \leq 1 \) if the sample belongs to the separating "strip";
3) \( \xi_i > 1 \) if the class belonging is incorrect.

So, for the correct samples \( w^T \phi(x_i) + b \geq 1 \) for \( y_i = 1 \) and \( w^T \phi(x_i) + b \leq -1 \) for \( y_i = -1 \) and this samples will be separated.

Since it is obvious that the sufficient accuracy of the classification method is achieved, in particular, by increasing the count of training sample \( K \), the optimization problem (1)-(3) will have a large dimension: \( M + 1 + K \) variables and \( 2K \) constraints in (2), (3).

The first term in the objective function (1) maximizes the width of the separating "strip". This term is main in classical model. The second term is auxiliary and represents a penalty for classification errors.

There is a somewhat different interpretation of the objective function (1) ([3]): the main term here is the error function \( \sum_{i=1}^{K} \xi_i \) and the term \( \frac{1}{2C} \|w\|^2 \) is a regularizer. Next, the following modifications of the objective function (1) are proposed, in which the error term is the main term.

Let us not be interested in the maximization of the separating "strip" and we will only consider the error function:

\[
\min_{w,b,\xi} \rightarrow \sum_{i=1}^{K} \xi_i \tag{4}
\]

with constraints

\[
y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1..K \tag{5}
\]

\[
\xi_i \geq 0, \quad i = 1..K. \tag{6}
\]

We call problem (4) - (6) Modif1.
As the objective error function, we can use not only the total error, but the maximum of errors. So, we formulate two modifications. The first modification of problem (1)-(3) with another error function (Modif2):

$$\min_{w,b,\xi} \rightarrow 0.5\|w\|^2 + C \max_{i=1..K} \xi_i \tag{7}$$

with constraints

$$y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1..K \tag{8}$$

$$\xi_i \geq 0, \quad i = 1..K \tag{9}$$

And modification of problem (4)-(6) with maximum of errors (Modif3):

$$\min_{w,b,\xi} \rightarrow \max_{i=1..K} \xi_i \tag{10}$$

with constraints

$$y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1..K \tag{11}$$

$$\xi_i \geq 0, \quad i = 1..K \tag{12}$$

From the point of view of applying another regularizer, we investigate the next modification (Modif4):

$$\min_{w,b,\xi} \rightarrow \sum_{j=1}^L |w_j| + C \sum_{i=1}^K \xi_i \tag{13}$$

with constraints

$$y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1..K \tag{14}$$

$$\xi_i \geq 0, \quad i = 1..K \tag{15}$$

A certain disadvantage of models (7)-(9), (10)-(12), (13)-(15) is the non-differentiability of the objective function, which complicates the methods of their solution.

The next modification is related to the application of the multicriteria approach. The formulation of problem (1)-(3) is connected with the application of two criteria: maximizing the width of the separating "strip" and minimizing the convolution of the classification errors. Actually, the problem (1)-(3) can be considered as a problem of minimizing the convolution function of these two criteria for obtaining the Pareto-optimal solution of the multicriteria problem. Moreover, the value of classification error for each sample of the training dataset can be considered as a separate criterion, which we want to minimize. Thus, in this case the problem can have $K+1$ criteria. Although such a detailed approach is not practical, due to the large number of criteria and, consequently, the complexity of the solution such multicriteria problem. Nevertheless, this approach is conceptually interesting enough.

Let us consider a model of a multicriteria optimization problem in which two variants of the aggregated classification error considered in the previous versions of models – the total error and the maximum error - will be considered as two criteria (Modif5):

$$\min_{w,b,\xi} \rightarrow \sum_{i=1}^K \xi_i \tag{16}$$

$$\min_{w,b,\xi} \rightarrow \max_{i=1..K} \xi_i \tag{17}$$

with constraints

$$y_i \left( w^T \phi(x_i) + b \right) \geq 1 - \xi_i, \quad i = 1..K \tag{18}$$

$$\xi_i \geq 0, \quad i = 1..K \tag{19}$$

The last modification is based on the separation of two types of classification errors. As stated above, a positive error value can have two values. When $0 < \xi_i \leq 1$, a sample fall into the separating
"strip". It is thus in a state of "uncertainty", although the classifier will refer it to that class whose boundary is closer. The second type of error occurs when \( \xi_i > 1 \) and is associated with a truly incorrect classification - assigning an object to another class. The second type of error is more serious, so the falling into "strip" is online identified when classifying objects and can then be further investigated by other methods.

To separate these two types of errors for each sample, let us define two variables, presenting an overall error in the form of their sum (\( \xi_i = \lambda_i + \eta_i \)) under conditions \( 0 \leq \lambda_i \leq 1 \) and \( \eta_i \geq 0 \). Thus, the following possibilities are obtained for error detection:

1) \( \xi_i = 0 \), consequently, \( \lambda_i = 0 \), \( \eta_i = 0 \);
2) if \( 0 < \xi_i \leq 1 \), we can get a few cases: \( 0 < \lambda_i \leq 1 \) and \( \eta_i = 0 \); \( 0 \leq \lambda_i \leq 1 \) and \( 0 < \eta_i \leq 1 \) on condition \( \lambda_i + \eta_i \leq 1 \); \( \lambda_i = 0 \) and \( 0 < \eta_i \leq 1 \).
3) if \( \xi_i > 1 \), we can get the following situations: \( \lambda_i = 0 \) and \( \eta_i > 1 \); \( \lambda_i > 0 \) and \( \eta_i > 0 \).

