EFFICIENT REDUCTION OF SUPPORT VECTORS IN KERNEL-BASED METHODS

Takumi Kobayashi and Nobuyuki Otsu

National Institute of Advanced Industrial Science and Technology
Tsukuba, 1-1-1 Umezono, Japan

ABSTRACT

Kernel-based methods, e.g., support vector machine (SVM), produce high classification performances. However, the computation becomes time-consuming as the number of the vectors supporting the classifier increases. In this paper, we propose a method for reducing the computational cost of classification by kernel-based methods while retaining the high performance. By using linear algebra of a kernel Gram matrix of the support vectors (SVs) at low computational cost, the method efficiently prunes the redundant SVs which are unnecessary for constructing the classifier. The pruning is based on the evaluation of the performance of the classifier formed by the reduced SVs in SVM. In the experiment of classification using SVM for various datasets, the feasibility of the evaluation criterion and the effectiveness of the proposed method are demonstrated.

***Index Terms*—** Reduction of support vectors, Kernel-based method, Support vector machine

1. INTRODUCTION

Kernel-based methods, such as KPCA [1] and KDA [2], extend existing linear methods to non-linear ones, often achieving the state-of-the-art performance. In particular, support vector machine (SVM) [3] with kernel has been successfully applied in versatile problems. These kernel-based methods, however, require rather large computational cost since we must calculate the kernel functions for all vectors, called support vectors (SVs)[1], supporting the classifier (or the projection). The number of SVs increases linearly in SVM for a large scale problem [4] and time-consuming, and still more in the case of the SVs correspond to *all* samples in the other kernel-based methods in general. Therefore, the classification becomes much more time-consuming, especially for large-scale problems. Practically speaking, it is desirable to reduce the computational cost required for the classification by kernel-based methods.

Some previous works have shown that a classifier constructed by a reduced set of SVs can still retain high performance. Burges et al. [5] have approximated the SVM classifier by using a smaller number of vectors in terms of Euclidean distance, but they stated that it is computationally expensive to find such a reduced set. Previous works closely related to our method are those of [6, 7] which focused on linear dependency of SVs. We can perform such a method for reducing the computational cost required in the classifier as post-processing, not pre-processing [8, 9]. In [8, 9], the candidates for SVs are assumed to be predefined, and it is a difficult and exhaustive task to select the candidates that increase the performance.

In this paper, we propose a method for efficiently pruning the SVs used in kernel-based methods to reduce computational cost of the classification. The method is based on linear algebra of a kernel Gram matrix of SVs and prunes the redundant SVs at low computational cost while keeping the high performance of the classifier. For the pruning in SVM, we utilize the criterion for evaluating the classifier constructed by the reduced set of SVs. The effectiveness of the proposed method and the feasibility of the criterion are exhibited in experiments using SVM on various datasets.

2. REDUCTION OF SUPPORT VECTORS

2.1. Redundant Support Vector

In kernel-based methods, we consider the feature space mapped via nonlinear function ϕ from input space. The classifier corresponds to a hyperplane, the normal vector of which is represented by a linear combination of SVs:

$$y = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b, \qquad \boldsymbol{w} = \sum_{i=1}^{n_s} \alpha_i \phi(\boldsymbol{s}_i), \tag{1}$$

where y is the output value of the classifier, $\boldsymbol{x}$ is the input vector, $\boldsymbol{w}$ and b are the normal vector and the bias of the hyperplane, respectively, n_s is the number of support vectors ($\boldsymbol{s}_i$), and α_i is the linear coefficient for mapped support vectors $\phi(\boldsymbol{s}_i)$. Suppose a mapped support vector is linearly dependent on the others:

$$\phi(\boldsymbol{s}_i) = \sum_{j \neq i} \beta_j \phi(\boldsymbol{s}_j). \tag{2}$$

[1]In this paper, we use the term of support vectors (SVs) as the samples that contribute to the classifier not only in SVM but also in the other kernel-based methods.

