

# Expediting Model Selection for Support Vector Machines Based on Data Reduction\*

Yu-Yen Ou, Chien-Yu Chen, Shien-Ching Hwang and Yen-Jen Oyang

Department of Computer Science and Information Engineering

National Taiwan University, Taipei, Taiwan

{yien, yjoyang}@csie.ntu.edu.tw; {cychen, schwang}@mars.csie.ntu.edu.tw

**Abstract** – *In recent years, Support Vector Machines (SVM) have been extensively applied to deal with various data classification problems. However, in some cases, the application of SVM is limited due to the time taken to conduct model selection for SVM. This issue is of particular significant for some modern applications, such as web mining, in which the large-scale database is frequently updated. This paper proposes a data reduction based mechanism aimed at expediting the model selection process in SVM. Experimental results show that the proposed mechanism is able to greatly reduce the time taken to carry out model selection at minimum cost.*

**Keywords:** Support Vector Machines, Data Classification, Model Selection.

## 1 Introduction

The support vector machine was first proposed by Vapnik [5] and has since attracted a high degree of interest in the machine learning research community. Several recent studies have reported that the SVM (support vector machines) generally are capable of delivering higher performance in terms of classification accuracy than the other data classification algorithms. However, for some datasets, the performance of SVM is very sensitive to how the cost parameter and kernel parameters are set. As a result, the user normally needs to conduct extensive cross validation in order to figure out the optimal parameter setting. This process is commonly referred to as model selection.

One practical issue with model selection is that this process is very time-consuming. For example, if the model selection procedure adopted in [3] and [9] is employed to construct a SVM for the shuttle dataset in the Statlog collection [12], the complete grid-search model selection process will take over 60 hours on a machine equipped with dual pentium-III-1GHz CPUs and 1GB RAM. As the shuttle dataset, containing 43500 training instances, is not considered as a large case in the contemporary environment, how to speed up the model selection process for SVM becomes a crucial issue and

several studies have been conducted to address this issue in recent years [2, 10, 8, 7]. These studies share a common ground aimed at reducing the search space of parameter combinations.

In this paper, a data reduction based approach aimed at expediting the model selection process of SVM is proposed. The main idea of the proposed approach is to employ a data reduction mechanism to remove the non-essential training instances and thus to reduce the size of the training dataset. Experimental results reveal that the data reduction based approach is able to reduce the execution time of model selection process to 4.7%, comparing to the original grid-search method. Furthermore, the experimental results reveal that the classification accuracy of SVM is not traded by employing the proposed approach. In other words, the SVM with the parameter setting determined by the proposed approach is able to achieve the same level of classification accuracy as the SVM with the parameter setting determined by the traditional grid-search model selection process.

As far as the execution time of the proposed approach is concerned, the average time complexity of the data reduction process is  $O(n \log n)$ , where  $n$  is the number of instances in the training dataset.

This paper is organized as follows. In next section, we introduce some related background including some basic concepts of SVM, kernel function selection, and model selection (parameters selection) of SVM. In Section 3, we detail the proposed approach which is applied on SVM model selection process. Next, numerical experiments are in Section 4. Finally, we have some discussions and conclusions in Section 5.

## 2 Overview of SVM

In this section we introduce some basic concepts of SVM, kernel function selection, and model selection (parameters selection) of SVM.

### 2.1 Support Vector Machines

First, we briefly describe some concepts of SVM. Given training data  $x_i, i = 1, \dots, n$ , in two classes, and

a label vector  $y_i$  such that  $y_i \in \{1, -1\}$ , the standard SVM formulation [6] is as follows:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C \sum_{i=1}^n \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, n. \end{aligned} \quad (1)$$

If  $\phi(x_i) = x_i$ , we often call (1) as the linear kernel SVM, which is mainly to solve the linearly separable problem. Unfortunately, many applications in the real world are not linearly separable problems. Accordingly, we use  $\phi$  to map  $x_i$  into a higher dimensional space, and then call (1) a non-linear SVM.

