A Novel Model of Working Set Selection for SMO Decomposition Methods

Zhen-Dong Zhao† Lei Yuan‡ Yu-Xuan Wang‡ Forrest Sheng Bao‡ Shun-Yi Zhang† Yan-Fei Sun‡
1. Institution of Information & Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, 210003, CHINA
2. School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, 210003, CHINA
{zhaozhendong, lyuan0388, logpie}@gmail.com {dirzsy, sunyanfei}@njupt.edu.cn

Abstract—In the process of training Support Vector Machines (SVMs) by decomposition methods, working set selection is an important technique, and some exciting schemes were employed into this field. To improve working set selection, we propose a new model for working set selection in sequential minimal optimization (SMO) decomposition methods. In this model, it selects B as working set without reselection. Some properties are given by simple proof, and experiments demonstrate that the proposed method is in general faster than existing methods.

Index Terms—support vector machines, decomposition methods, sequential minimal optimization, working set selection

I. INTRODUCTION

In the past few years, there has been huge of interest in Support Vector Machines (SVMs) [1], [2] because they have excellent generalization performance on a wide range of problems. The key work in training SVMs is to solve the following quadratic optimization problem.

$$\min_{\alpha \in B} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

subject to

$$0 \leq \alpha_i \leq C, i = 1, \ldots, l$$

$$y^T \alpha = 0$$

where e is the vector of all ones, C is the upper bound of all variables, and $$Q_{ij} = y_i y_j K(x_i, x_j), K(x_i, x_j)$$ is the kernel function.

Notable effects have been taken into training SVMs [3], [4], [5], [6]. Unlike most optimization methods which update the whole vector $$\alpha$$ in each iteration, the decomposition method modifies only a subset of $$\alpha$$ per iteration. In each iteration, the variable indices are split into a "working set": $$B \subseteq \{1, \ldots, l\}$$ and its complement $$N = \{1, \ldots, l\} \setminus B$$. Then, the sub-problem with variables $$x_i, i \in B$$, is solved, thereby, leaving the values of the remaining variables $$x_j, j \in N$$ unchanged. This method leads to a small sub-problem to be minimized in each iteration. An extreme case is the Sequential Minimal Optimization (SMO) [5], [7], which restricts working set to have only two elements. Comparative tests against other algorithms, done by Platt [5], indicates that SMO is often much faster and has better scaling properties.

Since only few components are updated per iteration, for difficult problems, the decomposition method suffers from slow convergence. Better method of working set selection can reduce the number of iterations and hence is an important research issue. Some methods were proposed to solve this problem and to reduce the time of training SVMs [8]. In this paper, we propose a new model to select the working set. In this model, specially, it selects B without reselection. In other word, once $$\{\alpha_i, \alpha_j\} \subseteq B$$ are selected, they will not be tested or selected during the following working set selection. Experiments demonstrate that the new model is in general faster than existing methods.

This paper is organized as following. In section II, we give literature review, SMO decomposition method and existing working set selection are both discussed. A new method of working set selection is then presented in section III. In section IV, experiments with corresponding analysis are given. Finally, section V concludes this paper.

II. LITERATURE REVIEW

In this section we discuss SMO and existing working set selections.

A. Sequential Minimal Optimization

Sequential Minimal Optimization (SMO) was proposed by Platt [5], which is an extreme case of the decomposition algorithm where the size of working set is restricted to two. This method is named as Algorithm 1. Keerthi improved the performance of this algorithm for training SVMs (classifications and regressions [9], [7]). To take into account the situation that the Kernel matrices are Non-Positive Definite, Pai-Hsuen Chen et al. [10], [8] introduce the Algorithm 2 by restrict proof.

Algorithm 2

Step 1: Find $$\alpha^k$$ as the initial feasible solution. Set k=1.

Step 2: If $$\alpha^k$$ is a stationary point of (1), stop. Otherwise, find a working set $$B \equiv \{i, j\}$$.