As you can see, a positive value of the variable \( \eta_i \) is necessary in case 3, which is associated with erroneous classification of the sample. In case 2 such a value of the variable \( \eta_i \) may be, but it will not be significant. Thus, in the last modification (Modif6) of the model, the same total error can be used as the objective function, but using weight coefficient \( C > 0 \), by which to determine the importance of each type of error: \( \min_{w, b, \lambda, \eta} \rightarrow \sum_{i=1}^{K} \lambda_i + C \sum_{i=1}^{K} \eta_i \) (20)

with constraints \( y_i (w^T \phi(x_i) + b) \geq 1 - \lambda_i - \eta_i \), \( i = 1..K \) (21)
\( 0 \leq \lambda_i \leq 1 \), \( i = 1..K \) (22)
\( \eta_i \geq 0 \), \( i = 1..K \). (23)

The model (20)-(23) has a more complex structure, because it doubles the number of variables for classification errors (the total number of variables \( L + 1 + 2K \)), and also increases the number of constraints (\( 4K \)). However, by increasing the coefficient \( C > 0 \), we can significantly reduce the values of large classification errors.

3. Experimental comparison of modifications of SVM optimization models

The experiment contains several typical problems generated by well-known datasets from samples (a1a, a2a, a3a, a4a, a5a, a6a, a7a, a8a, a9a, [4]). These samples containing some examples with the vectors of characteristics with size \( L = 14 \). For simplicity, we consider \( \phi(x) = x \), since, as experiments with the library of LibSVM ([5]) have shown, the use of other types of function \( \phi(x) \) on these datasets does not lead to an increase in accuracy.

Two series of experiments were conducted on different volumes of the training and test set. The first series of the experiment consisted of 100 tasks, generating 700 samples in the training set and 350 samples in the test set. The best value of parameter of the penalty coefficient \( C > 0 \) was chosen by the library's tools and a 350-sample validation set. So, a computational comparison of modifications of SVM models was carried out with the help of this value of the penalty coefficient. For the second series of experiments, which was conducted in the same scenario, the following sizes were selected: training set - 1200 samples, 420 samples in the validation and test set. A total of 20 tasks were considered. The absence of samples from the training set in the validation and test sets was guaranteed.
Used datasets a1a, a2a, a3a, a4a, a5a, a6a, a7a, a8a, a9a ([4]) make it possible to construct linear classifiers with an accuracy of 78-85%. Variations in the size of the training set and the use of the library’s tools LibSVM have shown the stability of this indicator. Therefore, there were two cases of used datasets that seemed sufficient to show the time dependencies and the resulting models accuracy.

To solve optimization problems (1)-(3), (4)-(6), (7)-(9), (10)-(12), (13)-(15), (16)-(19), (20)-(23) we used the Python tools and the scipy.optimize package. The solution of the multicriteria problem (16)-(19) was carried out using the method of concessions with the size of concession for the main criterion of 0.01.

The following tables shows the accuracy results (the percentage of correctly received responses for the test samples) and the time of the classical C-SVM model and all its proposed modifications.

In Table 1, the analysis of the accuracy indicators is aggregated for both series of experiments for all proposed modification models: the percentage of tasks in which the best accuracy is obtained in comparison with the classical C-SVM model ($P_{\text{modif}}$), the percentage of tasks in which the same accuracy is obtained in comparison with the classical C-SVM model ($P_{\text{same}}$), the average improvement in the accuracy in percent compared to C-SVM model ($P_{\text{acc}}$), the percentage of tasks in which some modification showed the highest accuracy among all the proposed models ($P_{\text{best}}$).

**Table 1. The accuracy of modifications.**

| Modification | $P_{\text{modif}}$ | $P_{\text{same}}$ | $P_{\text{acc}}$ | $P_{\text{best}}$ |
|--------------|-------------------|-------------------|-----------------|-----------------|
| Modif1       | 0.83%             | 98.36%            | 0.06%           | 20% (as Modif4) |
| Modif2       | 43.33%            | 13.33%            | 0.06%           | 10%             |
| Modif3       | 45%               | 13.33%            | 0.07%           | 13.33%          |
| Modif4       | 0.83%             | 98.36%            | 0.06%           | 20% (as Modif1) |
| Modif5       | 45%               | 13.33%            | 0.07%           | 33.33%          |
| Modif6       | 48.33%            | 2.5%              | 0.16%           | 23.33%          |

The time characteristics of constructing a classifier process for the classical C-SVM model and each of its modifications are presented below: the minimum ($T_{\text{min}}$), average ($T_{\text{avg}}$) and maximum ($T_{\text{max}}$) values of the total time of the computational process. Table 2 shows the results for a series of experiments for the training set with $K = 700$, in Table 3 - for the training set with $K = 1200$.