For a non-linear SVM, after mapping by  $\phi$ , the number of variables of  $w$  can be very large or even infinite, so that it is very difficult to solve this problem from (1). As a result, people often solve the problem from the following dual formulation:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\alpha^T Q \alpha - e^T \alpha \\ \text{subject to} \quad & y^T \alpha = 0, \\ & 0 \leq \alpha_i \leq C, i = 1, \dots, n, \end{aligned} \quad (2)$$

where  $Q$  is an  $n \times n$  positive semi-definite matrix with  $Q_{ij} = y_i y_j \phi(x_i)^T \phi(x_j)$ , and  $e$  is the vector with all 1 elements. Usually we call  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$  the kernel function. Some popular kernel functions are, for example,  $e^{-\gamma \|x_i - x_j\|^2}$  (RBF),  $(x_i^T x_j / \gamma + \delta)^d$  (polynomial),  $\tanh(ax^T y + b)$  (hyperbolic tangent) etc., where  $\gamma$ ,  $d$ , and  $\delta$  are kernel parameters. In addition, (2) is easier to be solved than (1) because the number of variables in (2) is the size of the training dataset,  $n$ , not the dimensionality of  $\phi(x)$ .

It can be shown that if  $\alpha$  is an optimal solution of (2), then

$$w = \sum_{i=1}^n \alpha_i y_i \phi(x_i)$$

is the optimal solution of the primal (1). Then a decision function is written as

$$\text{sgn}(w^T \phi(x) + b) = \text{sgn}\left(\sum_{i=1}^n \alpha_i y_i K(x_i, x) + b\right).$$

That is, for a test vector  $x$ , if  $\sum_{i=1}^n y_i \alpha_i (\phi(x_i)^T \phi(x)) + b > 0$ , we classify it to be in the class 1. Otherwise, we think it is in the second class. Moreover, after (2) is solved with a solution  $\alpha$ , the vectors for which  $\alpha_i > 0$  are called *support vectors*. We can see that only support vectors will affect results in the prediction stage.

Table 1: The execution times of different phases of the SVM software in seconds

	model selection	training	testing
satimage	29052.1	31.85	11.75
letter	192358.6	141.66	85.65
shuttle	252018.5	112.51	2.13

## 2.2 Kernel Selection of SVM

There are many kernel functions in SVM, so how to select a good kernel function is also a research issue. However, for general purposes, there are some popular kernel functions: linear kernel, RBF kernel, polynomial kernel and hyperbolic tangent kernel. In these popular kernel functions, we often choose the RBF kernel function because of following reasons: 1. Some problems are not linearly separable, so we don't choose the linear kernel function. 2. We don't choose polynomial kernel function due to some numerical difficulties such as  $(< 1)^d \rightarrow 0$ , and  $(> 1)^d \rightarrow \infty$ . 3. Hyperbolic tangent kernel is not well studied now, but it seems to behave like RBF kernel for certain parameters.

## 2.3 Model Selection of SVM

Model selection is also an important issue in SVM. Recently, SVM have shown good performance in data classification. Its success depends on the tuning of several parameters which affect the generalization error. We often call this parameter tuning procedure as the model selection. If you use the linear SVM, you only need to tune the cost parameter  $C$ . Unfortunately, linear SVM are often applied to linearly separable problems. Many problems are non-linearly separable. For example, the problem shown in Figure 1 is obviously not linearly separable. Therefore, we often apply non-linear kernel to solve classification problems, so we need to select the cost parameter and kernel parameters. As we discussed in the previous subsection, we often choose the RBF kernel function in general applications. In the RBF kernel function  $K(x_i, x_j) \equiv e^{-\gamma \|x_i - x_j\|^2}$ , we need to select the cost parameter  $C$  and kernel parameter  $\gamma$ .

We usually use the grid-search method in cross validation to select the best parameter set. That is, to do the cross validation in training dataset by trying different parameter combinations (often  $15 \times 15 = 225$  combinations) to get the best one. Then apply this parameter set to the training dataset and then get the classifier. After that, use the classifier to classify the testing dataset to get the generalization accuracy.