Step 3: Let $$a_{ij} = K_{ii} + K_{jj} - K_{ij}$$. If $$a_{ij} > 0$$, solve the sub-problem. Otherwise, solve:
\[
\min_B \frac{1}{2} \begin{bmatrix} a_i & a_j \end{bmatrix} \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ji} & Q_{jj} \end{bmatrix} \begin{bmatrix} a_i \\ a_j \end{bmatrix} + (-\epsilon_B + Q_{BN} \alpha_N^k)^T \begin{bmatrix} a_i \\ a_j \end{bmatrix} + \frac{\tau - \alpha_{ij}((\alpha_i - \alpha_k^i)^2 + (\alpha_j - \alpha_k^j)^2)}{4}
\]
subject to \(0 \leq \alpha_i \leq C\),
\[y_i\alpha_i + y_j\alpha_j = -y_i^N\alpha_k^k\]

where \(\tau\) is a small positive number.

**Step 4:** Set \(\alpha_N^{k+1} = \alpha_N^k\). Set \(k \leftarrow k + 1\) and goto step 2.

Working set selection is a very important procedure during training SVMs. We discuss some existing methods in next subsection.

**B. Existing Working Set Selections**

Currently a popular way to select the working set \(B\) is via the "maximal violating pair", we call it WSS 1 for short. This working set selection was first proposed in [6], and is used in, for example, the software LIBSVM [3]. Instead of using first order approximation which WSS 1 has used, a new method which consider the more accurate second order information has been used, a new method recently proposed in [8], propose a new working set selection where a certain \(K\) is only positive semi-definite, so it is possible that \(K_{ii} + K_{jj} - 2K_{ij} = 0\). Moreover, some existing kernel functions (e.g., sigmoid kernel) are not the inner product of two vectors, so \(K\) is even not positive semi-definite. Then \(K_{ii} + K_{jj} - 2K_{ij} < 0\) may occur .

For this reason, Chen et al. [8], propose a new working set selection named WSS 3.

**WSS 3**

**Step 1:** Select
\[\alpha_{ts} \equiv \begin{cases} a_{ts} & \text{if } a_{ts} > 0 \\ \tau & \text{otherwise} \end{cases}\]

\[i \in \arg \max_j \{-y_i \nabla f(\alpha^k)_i | t \in I_{up}(\alpha^k)\},\]  \(3\)

**Step 2:**
Consider Sub(B) defined and select
\[j \in \arg \min_t \left\{-\frac{\partial^2}{\partial a_{ij}^2} | t \in I_{low}(\alpha^k), -y_i \nabla f(\alpha^k)_t < -y_j \nabla f(\alpha^k)_t \right\},\]  \(4\)

**Step 3:** Return \(B \equiv \{i, j\}\) where
\[a_{ij} \equiv K_{ii} + K_{jj} - 2K_{ij} > 0\]
\[b_{ij} \equiv -y_i \nabla f(\alpha^k)_i + y_j \nabla f(\alpha^k)_j > 0\]

**III. OUR NEW METHOD**

**A. Interesting Phenomena**

Interestingly, when we test the datasets by LIBSVM [3] which employs Algorithm 2 and WSS 3, some phenomena attract us. Fig. 1 illustrates two of them.

The first phenomenon is, lots of \(\alpha\) have not been selected at all in the training process. Because of the using of "Shrink" [4], some samples will be "shrunk" during the training procedure, thus, they will never be selected and optimized. At the same time, we notice another interesting phenomenon that several \(\alpha\) are selected to optimize the problem again and again, while others remain untouched. But does this kind of reselection necessary?

**B. Working Set Selection Without Reselection(WSS-WR)**

After investigating the phenomena above, if we limit the reselection of \(\alpha\), we could effectively reduce the time for training SVMs, with an acceptable effect on the ability of generalization. Thus, we propose our new method of working set selection where a certain \(\alpha\) can only be selected once.

Before introducing the novel working set selection, we give some definitions:

**Definition 1:** \(T^{k+1}, k \in \{1, \cdots, \frac{I}{2}\}\) is denoted as optimized set, in which \(\forall \alpha \in T\) has been selected and optimized once in working set selection.

