Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007)

An Approach to the Learning Curves of an Incremental Support Vector Machines

Takemasa Yamasaki Kazushi Ikeda Yoshihiko Nomura
Graduate School of Informatics, Kyoto University
Kyoto 606-8501 Japan
Email: tyamasaki@sys.i.kyoto-u.ac.jp, kazushi@i.kyoto-u.ac.jp, y-nomura@sys.i.kyoto-u.ac.jp

***Abstract*— Support vector machines (SVMs) are known to result in a quadratic programming problem, that requires a large computational complexity. To overcome this problem, the authors proposed two incremental SVMs from the geometrical point of view in the previous study, both have a linear complexity with respect to the number of examples on average. One method was shown to produce the same solution as an SVM in batch mode, but the other, which stores the set of support vectors, was known to have a larger generalization error. In this study, we derive the learning curves of the latter method, assuming that the probability the set of support vectors is updated is proportional to the current margin and so is the decrease of the margin in the update, too. In the derivation, we employ the disc approximation which is to be justified yet, but the result agrees well with computer simulations.**

I. Introduction

A support vector machine (SVM) nonlinearly maps given input vectors to feature vectors in a high-dimensional space and linearly separates the feature vectors with an optimal hyperplane in terms of margin [1]–[5]. Since finding the optimal hyperplane results in a convex quadratic programming problem (QP) with linear constraints, it has an advantage that there are no local minima in the error surface, from which traditional gradient-based methods, such as multi-layer perceptrons [6], suffer in convergence. However, a QP requires a high computational complexity and even good QP solvers, such as interior-point methods, can solve problems of a limited size.

Another property of SVMs is that they have a sparse solution; that is, only a limited number of the examples contribute to the SVM solution while the others do not. This means that we could reduce the computational complexity if such useless examples could be removed in advance.

In order to reduce the complexity, we proposed two incremental methods in the previous study, based on the properties of SVMs mentioned above [7]. One can produce the same solution as that of the SVM in a batch mode, however, its implement is not easy. The other is simple and has a less complexity but its performance is a little worse. A rough geometrical analysis showed that the degradation of performance is limited; its generalization error has the same order as that of the SVM in a batch mode [7]. In this paper, we derive the learning curves more quantitatively based on the disc approximation. Although the disc approximation is to be justified yet, the theoretical learning curves agree well with those of computer simulations.

II. Effective Examples and Support Vectors

An SVM maps an input vector $\boldsymbol{x}$ to a vector $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{x})$ called a feature vector in the feature space. In this study, however, we employ the so-called linear kernel and assume that the feature vector is normalized. That is, $\|\boldsymbol{f}\| = \|\boldsymbol{f}(\boldsymbol{x})\| = \|\boldsymbol{x}\| = 1$ for any $\boldsymbol{x}$, as is done in [8]. In addition, we only consider SVMs with homogeneous separating hyperplanes, $\boldsymbol{w}^T\boldsymbol{f} = 0$, instead of inhomogeneous separating hyperplanes in the original SVMs, $\boldsymbol{w}^T\boldsymbol{f} + b = 0$, where ^{T} denotes the transposition. Note that a problem with inhomogeneous hyperplanes is easily transformed to one with homogeneous hyperplanes using the so-called lifting up (Fig. 1), $\tilde{\boldsymbol{w}}' := (\boldsymbol{w}', b)$ and $\tilde{\boldsymbol{f}}' := (\boldsymbol{f}', 1)$, where $:=$ means definition, though they differ a little since the latter also penalizes the bias b [9].

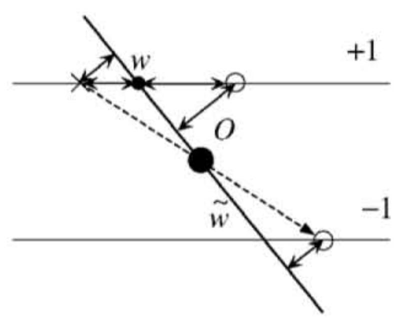

Fig. 1. Geometrical view of lifting up where the origin is denoted by O. Since the distances of examples from $\tilde{\boldsymbol{w}}$ (thick solid line) are proportional to those from $\boldsymbol{w}$ (black circle), lifting up does not change the problem of separating examples at all in terms of margin maximization. Neither does transforming a negative example (cross) to a positive one (white circle).