The grid-search model selection method is very time-consuming as shown in Table 1. Several researchers have worked on this issue. For example, Chung et al. [4] use the radius margin bound [2, 10, 8] to speed up the search process, and Keerthi and Lin [11] propose the two-line

**Algorithm 1:** The Data Reduction Algorithm

**Input:** Training dataset  $T$

**Output:** Reduced dataset  $R$

//  $NED(x_i)$  : the distance of  $x_i$ 's nearest enemy

//  $N_k(x_i)$  : the set of  $x_i$ 's  $k$  nearest neighbors

//  $C(x_i)$  : the class label of  $x_i$

**Data Reduction**( $T$ )

- (1)  $R \leftarrow T$
- (2) **ForEach** instance  $x_i$  in  $R$
- (3) Find  $NED(x_i)$
- (4) **Sort**  $R$  in descending order by  $NED(x)$
- (5) **ForEach** instance  $x_i$  in  $R$
- (6) Find  $N_k(x_i)$  in  $R$
- (7) **If**  $C(x_i) \neq C(p), \forall p \in N_k(x_i)$
- (8) Remove  $x_i$  from  $R$
- (9) **return**  $R$

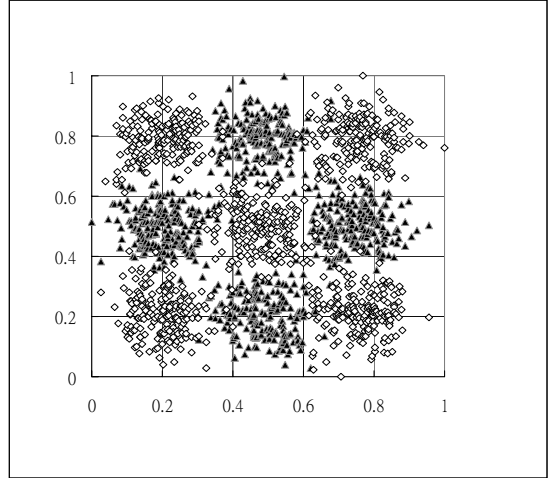


Figure 1: Original Training Dataset

search method which uses some searching heuristic to speed up the model selection process. In this paper, we try to solve this problem from another point of view.

### 3 Method

The task carried out by the learning algorithm of SVM is to figure out the boundary that separates the two different classes of training data, subject to certain optimization criteria. In the SVM algorithm, the exact profile of the boundary is determined by the training instances located in the proximity of the boundary. Therefore, those training instances that are far away from the boundary essentially play no role in determining the boundary. This observation forms the basis of the mechanism proposed in this paper for expediting the model selection process in the SVM algorithm.

The kernel of the proposed mechanism is a data reduction process employed to remove the training instances that are away from the boundary of different classes of training data. Figure 1 and Figure 2 illustrate the effect of the proposed data reduction mechanism. With the reduced training dataset, the conventional grid-search is then conducted to figure out the optimal parameter set for the given data classification problem.

The data reduction mechanism employed in this paper operates by first sorting the training dataset in the descending order according to the distance of each instance's *nearest enemy* [13], which is defined to be the nearest neighbor of the instance belonging to a different class. That is, instances will be examined for removal beginning at the instance furthest from its nearest enemy. This tends to remove instances furthest from the decision boundary first, which in turn increases the chance of retaining border points. With the sorted

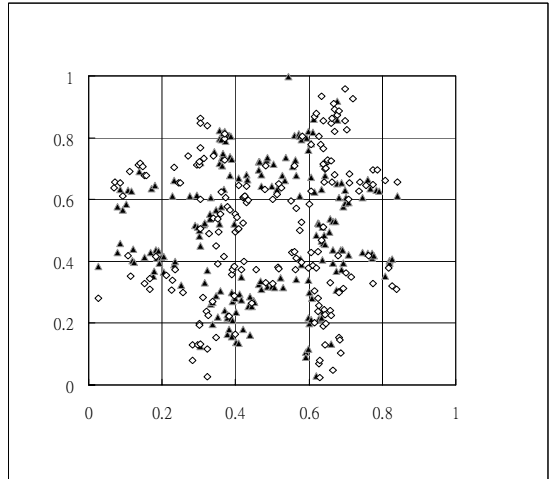


Figure 2: Reduced Training Dataset

list, the data reduction algorithm then examines the instances in the training dataset one by one. For each training instance, if the instance and all of its  $k$  nearest neighbors in the remaining training dataset belong to the same class, then the instance is considered as redundant and is removed from the training dataset.