**Definition 2:** \(C^{k+1} \subset \{1, \cdots, I\}\) \(T^k\) is called available set, in which \(\forall \alpha \in C\) has never been selected.

For optimization problem (1), \(\alpha \equiv C \cup T \cup B\).

Our method can be described as following: In iteration \(k\), a set \(B^k \equiv \{\alpha_k^i, \alpha_k^j\}\) will be selected from the new method which considers the more accurate second order information available set \(C^k\) by WSS 3, after optimization, both sample in this set will be put into \(T^{k+1}\), that is to say, \(T^{k+1} = T^k \cup B^k, C^{k+1} = C^k \setminus B^k\). In other words, once a working set is selected and optimized, all samples in it will not be examined anymore in the following selection. The relationship between sets \(B, C, T\) are shown in the Fig. 2
C. Some Properties of WSS-WR Model

We name this new method as Working Set Selection Without Reselection (WSS-WR), in which all \( \alpha \) will not be reselected.

1) Working Set Selection Without Reselection (WSS-WR):
Step 1: Select
\[
\frac{a_{ts}}{a_{ts}} \equiv \begin{cases} 
\frac{a_{ts}}{\tau} & \text{if } a_{ts} > 0 \\
\tau & \text{otherwise}
\end{cases}
\]
\[i \in \arg \max_{t} \{-y_{i} \nabla f(a_{k})|t \in I_{up}(a_{k}) \cap t \notin T^{k}\}, \quad (5)\]
\[j \in \arg \min_{t} \{ \frac{k_{i}^{2}}{a_{i}}t \in I_{low}(a_{k}) \cap t \notin T^{k}, \]
\[\frac{y_{i} \nabla f(a_{k})}{a_{i}} < \frac{y_{j} \nabla f(a_{k})}{a_{j}}\}. \quad (6)\]
Step 2: \( T^{k+1} = T^{k} \cup B^{k}, C^{k+1} = C^{k} \setminus B^{k} \)
Step 3: Return \( B = \{i, j\} \)
where
\[a_{ij} \equiv K_{i} + K_{ij} - 2K_{ij} > 0\]
\[b_{ij} \equiv -y_{i} \nabla f(a_{k})_{i} + y_{j} \nabla f(a_{k})_{j} > 0\]

We name this new method as Working Set Selection Without Reselection (WSS-WR), in which all \( \alpha \) will not be reselected.

Proof: According to the Theorem 1, \( \alpha \equiv 0 \) which is chosen by WSS-WR. And Keerthi [9] define the \( I_{up}, I_{low} \) as:
\[I_{0} \equiv \{i : 0 < \alpha_{i} < C\}\]
\[I_{1} \equiv \{i : y_{i} = +1, \alpha_{i} = 0\}\]
\[I_{2} \equiv \{i : y_{i} = -1, \alpha_{i} = C\}\]
\[I_{3} \equiv \{i : y_{i} = +1, \alpha_{i} = C\}\]
\[I_{4} \equiv \{i : y_{i} = -1, \alpha_{i} = 0\}\]
Thus \( I_{up} \equiv I_{1} \text{ and } I_{low} \equiv I_{4} \text{ in WSS-WR model.} \]

Lemma 1: In the Model of WSS-WR, \( a_{1}^{\text{new}} = a_{2}^{\text{new}} \)
Proof: During the SMO algorithm searches through the feasible region of the dual problem and minimizes the objective function (1). Because \( \sum_{i=1}^{N} y_{i} \alpha_{i} = 0 \), we have
\[y_{1} \alpha_{1}^{\text{new}} + y_{2} \alpha_{2}^{\text{new}} = y_{1} \alpha_{1}^{\text{old}} + y_{2} \alpha_{2}^{\text{old}}\]
Since
\[\alpha_{1}^{\text{old}} \in I_{up}, \alpha_{2}^{\text{old}} \in I_{low}\]
Therefore
\[y_{1} \alpha_{1}^{\text{new}} + y_{2} \alpha_{2}^{\text{new}} = 0\]
\[\Rightarrow y_{1} \alpha_{1}^{\text{new}} = -y_{2} \alpha_{2}^{\text{new}}\]

According to the Lemma 1
\[y_{1} = -y_{2}\]
So \( a_{1}^{\text{new}} = a_{2}^{\text{new}}\)

IV. Computational Comparison

In this section we compare the performance of our model against WSS 3. The comparison between WSS 1 and WSS 3 have been done by Rong-En Fan et al. [8].