Algorithm 1 Naive Support Vector Reduction

```
Require: Kernel gram matrix of SVs $\boldsymbol{K} \in \boldsymbol{R}^{n_s \times n_s}$ ($k_{ij} = k(\boldsymbol{s}_i, \boldsymbol{s}_j), 1 \le i, j \le n_s$)
Require: Linear coefficients for SVs $\boldsymbol{\alpha} \in \boldsymbol{R}^{n_s}$
 1: $n \leftarrow n_s, \hat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}, \hat{\boldsymbol{K}} \leftarrow \boldsymbol{K}$
 2: repeat
 3:     for $i = 1$ to $n$ do
 4:         $\boldsymbol{\beta}_i \leftarrow (\hat{\boldsymbol{K}}_{(i)} + \lambda \boldsymbol{I})^{-1} \hat{\boldsymbol{k}}_{(i)}$
 5:         $\epsilon_i \leftarrow \hat{k}_{ii} - 2\boldsymbol{\beta}_i^T \hat{\boldsymbol{k}}_{(i)} + \boldsymbol{\beta}_i^T \hat{\boldsymbol{K}}_{(i)} \boldsymbol{\beta}_i$
 6:     end for
 7:     $i^* \leftarrow \arg\min_i \epsilon_i$        /* redundant SV */
 8:     $\hat{\boldsymbol{\alpha}} \leftarrow \hat{\boldsymbol{\alpha}}_{(i^*)} + \hat{\alpha}_{i^*} \boldsymbol{\beta}_{i^*}$
 9:     $\hat{\boldsymbol{K}} \leftarrow \hat{\boldsymbol{K}}_{(i^*)}$
10:     $n \leftarrow n - 1$
11: until Stopping criterion is satisfied (see Sec.2.3)
```

In this case, the hyperplane in Eq.(1) can be rewritten as

$$\boldsymbol{w} = \sum_{j \neq i} \alpha_j \phi(\boldsymbol{s}_j) + \alpha_i \sum_{j \neq i} \beta_j \phi(\boldsymbol{s}_j) = \sum_{j \neq i} (\alpha_j + \alpha_i \beta_j) \phi(\boldsymbol{s}_j). \tag{3}$$

Thus, in the feature space, the linearly dependent SVs are redundant for constructing the classifier (hyperplane). This means that a set of SVs can be reduced by eliminating such redundant ones while maintaining the hyperplane.

The problem is how to find the linearly dependent SVs defined in Eq.(2). Downs et al. [7] have employed row reduced echelon form which can identify the SVs *strictly* satisfying Eq.(2). From the statistical viewpoint, however, it is sufficient that the redundant SVs satisfy the equation *approximately*, not *strictly*. Therefore, we employ sample-wise regression form. The linear coefficient $\boldsymbol{\beta}$ in Eq.(2) is obtained by the following least square problem:

$$\begin{aligned} \boldsymbol{\beta} &= \arg\min_{\boldsymbol{\beta}} ||\phi(\boldsymbol{s}_i) - \sum_{j \neq i} \beta_j \phi(\boldsymbol{s}_j)||^2 + \lambda ||\boldsymbol{\beta}||^2 \\ &= (\boldsymbol{K}_{(i)} + \lambda \boldsymbol{I})^{-1} \boldsymbol{k}_{(i)}, \end{aligned} \tag{4}$$

where $\boldsymbol{K}$ is the kernel Gram matrix of SVs ($k_{ij} = \phi(\boldsymbol{s}_i)^T \phi(\boldsymbol{s}_j) = k(\boldsymbol{s}_i, \boldsymbol{s}_j)$), (i) denotes the index excluding i, and thus $\boldsymbol{K}_{(i)}$ is the sub-matrix of $\boldsymbol{K}$ excluding the i-th row and column, $\boldsymbol{k}_{(i)}$ is the i-th column vector of $\boldsymbol{K}$ excluding the i-the row. We introduce a regularization term with parameter λ, say $\lambda = 0.001$, for avoiding rank reduction of the Gram matrix. The squared approximation error of Eq.(2) is

$$\epsilon \equiv ||\phi(\boldsymbol{s}_i) - \sum_{j \neq i} \beta_j \phi(\boldsymbol{s}_j)||^2 = k_{ii} - 2\boldsymbol{\beta}^T \boldsymbol{k}_{(i)} + \boldsymbol{\beta}^T \boldsymbol{K}_{(i)} \boldsymbol{\beta}. \tag{5}$$

By finding the SV of the least approximation error ϵ, the redundant SV is sequentially identified and eliminated. The naive algorithm is described in Algorithm 1.

