IOP Conf. Series: Materials Science and Engineering **158** (2016) 012006 doi:10.1088/1757-899X/158/1/012006

About decomposition approach for solving the classification problem

A A Andrianova

Department of System Analysis and Information Technologies, Kazan (Volga Region) Federal University, Kazan, Kremlevskaya st. 18, Russia

E-mail: Anastasiya.Andrianova@kpfu.ru

Abstract. This article describes the features of the application of an algorithm with using of decomposition methods for solving the binary classification problem of constructing a linear classifier based on Support Vector Machine method. Application of decomposition reduces the volume of calculations, in particular, due to the emerging possibilities to build parallel versions of the algorithm, which is a very important advantage for the solution of problems with big data. The analysis of the results of computational experiments conducted using the decomposition approach. The experiment use known data set for binary classification problem.

1. Introduction

The classification problem is one of the most popular big data problems. Often their solution reduces to optimization problems. However, the traditional computational approaches for solving optimization problems is not effective because of the large number of variables and constraints. For such problems, the calculation itself is gradient of function is a time-consuming task. Even worse is placed when the methods of the second order are used. Therefore, decomposition approach for solving such problems is promising, as it allows to reduce the solution to a simple computational procedures, as well as makes it possible to build distributed and parallel algorithms for solving big data problems.

In [1] was considered decomposition approach to the construction of coordinate descent algorithms for problem $\min_{\boldsymbol{x}\in X}\mu(\boldsymbol{x})$ where objective function define as sum $\mu(\boldsymbol{x})=f(\boldsymbol{x})+h(\boldsymbol{x})$, $f(\cdot)$ is a smooth, but not necessary convex function in Euclidean space R_n, $h(\cdot)$ is convex, but not necessary smooth function what is more $h(\boldsymbol{x})=\sum_{i=1}^{m}h_i(\boldsymbol{x}^i)$, $X=X_1\times X_2\times\ldots\times X_m$, $\boldsymbol{x}^i\in X_i$ $i=1..m$. The partition of vector $\boldsymbol{x}\in R_n$ into components $\{\boldsymbol{x}^i\}_{i=1}^m$ determines the possibility of using the decomposition approach.

In this paper, the algorithm will be reformulated for the case of classification problem solving method of Support Vector Machine and held on the known problems of the analysis of computational experiment ([2]).

2. Algorithm for Support Vector Machine method with decomposition

Let us formulate the optimization problem of Support Vector Machines for construction of linear binary classifier ([3]).

Suppose we have a training set of $K > 0$ examples $(\boldsymbol{x}_i, y_i)$ $i = 1..K$ of two known classes. There $\boldsymbol{x}_i \in R_L$ - L-dimensional vector characteristics, $y_i \in \{-1;1\}$ is a class label. The binary classification problem is a problem of constructing a linear classifier as a hyperplane $\langle \boldsymbol{w}, \boldsymbol{x} \rangle = 1$ where $\boldsymbol{w} \in R_L$.

Traditionally ([3]) Support Vector Machine method determines the separating hyperplane as a solution to the following optimization problem:

$$\min_{\boldsymbol{w}} \to 0.5\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{K} \xi_i \tag{1}$$

with constraints

$$y_i \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle \geq 1 - \xi_i, \quad i = 1..K \tag{2}$$

$$\xi_i \geq 0, \quad i = 1..K. \tag{3}$$

The variables ξ_i $i = 1..\mathrm{K}$ allows to determine the penalty for the error in determining the well-known i-th example to another class, $C > 0$ - fixed constant.

Instead, the problem (1) - (3) usually its dual problem solves:

$$\max_{0 \leq \lambda_i \leq C,\, i=1..K} \to \sum_{i=1}^{K} \lambda_i - 0.5\sum_{s=1}^{K}\sum_{t=1}^{K} \lambda_s \lambda_t y_s y_t \langle \boldsymbol{x}_s, \boldsymbol{x}_t \rangle, \tag{4}$$

where λ_i, $i = 1..\mathrm{K}$ are dual variables, whereby their optimum values the vector coefficients hyperplane $\boldsymbol{w}$ can be obtained:

$$\boldsymbol{w} = \sum_{i=1}^{K} \lambda_i y_i \boldsymbol{x}_i. \tag{5}$$

Thus, we can solve instead of the problem (1) - (3) with a large number of constraints its dual problem (4) a large number of variables, but with simple constraints, and then to obtain the solution of primary problem (1)-(3) using formula (5).