A. Data and Experimental Settings

Some small datasets (around 1,000 samples) including nine binary classification and two regression problems are investigated under various settings. And the large (more than 30,000 instances) classification problems are also taken into account. We select splice from the Delve archive (http://www.cs.toronto.edu/~delve). Problems german,numer,heart and australian are from the Statlog collection [11]. Problems fourclass is from [12] and be transformed to a two-class set. The datasets diabetes, breast-cancer, and mpg are from the UCI machine learning repository [13]. The dataset mg from [14]. Problems a1a and a9a are compiled in Platt [5] from the UCI "adult" dataset. Problems w1a and w8a are also from Platt [5]. The problem IJCNN1 is from the first problem of IJCNN 2001 challenge [15].

For most datasets, each attribute is linearly scaled to \([-1, 1]\) except a1a, a9a, w1a, and w8a, because they take two values, 0 and 1. All data are available at http://www.csie.ntu.edu.tw/
problems type

| Kernel | Problems Type | logγ | log2^C |
|---|---|---|---|
| RBF | Classification | -5,15,2 | 3,15,2 |
| | Regression | -1,15,2 | 3,15,2 |
| Linear | Classification | -3,5,2 | -3,5,2 |
| Polynomial | Classification | -3,5,2 | -3,5,2 |
| | Regression | -3,5,2 | -3,5,2 |
| | Regression | -3,5,2 | -3,5,2 |
| Sigmoid | Classification | -3,12,3 | -12,3,3 |
| | Regression | γ = 1 | -8,1,3 |

Table I
Parameters used for various kernels: values of each parameter are from a uniform discretization of an interval. We list the left, right end points and the space for discretization. For example: -5,15,2. For log 2^C means log 2^C = -5, -3, ..., 15

We test various situations concerning all the commonly used kernels, with and without shrinking technique and 100MB/100KB cache. The Table II-V show that cross validation accuracy between WSS-WR and WSS 3 method are almost the same. To be specifically, |Accuracy_WSS-WR − Accuracy_WSS 3| < 0.026. And in most of the datasets, the accuracy of WSS 3 outperform that of WSS-WR. There are also several datasets in which WSS-WR performs even more accurate than WSS 3. Besides, great improvement are made both in the number of iterations as well as the consumption of time.

B. Numerical Experiments

1) Comparison of Cross Validation Accuracy of Classification: First, the grid method is applied. Cross validation accuracy is compared during "parameters selection" and "final training".
illustrate the iteration and time ratios between WSS-WR and WSS 3.

For each kernel, we give two figures showing results of "parameter selection" and "final training" steps, respectively. We further separate each figure to two situations: without/with shrinking, and present five ratios between using WSS-WR and using WSS 3:

\[
\begin{align*}
\text{ratio}1 & \equiv \frac{\text{time of WSS-WR (100M cache,shrinking)}}{\text{time of WSS 3 (100M cache,shrinking)}} \\
\text{ratio}2 & \equiv \frac{\text{time of WSS-WR (100M cache,nonshrinking)}}{\text{time of WSS 3 (100M cache,nonshrinking)}} \\
\text{ratio}3 & \equiv \frac{\text{time of WSS-WR (100K cache,shrinking)}}{\text{time of WSS 3 (100K cache,shrinking)}} \\
\text{ratio}4 & \equiv \frac{\text{time of WSS-WR (100K cache,nonshrinking)}}{\text{time of WSS 3 (100K cache,nonshrinking)}} \\
\text{ratio}5 & \equiv \frac{\text{Total iteration of WSS-WR}}{\text{Total iteration of WSS 3}}
\end{align*}
\]