Algorithm 2 Efficient Support Vector Reduction

```
Require: Kernel gram matrix of SVs $\boldsymbol{K} \in \boldsymbol{R}^{n_s \times n_s}$ ($k_{ij} = k(\boldsymbol{s}_i, \boldsymbol{s}_j), 1 \le i, j \le n_s$)
Require: Linear coefficients for SVs $\boldsymbol{\alpha} \in \boldsymbol{R}^{n_s}$
1: $\boldsymbol{H} \leftarrow (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}, \hat{\boldsymbol{\alpha}} \leftarrow \boldsymbol{\alpha}$
2: repeat
3:     $i^* \leftarrow \arg\max_i h_{ii}$        /* redundant SV */
4:     $\hat{\boldsymbol{\alpha}} \leftarrow \hat{\boldsymbol{\alpha}}_{(i^*)} - \hat{\alpha}_{i^*} \frac{\boldsymbol{h}_{(i^*)}}{h_{i^* i^*}}$
5:     $\boldsymbol{H} \leftarrow \boldsymbol{H}_{(i^*)} - \frac{\boldsymbol{h}_{(i^*)} \boldsymbol{h}_{(i^*)}^T}{h_{i^* i^*}}$
6: until Stopping criterion is satisfied (see Sec.2.3)
```

2.2. Efficient Reduction Method

The algorithm described above is time-consuming since the least square problem of Eq.(4) has to be solved for every SVs (lines 3∼6 in Algorithm 1). In order to efficiently solve it, we employ elementary linear algebra as follows.

First, we consider the inverse matrix of the regularized Gram matrix $\boldsymbol{K} + \lambda \boldsymbol{I}$:

$$\begin{aligned} \boldsymbol{H} \equiv (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} &= \begin{pmatrix} \boldsymbol{K}_{(i)} + \lambda \boldsymbol{I} & \boldsymbol{k}_{(i)} \\ \boldsymbol{k}_{(i)}^T & k_{ii} + \lambda \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \check{\boldsymbol{K}}_{(i)}^{-1} + \frac{\check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)} \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1}}{\check{k}_{ii} - \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)}} & \frac{-\check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)}}{\check{k}_{ii} - \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)}} \\ \frac{-\boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1}}{\check{k}_{ii} - \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)}} & \frac{1}{\check{k}_{ii} - \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)}} \end{pmatrix} \end{aligned} \tag{6}$$

where $\check{\boldsymbol{K}}_{(i)} = \boldsymbol{K}_{(i)} + \lambda \boldsymbol{I}$, $\check{k}_{ii} = k_{ii} + \lambda$, and we focus on the i-th sample by permutating the order in $\boldsymbol{K}$. By comparing Eq.(6) with Eq.(4) and Eq.(5), the solution of the least square problem can be simply described as

$$\boldsymbol{\beta} = -\frac{\boldsymbol{h}_{(i)}}{h_{ii}}, \quad \epsilon + \lambda ||\boldsymbol{\beta}||^2 = \frac{1}{h_{ii}} - \lambda \equiv \check{\epsilon}, \tag{7}$$

where $\check{\epsilon} \approx \epsilon$ due to $\lambda \ll 1$. Thus, once we calculate the inverse matrix $\boldsymbol{H}$ of the Gram matrix $\boldsymbol{K}$, the least square problem for every SV can be solved at quite a low computational cost. This enables us to avoid time-consuming processing.