An SVM is given N examples and the ith example is a pair of an input vector $\boldsymbol{f}_i$ in the M-dimensional unit hypersphere S^M and the corresponding label $y_i \in \{\pm 1\}$ satisfying $y_i = \mathrm{sgn}(\boldsymbol{w}^{*T}\boldsymbol{f}_i)$, where $\boldsymbol{w}^*$ is the true weight vector to be estimated. Since the separating hyperplane is homogeneous, an example $(\boldsymbol{f}_i, y_i)$ is completely equivalent to $(y_i\boldsymbol{f}_i, 1)$ as seen in Fig. 1 and hence we can consider that any example has a positive label. In short, input vectors $\boldsymbol{f}$ are chosen $S_+^M = \{\boldsymbol{f} | \boldsymbol{f}^T\boldsymbol{w}^* > 0\}$, which we call the input space.

Since the magnitude of $\boldsymbol{w}$ does not affect its separation ability, we assume that $\boldsymbol{w} \in S^M$ without loss of generality where S^M is called the weight space. When an example $(\boldsymbol{f}_i, y_i)$ is given, the true vector $\boldsymbol{w}^*$ must be in the hyper-semisphere $\{\boldsymbol{w}|y_i\boldsymbol{w}^T\boldsymbol{f}_i > 0\}$. This means that an example is represented as a point in the input space and a hyperplane in the weight space (Fig. 2). On the other hand, a weight vector is represented as a hyperplane in the input space and a point in the weight space.

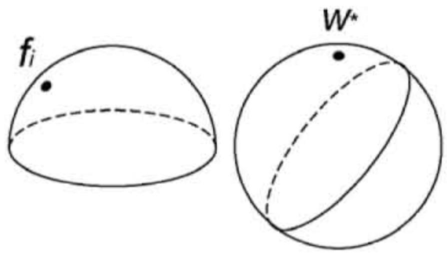

Fig. 2. An example in the input and weight spaces.

When N examples are given, $\boldsymbol{w}^*$ has to be in an area

$$A_N = \{\boldsymbol{w}|y_i\boldsymbol{w}^T\boldsymbol{f}_i > 0, i = 1, \ldots, N\}, \tag{1}$$

which we call the admissible region [10] (Fig. 3). The admissible region A_N, called the version space in physics, is a polyhedron in S^M. If the admissible region changes when an example is removed, the example is called effective. Note that the set of effective examples, referred to as the effective set, makes the same admissible region as all the examples. So, some algorithms for estimating $\boldsymbol{w}$, including SVMs, utilize only effective examples. This fact implies that any support vector is an effective example.

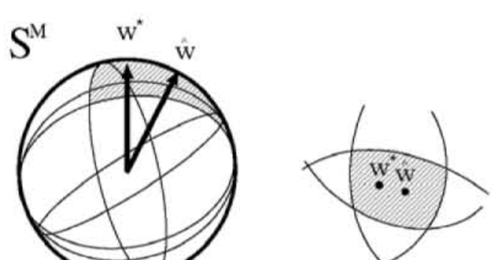

Fig. 3. Admissible region in the weight space

Under the assumption that the feature vectors are normalized, an SVM solution has a clear geometrical picture. Finding a hyperplane that maximizes the margin results in a quadratic programming problem,

$$\min_{\boldsymbol{w},\xi_i} \frac{1}{2}\|\boldsymbol{w}\|^2 \qquad \text{s.t. } \boldsymbol{w}'\boldsymbol{f}_i \geq 1. \tag{2}$$

It is known that the SVM solution $\hat{\boldsymbol{w}}$ necessarily has the form

$$\hat{\boldsymbol{w}} = \sum_{i=1}^{N} \alpha_i \boldsymbol{f}_i \tag{3}$$

where α_i are the Lagrangian multipliers. When $\alpha_i \neq 0$, $\boldsymbol{f}_i$ is called a support vector. In other words, $\hat{\boldsymbol{w}}$ consists of only support vectors. From the Karush-Kuhn-Tucker optimality conditions, support vectors $\boldsymbol{f}_i$ satisfy $\hat{\boldsymbol{w}}^T\boldsymbol{f}_i = 1$ and the others do not. This means that the SVM solution $\hat{\boldsymbol{w}}$ is equidistant from support vectors [8]. Since $\|\hat{\boldsymbol{w}}\|$ is not necessarily unity, we consider the meaning of the above in the weight space S^M. It is easily shown that the normalized $\hat{\boldsymbol{w}}$ in S^M (that is, $\hat{\boldsymbol{w}}/\|\hat{\boldsymbol{w}}\|$) is still equidistant from support vectors in the angular distance of S^M and the SVM solution $\hat{\boldsymbol{w}}$ is the center of maximum inscribed sphere in the admissible region A_N (Fig. 4) Note that the other examples are more distant from the center, even though they are effective. [11].