**Table 2. The classifier calculation time for $K = 700$.**

| Modification | $T_{\text{min}}$ | $T_{\text{avg}}$ | $T_{\text{max}}$ |
|--------------|-----------------|-----------------|-----------------|
| C-SVM        | 137.08 sec      | 186.76 sec      | 292.12 sec      |
| Modif1       | 128.93 sec      | 184.31 sec      | 286.37 sec      |
| Modif2       | 20.42 sec       | 24.44 sec       | 30.37 sec       |
| Modif3       | 10.57 sec       | 13.75 sec       | 24.97 sec       |
| Modif4       | 115.96 sec      | 184.27 sec      | 286.59 sec      |
| Modif5       | 172.63 sec      | 266.71 sec      | 354.56 sec      |
| Modif6       | 303.22 sec      | 978.34 sec      | 1516.92 sec     |
Table 3. The classifier calculation time for \( K = 1200 \).

| Modification | \( T_{\text{min}} \)         | \( T_{\text{avg}} \)         | \( T_{\text{max}} \)         |
|--------------|-------------------------------|-------------------------------|-------------------------------|
| C-SVM        | 805.44 sec                    | 1090.77 sec                   | 1944.66 sec                   |
| Modif1       | 760.42 sec                    | 1078.56 sec                   | 1889.73 sec                   |
| Modif2       | 98.6 sec                      | 109.7 sec                     | 117.71 sec                    |
| Modif3       | 55.08 sec                     | 57.44 sec                     | 60.94 sec                     |
| Modif4       | 767.44 sec                    | 1074.86 sec                   | 1892.37 sec                   |
| Modif5       | 999.51 sec                    | 1174.5 sec                    | 1929.58 sec                   |
| Modif6       | 2260.61 sec                   | 5308.46 sec                   | 6114.07 sec                   |

Thus, on the basis of the accuracy of the classifiers and the time of its construction, the following conclusions can be drawn:

1) The values of the indicator \( P_{\text{acc}} \) in Table 1 are not significant but show an improvement (on average) in the accuracy indicator of the constructed classifiers. Note that in several cases, the accuracy improvement reached to 5%.

2) Modif1 and Modif4 have almost identical accuracy values with the classic C-SVM model (98% coincidence) but have a slight advantage in time characteristics. And it should be noted that from the point of view of constructing the model as a problem of minimizing the error function with regularization, the type of regularizer used in Modif1 and Modif4 had practically no effect.

3) The error function in Modif2 and Modif3 had the form of an non-differential maximum function. Usually such functions have a high computational complexity. However, the experiment showed the advantage of this error function, since in almost 60% of cases the advantage or equivalence of the classifier accuracy was obtained, and from the Table 2 and Table 3 we can observe a significant reduction in the time of constructing the classifier.

4) The multicriteria modification Modif5 is more laborious due to the specificity of its solution in the form of two optimization problems, but the Modif5 obtained accuracy in 33.33% of cases \( P_{\text{best}} \) were the best among all modifications of SVM.

5) The best improvement in the quality of classification is achieved using Modif6, and in 23.33% of cases this modification gave the best accuracy. However, as can be seen from Tables 2 and 3, the complexity of the solution of this problem greatly increases due to, first of all, an increase in the number of model variables. So, for \( K = 700 \), the number of variables will be 1415 compared to the other models, in which there are 715 variables. For the case \( K = 1200 \), the number of variables is 2415 compared to 1215.

4. Conclusions
In general, the performed experiment shows the perspective of the approach to constructing optimization models based on minimizing the error function and developing this approach by using different types of error function. The peculiarities of the error functions can also contribute to the development of the applied optimization methods for large-dimensional problems of the type suitable for new SVM models, such as coordinate descent method using the decomposition approach, which is developed for some classical SVM models [6-8].

Acknowledgments
This work is performed by financial support of RFBR (project 16-01-00109).
References

[1] Burges C J 1998 A Tutorial on Support Vector Machines for Pattern Recognition Data Mining and Knowledge Discovery 2 (2) 121–167
[2] Cortes C and Vapnik V 1995 Support-vector network Machine Learning 20 273–297
[3] Munoz A 2014 Machine learning and optimization (https://www.cims.nyu.edu/~munoz/files/ml_optimization.pdf)
[4] Chang C C and Lin C J LIBSVMdata: Classification, regression and multi-label (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/)
[5] Chang C C and Lin C J 2011 ACM Transactions on Intelligent Systems and Technology 2 (3) 27
[6] Konnov I V 2015 Optimization Methods and Software 30(6) 1238-1254
[7] Chang K W, Hsieh C J and Lin C J 2008 Journal of Machine Learning Research 9 1369–1398
[8] Andrianova A A 2016 About decomposition approach for solving the classification problem IOP Conf. Series: Materials Science and Engineering 158 (1) 012006