For sequentially eliminating SVs, it still takes a large computational load to calculate the inverse matrix ($O(n^3)$) at every step (lines 2∼10 in Algorithm 1). We can efficiently reduce the cost by focusing on the property of $\boldsymbol{H}$ again:

$$\boldsymbol{H}_{(i)} = \check{\boldsymbol{K}}_{(i)}^{-1} + \frac{\check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)} \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1}}{\check{k}_{ii} - \boldsymbol{k}_{(i)}^T \check{\boldsymbol{K}}_{(i)}^{-1} \boldsymbol{k}_{(i)}} = \check{\boldsymbol{K}}_{(i)}^{-1} + \frac{\boldsymbol{h}_{(i)} \boldsymbol{h}_{(i)}^T}{h_{ii}}$$

$$\therefore (\boldsymbol{K}_{(i)} + \lambda \boldsymbol{I})^{-1} = \check{\boldsymbol{K}}_{(i)}^{-1} = \boldsymbol{H}_{(i)} - \frac{\boldsymbol{h}_{(i)} \boldsymbol{h}_{(i)}^T}{h_{ii}}. \tag{8}$$

The inverse matrix of the reduced Gram matrix $(\boldsymbol{K}_{(i)} + \lambda \boldsymbol{I})^{-1}$, in which the i-th SV is eliminated, can be easily calculated (updated) from that of the original inverse matrix $\boldsymbol{H}$.

In total, the time-consuming task is only a single calculation of the inverse matrix of the Gram matrix as in Eq.(6). The identification of the redundant SVs and the calculation of the linear coefficients in Eq.(2) are subsequently performed at small extra computational cost. The proposed algorithm for efficiently reducing SVs is described in Algorithm 2.

2.3. Stopping Criterion

Since there is a trade-off between the number of SVs and the performance in general, a stopping criterion for sequential reduction is required. We focus on the increase of cost (loss) values for the training dataset. In SVM, Hinge losses are employed:

$$HingeLoss = \frac{1}{N}\sum_{i=1}^{N}\max(0, 1 - g_i y_i) \tag{9}$$

where N is the number of training samples, $g_i \in \{+1, -1\}$ is the true label and y_i is the output value of the classifier for the i-th sample. We define the criterion as the increase of the Hinge loss from the original classifier. The reduction of SVs is stopped when the criterion value exceeds a certain threshold τ. Note that a criterion might be practically determined as the recognition error calculated by using a validation dataset like cross-validation procedure.

3. EXPERIMENTAL RESULT

We conducted two types of experiments using benchmark datasets and the MNIST handwritten digit dataset. In the experiments, SVs are obtained by SVM with gaussian kernel, the parameters of which are determined by cross validation. The proposed method reduces those SVs.

[Benchmark Datasets] We employ various benchmark datasets [10] containing sample vectors of various dimensionalities. In each dataset, 10 pairs of training and test sets are used and the results are the mean of 10 trials.

First, the results of the performance for various threshold values τ of the criterion (Eq.(9)) are shown in Fig. 1. It is found that there is roughly *linear* relationship: 0.1 increase of Hinge loss causes about 2% decrease of the recognition rate, which is described by the dotted line in the figure. On the bais of the observation, we determine the threshold $\tau = 0.025$ for the criterion by assuming 0.5% decrease of recognition rate. The recognition rates of the classifier reduced by the proposed method are shown in Table 1, compared to those of Downs et al. [7]. The method of [7] yields the same performance as those of the original classifier and thus we show only the number of remained SVs for the method. It should be noted that the proposed method can effectively reduce SVs (about half) in all datasets while keeping the performance, even slightly outperforming the original result in several datasets. The complexity of the classifier is also reduced