Algorithm 1 shows the pseudo-code of the data reduction algorithm employed in this paper. There is a parameter to be set in our algorithm, parameter  $k$  in the pseudo-code. We tried some  $k$  values from 3 to 10, and then we choose  $k = 3$  in the experiments reported in the next section. Generally speaking, if we select a larger  $k$ , the reduction rate will lower (fewer training instances will be removed from the training dataset). On the other hand, if the  $k$  is smaller, the reduction rate will higher. By some experimental results, we conclude  $k = 3$  is a suitable value because it has a reasonable reduction rate.

As far as the time complexity of the proposed data reduction mechanism is concerned, for each training instance, we need to identify its  $k$  nearest neighbors and its nearest enemy. If the *kd-tree* structure is employed [1], then the average time complexity of this process is  $O(n \log n + k n \log n)$ , where  $n$  is the total number of training data. The term  $n \log n$  is due to construction of the *kd-tree*. With the *kd-tree* constructed, we then need to search the *kd-tree* to figure out the  $k$  nearest neighbors of each training instance. This task can be carried out in  $O(k n \log n)$ . In addition, for all training data, we only need to construct the *kd-tree* only one time, and if we need to remove a training instance, we can remove it from *kd-tree* in  $O(\log n)$ . It is unnecessary to reconstruct the *kd-tree* in each search for each training instance. As a result, the time complexity is  $O(n \log n + k n \log n)$ . If  $k$  is treated as a constant, then the average time complexity of the proposed mechanism is  $O(n \log n)$ .

## 4 Experiments

This section reports the experiments conducted to evaluate the effects of the proposed data reduction based mechanism for expediting the model selection process of SVM. In particular, we are interested in how the mechanism proposed in this paper performs in comparison with the two-line search method proposed by [11] and [3]. In these experiments, LIBSVM with RBF kernel is employed. Accordingly, there are two parameters, the RBF kernel parameter  $\gamma$  and the cost parameter  $C$ , to be set. Table 2 lists the main characteristics of the three datasets used in the experiments. All three datasets, *satimage*, *letter*, and *shuttle*, are from the Statlog collection [12]. In these experiments, 5-fold cross validation is conducted to determine which of the following 225 combinations of  $(\gamma, C)$  is the most appropriate for the given data classification problem with respect to prediction accuracy. The hardware platform used in the ex-

Table 2: Statistics of the datasets used in the experiments

	#training	#testing	#class	#att
<i>satimage</i>	4435	2000	6	36
<i>letter</i>	15000	5000	26	16
<i>shuttle</i>	43500	14500	7	9

Table 3: Comparison of three methods in execution time of model selection

	<i>satimage</i>	<i>letter</i>	<i>shuttle</i>	Avg
Proposed method	1791s 6.2%	14528s 7.6%	63s 0.2%	4.7%
two-line method	3244s 11.1%	15716s 8.2%	7887s 3.1%	7.5%
Original method	29052s 100%	192358s 100%	252018s 100%	100%

periments is a workstation with dual Pentium-III-1GHz CPUs, 2GB RAM, and the FreeBSD UNIX-release 4.7.

The following three tables compare the effects of the data reduction mechanism proposed in this paper and that of alternative approaches. Table 3 shows how the execution time of model selection compares and Table 4 shows how the classification accuracy compares. As Table 3 reveals, both the mechanism proposed in this paper and the two-line method can significantly reduce the execution time taken to carry out model selection process, which is the main concern of this paper. In fact, the numbers listed in Table 3 with the two-line method are extracted from [3] and are listed here for reference only. In [3], the authors used almost identical hardware platform as what is used in this paper.

Table 4 compares the classification accuracy delivered with alternative approaches. As Table 4 reveals, both the mechanism proposed in this paper and the two-line search method cause slight degradation of classification accuracy. However, the degradation caused by the mechanism proposed in this paper is smaller than that caused by the two-line method.