The problem (4) has a form suitable for decomposition. The set $X = \{\boldsymbol{\Lambda} = (\lambda_1, \lambda_2 ... \lambda_K) \in R_K \mid 0 \leq \lambda_i \leq C, i = 1..K\}$ allows decomposed of the vector $\boldsymbol{\Lambda}$ on $m > 0$ disjoint groups of variables $\boldsymbol{\Lambda}^i$ $i = 1..m$ and their corresponding sets of X_i. Thus, we have

$$h(\boldsymbol{\Lambda}) = -\sum_{i=1}^{K} \lambda_i, \qquad f(\boldsymbol{\Lambda}) = -0.5\sum_{i=1}^{K}\sum_{j=1}^{K} \lambda_i \lambda_j y_i y_j < \boldsymbol{x}_i, \boldsymbol{x}_j >.$$

In paper [1] it was proved that the solution of problem решение задачи (4) $\boldsymbol{\Lambda}^*$ is a solution a mixed variational inequality $\sum_{i=1}^{m} \left\langle \boldsymbol{g}^i(\boldsymbol{\Lambda}^{i*}), \boldsymbol{Y}^i - \boldsymbol{\Lambda}^{i*} \right\rangle + \sum_{i=1}^{m} \left[h_i(\boldsymbol{Y}^i) - h_i(\boldsymbol{\Lambda}^{i*}) \right] \geq 0$ for all vector $\boldsymbol{Y}^i \in X_i$, $i = 1..m$. Here $\boldsymbol{g}^i(\boldsymbol{\Lambda}^i)$ is a vector with components of the gradient of the function $f(\cdot)$ corresponding group of variables $\boldsymbol{\Lambda}^i$.

After apply the methods for mixed variational inequalities problem we can using as a the direction of decrease of the function $\mu(\boldsymbol{\Lambda}) = f(\boldsymbol{\Lambda}) + h(\boldsymbol{\Lambda})$ in point $\boldsymbol{\Lambda}$ the vector $\boldsymbol{d} \in R_K$ which components are follows:

$$d_i = \begin{cases} Y_i(\boldsymbol{\Lambda}) - \Lambda_i, & \text{при } i = i_0 \\ 0, & \text{при } i \neq i_0 \end{cases}, \tag{6}$$

Here $\boldsymbol{Y}(\boldsymbol{\Lambda})$ is a solution with a predetermined $\boldsymbol{\Lambda}$ for a problem:

$$\min_{\boldsymbol{\Lambda}^i \in X_i,\, i=1..m} \to \sum_{i=1}^{m} \left(\left\langle \boldsymbol{g}^i(\boldsymbol{\Lambda}), \boldsymbol{Y}^i \right\rangle + 0.5\alpha^{-1} \left\| \boldsymbol{\Lambda}^i - \boldsymbol{Y}^i \right\|^2 + h_i(\boldsymbol{Y}^i) \right). \tag{7}$$

Obviously, the problem (7) can be decomposed into the m problems of follow form:

$$\min_{\boldsymbol{\Lambda}^i \in X_i} \rightarrow \left\langle \boldsymbol{g}^i(\boldsymbol{\Lambda}), \boldsymbol{Y}^i \right\rangle + 0.5\alpha^{-1} \left\| \boldsymbol{\Lambda}^i - \boldsymbol{Y}^i \right\|^2 + h_i(\boldsymbol{Y}^i) \quad i = 1..m. \qquad (8)$$

These problems may be solved independently including their parallel solution.

For determining the direction of decrease of function $\mu(\boldsymbol{\Lambda})$ according formula (6) the index i_0 should also be specified. For this index the following condition must be performed:

$$\left| Y_{i_0}(\Lambda) - \Lambda_{i_0} \right| > \delta. \qquad (9)$$

Here $\delta > 0$ is the parameter that provides a choice of reasonably good direction of decrease. If such an index does not exist, the value of parameter decreases (for example, values of parameter δ we can choose from sequence $\{\delta_i\}$: $\lim_{i \to \infty} \delta_i = 0$).

The search of index i_0 can be carried out from sequentially view the decomposed set of variable (we call it decomposed block). The origin of this search can be random. Then we must to solve for decomposed block the problem (8) and to check the condition (9). After obtaining the index i_0 all remaining unsolved problems (8) for other decomposed blocks will not be further considered.