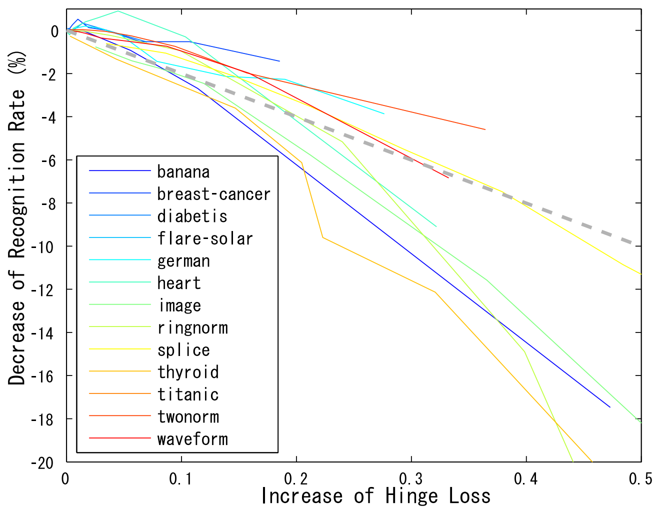

Fig. 1. The recognition rates with the criterion threshold values (increase of Hinge loss) for various datasets . There is roughly linear relationship as denoted by dashed line.

by the proposed method, which may cause higher generalization performance. On the other hand, the method [7] seeks the redundant SVs strictly satisfying Eq.(2) and the final reduced (remained) set of SVs is larger than that obtained by the proposed method which statistically approximates Eq.(2). In particular, the method [7] can not reduce SVs in some datasets, while our method achives the reduction in all datasets.

[MNIST Dataset] Next, we conducted an experiment on handwritten digit recognition, using MNIST dataset [11] which contains 10 types of digits from '0' to '9'. For training, we use 60000 images (6000 images for each digit class), while 10000 images (1000 images for each digit class) are used for test. We simply break down the recognition task to 10 individual recognition tasks.

The recognition rate and the number of remained SVs are shown in Table 2 for these 10 tasks. The SVs are greatly reduced in every task, while retaining the recognition performance: about a half reduction of SVs hardly affects the recognition performance. Then, we also investigate the number of SVs under varying the number of training samples. By focusing on the task of recognizing the digit '3', we applied downsampling to the training dataset with keeping the ratio of samples of each digit: from 6000 to 60000 samples (original dataset size). The results are shown in Table 3. As described in [4], the numbers of SVs are linearly increased along the number of training samples. The number of remained SVs in the reduction is also linearly increased but the increasing rate is significantly suppressed. The recognition rate is hardly decreased even though half or more of SVs (at $\tau = 0.025$) are eliminated. As the threshold τ becomes larger, the SVs are further reduced, but the decrease of the recognition rate becomes not negligible. This is similar tendency in Fig. 1, and the threshold $\tau = 0.025$ is also feasible for this dataset.

Table 1. Recognition rates and the number of finally remained SVs for various datasets [10] using the different methods. The numbers in the parentheses are the standard deviation.

	Original SVM		Proposed Method ($\tau = 0.025$)		Downs et al. [7]
Dataset	#SVs	Recog. Rate (%)	#SVs	Recog. Rate (%)	#SVs
banana	143 (± 23)	89.19 (± 0.73)	40 (± 24)	89.11 (± 0.80)	120 (± 37)
breast-cancer	124 (± 9)	73.12 (± 5.20)	51 (± 18)	72.99 (± 5.02)	119 (± 9)
diabetis	267 (± 13)	76.33 (± 1.80)	40 (± 17)	76.43 (± 1.97)	267 (± 13)
flare-solar	485 (± 19)	67.85 (± 2.20)	49 (± 2)	67.85 (± 2.20)	62 (± 3)
german	423 (± 30)	77.07 (± 1.97)	235 (± 88)	77.07 (± 1.67)	423 (± 30)
heart	95 (± 8)	83.60 (± 3.01)	33 (± 15)	83.90 (± 2.84)	95 (± 8)
image	320 (± 112)	96.75 (± 0.73)	266 (± 133)	96.26 (± 0.73)	312 (± 114)
ringnorm	97 (± 10)	98.10 (± 0.23)	73 (± 7)	98.02 (± 0.29)	97 (± 10)
splice	650 (± 67)	89.02 (± 0.75)	600 (± 65)	88.64 (± 0.73)	645 (± 68)
thyroid	27 (± 19)	95.07 (± 2.80)	19 (± 16)	94.27 (± 3.48)	27 (± 19)
titanic	76 (± 9)	77.33 (± 0.25)	8 (± 1)	77.36 (± 0.31)	10 (± 1)
twonorm	125 (± 5)	97.36 (± 0.19)	53 (± 33)	97.38 (± 0.15)	125 (± 5)
waveform	155 (± 22)	89.89 (± 0.51)	77 (± 25)	89.57 (± 0.70)	155 (± 22)

Table 2. Recognition rate and the number of finally remained SVs for MNIST [11].