Table 5 presents further insight about why the speed-up of the model selection process with the proposed mechanism could vary significantly in different datasets. In Table 5, the number of training data remains in each dataset after data reduction has been applied is listed. *Shuttle* is a good example of those datasets that contains a very high percentage of redundant training data. For such a dataset, only a very small percentage of the training data will remain after data reduction is applied and the speed-up of the model selection process will be most significant. On the other hand, for a dataset that does not contain a very high percentage of redundant train-

Table 4: Comparison of three methods in generalization accuracy

	satimage	letter	shuttle	Avg
Proposed	91.2%	97.82%	99.92%	96.31%
Two-line	91.55%	96.54%	99.81%	95.97%
Original	91.8%	97.82%	99.92%	96.51%

Table 5: The numbers of training samples left after data reduction is applied.

		satimage	letter	shuttle	Avg
Proposed method	#obj %	1167 26.3%	4027 26.8%	272 0.61%	17.9%
Original datasets	#obj %	4435 100%	15000 100%	44500 100%	100%

ing instances, e.g. `satimage` and `letter`, the speed-up of the model selection process will be less significant.

The execution time of our proposed method are listed in Table 6. Comparing to the execution time of model selection, the execution time taken by proposed method is very few.

The overall observation is that both our proposed method and two-line search method can significantly reduce the execution time of model selection process. Our proposed method has the comparable or even better results than the two-line search method. Furthermore, as Table 4 reveals, classification accuracy of our proposed method is almost the same with the original grid-search method, but the generalization accuracy of two-line search method is slightly worse than our proposed method and the original method.

## 5 Discussion and Conclusion

In recent years, how to speed up the model selection process of SVM has emerged as a critical issue for extending the applications of SVM. This issue is getting more and more significant because the model selection process dominates the time taken to construct a SVM based classifier and many contemporary databases are not only large but also frequently updated. This paper presents a data reduction based mechanism to cope with this challenge. The basis of the proposed mechanism is very different from the other speed-up approaches that have been developed recently. Most of the existing approaches attempt to reduce the search space of model selection process, but our proposed mechanism is focus on reducing the number of training data. Experimental results show that the proposed mechanism is able to greatly reduce the time taken to carry out model selection at minimum cost.

Table 6: Execution time of proposed method

	satimage	letter	shuttle
Time	37.2s	260s	712.4s

As the mechanism proposed in this paper works well according to our experiments, it is of interest to investigate whether it will work as well in merging applications such as bioinformatics. The main issue here is whether the data reduction based mechanism can work well with some novel kernel functions developed for applying the SVM to bioinformatics. Another issue that deserves additional study is whether the data reduction based mechanism can be employed in conjunction with the other approaches, e.g. the two-line search method, to deliver even more effective results.

## Acknowledgment

The authors would like to thank Chih-Jen Lin for his useful software and many fruitful comments.

## References

- [1] J. Bentley. Multidimensional binary search trees used for associative searching. In *Communications of the ACM*, pages 18(9):509–517, 1975.
- [2] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46:131–159, 2002.
- [3] K.-M. Chung, W.-C. Kao, C.-L. Sun, and C.-J. Lin. Decomposition methods for linear support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2002.
- [4] K.-M. Chung, W.-C. Kao, C.-L. Sun, L.-L. Wang, and C.-J. Lin. Radius margin bounds for support vector machines with the rbf kernel. *Neural Computation*, 2003. To appear.
- [5] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.
- [6] C. Cortes and V. Vapnik. Support-vector network. *Machine Learning*, 20:273–297, 1995.
- [7] D. DeCoste and K. Wagstaff. Alpha seeding for support vector machines. In *Proceedings of International Conference on Knowledge Discovery and Data Mining (KDD-2000)*, 2000.
- [8] K. Duan, S. S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 2002. To appear.

- [9] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.
- [10] S. S. Keerthi. Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 2002. To appear.
- [11] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation*, 15:1667–1689, 2003.
- [12] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Prentice Hall, Englewood Cliffs, N.J., 1994. Data available at <ftp://ftp.ncc.up.pt/pub/statlog/>.
- [13] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.