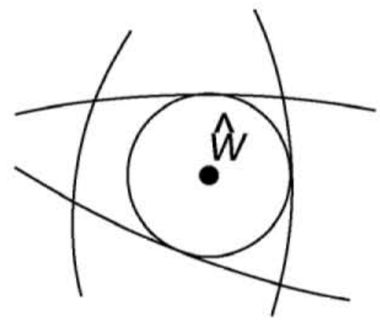

Fig. 4. The optimal weight $\hat{\boldsymbol{w}}$ is the center of maximum inscribed sphere in the admissible region.

III. Incremental SVMs

The discussion above claims that a learning machine can get the same information from only the set of effective examples. Thus, the incremental algorithm below referred to as Method 1, gives the same answer as the SVM in batch mode:

1. The machine has the effective set of n given examples.
2. Unless the $(n+1)$st example is effective, neglect it.
3. Otherwise, the effective set is remade, adding the $(n+1)$st example.

This algorithm has a low computational complexity in average, since the average number of effective examples does not depend on N [7], [10]. However, it is not easy to know whether an example is effective or not. To implement this, there are several packages, e.g. the function 'convhulln' in MATLAB, which is based on the Delaunay triangulation, but the complexity seems large.

To cope with the problem, we proposed another incremental method, referred to as Method 2, which stores support vectors instead of effective examples, since any support vector is effective by definition. Although there may be some loss in information, an example is easily determined whether it is a new support vector or not: the example is a support vector if and only if its distance from the current separating hyperplane is less than the current margin. Hence, Method 2 is written as below:

1. The machine has the set of support vectors of n given examples.
2. If the $(n+1)$st example is more distant from the separating hyperplane than the current margin, neglect it.
3. Otherwise, the set of support vectors is updated by an SVM solver with the support vectors and the $(n+1)$st example.

Method 2 neglects a new example that is effective but not a support vector. Since such a vector may become a support vector in the future, it is expected that Method 2 has a lower performance than a conventional SVM or Method 1. For instance, when a new example is located in (a) in Fig. 5, the example is thrown away in both incremental algorithms; in (b), each update procedure of the stored examples starts in both algorithms; in (c), an effective example is neglected in Method 2.

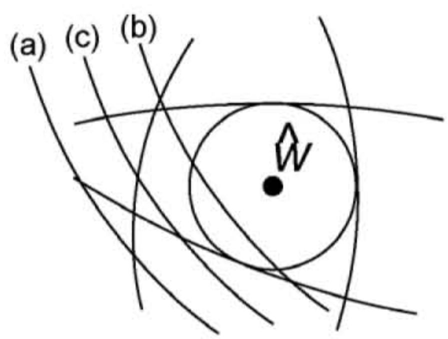

Fig. 5. A difference of the incremental algorithms appears in case (c).

IV. Learning Curves of Method 2

As mentioned before, Method 2 would have a lower performance than Method 1 since some effective examples are thrown away. How much is it? We give a more quantitative answer to this problem than [7].

We assume hereafter that examples are chosen from S_+^M uniformly and independently as well as a test input, as is done in [10]. The learning curves will be derived, as was in [7], based on the following two assumptions:

- The probability that the set of support vectors is updated is proportional to M_n.
- The decrease of the margin is also proportional to M_n.

The above assumptions lead to the following update equation

$$M_{n+1} = [1 - aM_n]M_n + aM_n[\lambda M_n] \tag{4}$$
$$= M_n - a[1-\lambda]M_n^2, \tag{5}$$

by simple calculation that leads to

$$M_N = \frac{c_{\rm ss}}{N} \qquad c_{\rm ss} = \frac{1}{a(1-\lambda)}. \tag{6}$$

We here introduce a new approximation, which we term the disc approximation, and evaluate the values of a and λ in (5). In short, the disc approximation regards the admissible region a disc.