	Original SVM		Proposed Method $\tau = 0.025$	
Class	Recog.	#SVs	Recog.	#SVs
0	99.75	1481	99.06	559
1	99.78	1251	99.75	708
2	99.43	2844	99.22	1227
3	99.47	3300	99.15	1479
4	99.47	2726	99.26	1042
5	99.44	3224	99.05	1293
6	99.64	1896	99.26	820
7	99.24	2442	99.08	933
8	99.31	3877	99.22	1968
9	99.00	4100	98.76	1733

Table 3. Recognition rate and the number of finally remained SVs by varying the number of training samples.

			Proposed Method					
Ratio of	Original SVM		$\tau = 0.025$		$\tau = 0.05$		$\tau = 0.075$	
Training size	Recog.	#SVs	Recog.	#SVs	Recog.	#SVs	Recog.	#SVs
0.1	98.74	706	98.28	380	97.43	283	96.34	202
0.2	99.09	1143	98.79	644	98.25	434	97.82	335
0.3	99.2	1468	98.75	764	97.92	490	97.81	373
0.4	99.21	1726	98.75	860	97.65	610	97.05	443
0.5	99.29	2100	99.05	1071	98.52	662	96.84	491
0.6	99.38	2309	98.92	1102	98.22	717	97.36	569
0.7	99.38	2550	99.09	1234	98.3	849	97.6	615
0.8	99.43	2851	99.04	1281	98.44	859	97.98	668
0.9	99.45	3106	99.04	1400	98.4	946	97.66	726
1	99.47	3300	99.15	1479	98.37	978	97.95	768

4. CONCLUSION

We have proposed an efficient method to reduce the computational cost of kernel-based classification. In the proposed method, once we calculate the inverse matrix of the kernel Gram matrix of support vectors, the redundant support vectors are sequentially identified and eliminated at quite low computational cost, while keeping the high performance of the classifier. For stopping the sequential reduction, we apply the criterion based on the increase of the Hinge loss calculated for training samples. In the experiments using various datasets, the proposed method effectively reduced the support vectors. In fact, even a half of the support vectors can be eliminated, preserving almost the same performance as that using the original set of support vectors. Note that the method is applicable to versatile kernel-based methods, not only SVM.

5. REFERENCES

[1] B. Schölkopf and A. Smola, Eds., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2002.

[2] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K. Müller, “Fisher discriminant analysis with kernels,” in *IEEE Neural Networks for Signal Processing Workshop*, 1999, pp. 41–48.

[3] V.N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[4] I. Steinwart, “Sparseness of support vector machines,” *Journal of Machine Learning Research*, vol. 4, pp. 1071–1105, 2003.

[5] C.J.C. Burges and B. Schölkopf, “Improving the accuracy and speed of support vector learning machines,” *Advances in Neural Information Processing Systems*, vol. 9, pp. 375–381, 1997.

[6] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K-R. Muller, G. Raetsch, and A.J.Smola, “Input space vs feature space in kernel-based methods,” *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000–1017, 1999.

[7] T. Downs, K.E. Gates, and A. Masters, “Exact simplification of support vector solutions,” *Journal of Machine Learning Research*, vol. 2, pp. 293–297, 2001.

[8] K.-M. Lin and C.-J. Lin, “A study on reduced support vector machines,” *IEEE Transactions on Neural Networks*, vol. 14, pp. 1449–1459, 2003.

[9] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds., *Large-Scale Kernel Machines*, MIT Press, 2007.

[10] http://ida.first.fraunhofer.de/projects/bench/.

[11] Y. LeCun and C. Cortes, “The mnist database of handwritten digits,” http://yann.lecun.com/exdb/mnist/.