The number of iterations is independent of the cache size. In the "parameter selection" step, time (or iterations) of all parameters is summed up before calculating the ratio. In general the "final training" step is very fast, so the timing result may not be accurate. Hence we repeat this step 4 times to obtain more reliable timing values. Fig. 3-10 present obtain ratios, They are in general less than 1. We can conclude that using WSS-WR is in general better than using WSS 3.
4) Caching and Shrinking techniques in WSS-WR: According to the analysis of the experimental results, in WSS-WR model, the caching and shrinking techniques do not have much more effects on SMO decomposition method. The data in Table VII, VIII certify the above conclusion.

5) Experiments of Large Classification Datasets: Next, the experiment with large classification sets is handled by a similar procedure. As the parameter selection is time consuming, we adjust the "parameter selection" procedure to a 16-point searching. The cache size are 300MB and 1MB. The experiments employ RBF and Sigmoid kernel methods, along with that Sigmoid kernel in general leads the worst ratio between

![Fig. 7. Iteration and time ratios between WSS-WR and WSS 3 using the Polynomial kernel for the "parameter selection" step (top: with shrinking, bottom: without shrinking).](image1)

![Fig. 8. Iteration and time ratios between WSS-WR and WSS 3 using the Polynomial kernel for the "final training" step (top: with shrinking, bottom: without shrinking).](image2)

![Fig. 9. Iteration and time ratios between WSS-WR and WSS 3 using the Sigmoid kernel for the "parameter selection" step (top: with shrinking, bottom: without shrinking).](image3)
WSS-WR and WSS 3. Table IX gives iteration and time ratios. Comparing the results among small problems, we safely draw the conclusion that ratios of time and iteration in large datasets are less than those of them in small ones, especially, in the situation of small size of cache.

C. Convergence Graph of WSS-WR

For the reason that WSS-WR employs Algorithm 2 and second order information method, which WSS 3 used, we just compare the convergence rates between them.

We set $C = 1, \gamma = 0, \epsilon = 10^{-3}$ during the whole comparison.

First, the evaluation is made on the following datasets: a1a, w1a, australian, splice, breast-cancer, diabetes, fourclass, german, numer and heart.

The illustrations in Fig. 11 are the first step evaluation, we compare the convergence ratios on nine datasets between WSS-WR and WSS 3 with RBF kernel. Other charts are omitted here for short.

For further analysis, we choose two datasets: w1a and breast-cancer to make evaluations by using diversity kernel methods (Linear, RBF, Polynomial, Sigmoid). Fig. 12 compare the convergence on W1a between WSS-WR and WSS 3. The illustrations of breast-cancer are omitted here for short.

From the Fig. 11, 12. The convergence rates of WSS-WR and WSS 3 are exactly the same at the beginning of the procedure. In addition, WSS-WR will be terminated soon when it reaches the optimum, while WSS 3, on the contrary, will hold the objective value or make insignificant progress with much more time consumed. Thus, it is reasonable to conclude that WSS-WR is much more efficient.

D. Discussion

With the above analysis, our main observations and conclusions from Fig. 3-10 and Table VII, VIII, IX are in the following:

1) The improvement of reducing the number of iterations is significant, which can be obtained from the illustrations.

2) Using WSS-WR dramatically reduces the cost of time. The reduction is more dramatic for the parameter selection step, where some points have low convergence rates.

3) WSS-WR outperforms WSS 3 in most datasets, both in the "parameter selection" step and in "final training" steps. Unlike WSS 3, the training time of WSS-WR does not increase when the amount of memory for caching drops. This property indicates that WSS-WR is useful under such situation where the datasets is too large for the kernel matrices to be stored, or where there is not enough memory.