Then we must calculate the step $\tau > O$ on the found direction $\boldsymbol{d} \in R_K$. In paper [1] it was proved that is sufficient to provide a reduce of the objective function is not less than an amount proportional to the $\left| Y_{i_0}(\boldsymbol{\Lambda}) - \Lambda_{i_0} \right|^2$. Such step can be easily found using finite procedures. Note that the increase of value $\mu(\boldsymbol{\Lambda})$ is possible only at i_0 variable component from vector $\boldsymbol{\Lambda}$. Therefore, the step on descent direction can be found if minimized only those terms of function $\mu(\boldsymbol{\Lambda})$ which depends on the variable $\boldsymbol{\Lambda}_{i_0}$. It also reduces the computational complexity. So, we move on to the next iteration point:

$$\boldsymbol{\Lambda}^{new} = \boldsymbol{\Lambda} + \tau \boldsymbol{d}.$$

In paper [1] it was proved that this process converges to the solution of the problem (4). Calculations can be stopped according to heuristic conditions about values of the objective function in neighboring iterative points.

3. Experimental study of the effectiveness of decomposition

The experiment contains several typical problems generated by well-known learning sets from samples (a1a, a2a, a3a, a4a, a5a, a6a, a7a, a8a, a9a, [2]). These samples containing an examples with the vectors of characteristics with size $L = 14$.

The solution was carried out with the help of three variants of the algorithm - sequential version without decomposition, sequential version with decomposition and parallel version using OpenMP.

The experiment was provided by a series of random samples from the data set with sizes $K = 100, 250, 500, 700, 1000, 1500$. We solved more than 100 different problems. The auxiliary problems (8) was solved using a gradient projection method.

The tests looked at various options of decomposed blocks of contiguous variables. Size of decomposed blocks took up the same (except for the last block, which contains all the remaining variables).

The following table shows the average time (in seconds) for solving the problem (4) by sequential version without decomposition (T_1) and with decomposition (T_2). Also in Table 1 the count of example in sample (K) and the size of decomposed block (S) are listed.

As seen from Table 1 for small amounts examples in the sample the decomposition approach has a little effect. When the sample size $K = 700$ difference becomes noticeable and in the future it begins to grow very quickly. Thus, the decomposition approach can give a great effect for serious practical problems.

Table 1. The time without decomposition and with it.

K	S	T_1	T_2
100	2,10,50	<<1 sec	<<1 sec
250	100	12 sec	4 sec
500	250	75 sec	15 sec
700	400	298 sec	48 sec
1000	500	30 min	140 sec
1500	500	1 hour 45 min.	346 sec

Then give the solution time performance when using the same decomposition in sequential (T_3) and parallel (T_4) versions of the algorithm. We note immediately that the size $K \le 500$ of the training set for the parallel version provides the benefits only for a small amount of decomposed block. In other cases, there was even the same time sequential and parallel versions of the algorithm. This is probably due to overhead costs of the time for parallelization problems.

Table 2. The time with different sizes of decomposed blocks.

K	S	T_3	T_4
1000	10	60 sec	18 sec
1000	50	21 sec	5 sec
1000	100	13 sec	8 sec
1000	200	21 sec	10 sec
1000	300	32 sec	28 sec
1500	100	39 sec	18 sec
1500	200	37 sec	14 sec
1500	300	48 sec	38 c sec
1500	500	254 sec	204 sec

As can be seen from Table 2, the use of parallel version of the algorithm have advantages in almost all values of the decomposed block sizes, but it becomes sensible when the size of at least 5 times smaller than the training set size. Another interesting conclusion is that with decrease of the decomposed block size the time is decreased for both versions of the algorithm only to a certain moment. Further reduction of the size decomposed block increases the computational time. So, it is clear for $K = 1000$ in the sequential version of the algorithm between 50 and 100 variables in the block. The computational time is worsened in 1.6 times. Parallel version worsened the time later – between 10 and 50 variables in a block. For $K = 1500$ we observe the same behavior between 100 and 200 variables in a decomposed block.

4. Conclusions

In general, the experiment allows us to conclude that the decomposition approach for solving the Support Vector Machine optimization problem (1)-(3) can provide significant effect. Note that the implementation of the algorithm is independent of the way of partition variables into decomposed blocks. Using of decomposition, compute the direction of descent the objective function (6) saves computing resources. However, an excessive reduction in the size of the decomposed block can lead to

additional computational time. Therefore, the size of the decomposed block is recommended in 5-10% of the training sets size.

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

References

[1] Konnov I V 2015 Sequantial threshold control in descent splitting methods for decomposable optimization problems *Optimization Methods and Software* **30(6)** 1238-1254
[2] Chang C C, Lin C J *LIBSVMdata: Classification, regression and multi-label* (http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/)
[3] Burges C J 1998 A Tutorial on Support Vector Machines for Pattern Recognition *Data Mining and Knowledge Discovery* **2(2)** 121–167