4) The shrinking technique of LIBSVM [3] was introduced by Pai-Hsuen Chen et al., [10] to make the decomposition method faster. But in the view of Table VII, VIII, experiments of regression and large classification problems, shrinking technique almost does not shorten the training time by using WSS-WR.

5) Fig. 3-10 indicate that the relationship of the five ratios
can be described as follows:
\[
\text{ratio}5 < \text{ratio}4 < \text{ratio}3 < \text{ratio}2 < \text{ratio}1
\]
Though this may not be valid in all datasets.

6) WSS-\text{WR} still has shortcomings when compared to the WSS 3. According to the table III-VI, WSS-\text{WR} achieves slightly lower Cross Validation Accuracy; and some situations may be sensitive to this.

V. CONCLUSIONS

By analyzing the available working set selection methods and some interesting phenomena, we have proposed a new working set selection model—Working Set Selection Without Reselection (WSS-\text{WR}). Subsequently, full-scale experiments were given to demonstrate that WSS-\text{WR} outperforms WSS 3 in almost datasets during the "parameters selection" and "final training" step. Then, we discussed some features of our new model, by analyzing the results of experiments. A theoretical study on the convergence of WSS-\text{WR} and to continually improve this model are our future work.

VI. ACKNOWLEDGEMENTS

This work is under the support of the National High Technology Research and Development Program of China (863 Program) under Grant NO.2006AA01Z232, Natural Science Foundation of Jiangsu Provincial (BK2007603). Thanks to Chih-Chung Chang and Chih-Jen Lin for their powerful software, LIBSVM.

REFERENCES

[1] B. Boser, I. Guyon, and V. Vapnik, “A training algorithm for optimal margin classifiers,” In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 1992.
[2] C. Cortes and V. Vapnik, “Support-vector network,” Machine Learning, 20 1995.
[3] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, 2001, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
[4] T. Joachims, “Making large-scale support vector machine learning practical,” in Advances in Kernel Methods: Support Vector Machines, A. S. B. Schölkopf, C. Burges, Ed. MIT Press, Cambridge, MA, 1998.
[5] J. C. Platt, “Fast training of support vector machines using sequential minimal optimization,” in Advances in Kernel Methods - Support Vector Learning, C. J. C. B. Bernhard Schölkopf and A. J. Smola, Eds. MIT Press, Cambridge, MA, 1998.
[6] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection,” In Proceedings of CVPR'97, 1997.
[7] J.C.Platt, “Using sparseness and analytic qcp to speed training of support vector machines,” in Advances in Neural Information Processing Systems 11, S. S. M.S.Kearns and d. D.A.Cohn, Eds. MIT Press, Cambridge, MA, 1999.
[8] R.-E. Fan, P.-H. Chen, and C.-J. Lin, “Working set selection using second order information for training support vector machines,” Journal of Machine Learning Research, 6 2005.
[9] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, “Improvements to platt’s smo algorithm for svm classifier design,” Neural Computation, 13 2001.
[10] P.-H. Chen, R.-E. Fan, and C.-J. Lin, “A study on smo-type decomposition methods for support vector machines,” IEEE Transactions on Neural Networks, 2006, to appear. http://www.csie.ntu.edu.tw/ cjlin/papers/generaSMO.pdf.
[11] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, “Machine learning, neural and statistical classification,” Prentice Hall, 1994, data available at http://www.ncc.up.pt/iliac/MLstatlog/datasets.html.

[12] T. K. Ho and E. M. Kleinberg, “Building projectable classifiers of arbitrary complexity,” In Proceedings of the 13th International Conference on Pattern Recognition, August 1996.
[13] C. L. Blake and C. J. Merz, “Uci repository of machine learning databases,” Tech. Rep., 1998, available at http://www.ics.uci.edu/ mllearn/MLRepository.html.
[14] G. W. Flake and S. Lawrence, “Efficient svm regression training with smo,” Machine Learning, 46 2002.
[15] D. Prokhvorov, “Ijcnn 2001 neural network competition,” Tech. Rep., 2001, available. http://www.geocities.com/ijcnn/nnet_ijcnn01.pdf.