The probability aM_n that the set of support vectors is updated is approximately expressed as the ratio of the radius of the admissible region to that of the hemisphere. In asymptotics of $N \to \infty$, the admissible region shrinks and can be regarded as a disc in a plane, however, the hemisphere cannot, since it is curved. Therefore, we evaluate an approximate of the radius of a hemisphere from its volume, using the fact that the volume is proportional to the radius power to M. As a result, the probability aM_n is evaluated as

$$aM_n = \left(\frac{\int_{S^{M-1}} \int_0^{M_n} \sin^{M-1} r \mathrm{d}r\mathrm{d}\omega}{\int_{S^{M-1}} \int_0^{\pi/2} \sin^{M-1} r \mathrm{d}r\mathrm{d}\omega} \right)^{1/M} \tag{7}$$
$$\approx \frac{M_n}{(MI_M)^{1/M}}, \tag{8}$$

where

$$I_M = \int_0^{\pi/2} \sin^{M-1} r \mathrm{d}r\mathrm{d}w = \frac{\sqrt{\pi}\Gamma[M/2]}{2\Gamma[(M+1)/2]}. \tag{9}$$

The decrease of the margin is also evaluated based on the volume of the admissible region. When the admissible region is a disc and the new example intersecting the region is distributed uniformly thereon, the decrease of the volume can be calculated as below, using the disc approximation and the radius-evaluation based on the volume, as before.

Suppose that the new example divides the admissible region A_n with radius M_n into two regions, $A_{n+1}^{\rm L}$ and $A_{n+1}^{\rm R}$, at $x = \theta \in (-M_n, M_n)$ (see Fig. 6). Then, the radius of the maximum inscribed sphere in $A_{n+1}^{\rm L}$ is $M_n + \theta$ and that in $A_{n+1}^{\rm R}$ is $M_n - \theta$. Based on the disc approximation, their volumes are written as

$$|A_{n+1}^{\rm L}| = |D^M|(M_n+\theta)^M, \tag{10}$$
$$|A_{n+1}^{\rm R}| = |D^M|(M_n-\theta)^M, \tag{11}$$
$$|A_n| = |D^M|M_n^M, \tag{12}$$

where $|D^M|$ is the volume of the unit M-dimensional disc. Taking into account that the probability of the true parameter being located in $A_{n+1}^{\rm L}$ is given as $|A_{n+1}^{\rm L}|/|A_n|$, the average ratio of the volume of the updated admissible region to the original is written as

$$\mathrm{E}\left[\frac{|A_{n+1}|}{|A_n|}\right] = \frac{1}{2M_n}\int_{-M_n}^{M_n} \left(\frac{|A_{n+1}^{\rm L}|}{A_n}\right)^2 + \left(\frac{|A_{n+1}^{\rm R}|}{A_n}\right)^2 \mathrm{d}\theta \tag{13}$$
$$= \frac{2}{2M+1}. \tag{14}$$

Then λ is

$$= \left(\frac{2}{2M+1}\right)^{1/M}. \tag{15}$$

In total, $c_{\rm ss}$ is expressed as

$$c_{\rm ss} = \frac{(MI_M)^{1/M}}{1 - \left(\frac{2}{2M+1}\right)^{1/M}} \tag{16}$$

from (6), (7) and (15).

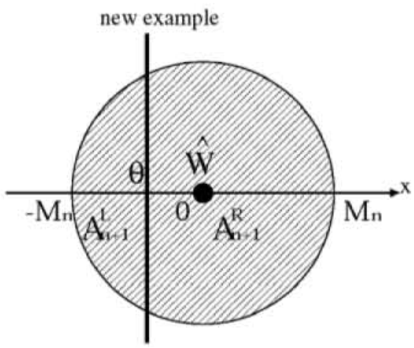

Fig. 6. The new example divides the admissible region into two regions at $x = \theta \in (-M_n, M_n)$

V. Computer Simulations

In order to confirm the validity of (16), some computer simulations were carried out. $N = 5000$ examples are chosen from S_+^M uniformly and independently and Method 2 learns the examples gradually.

Fig. 7 shows the average margins versus the number of examples, where the solid lines represent the theoretical results and dashed lines the experimental results for $M = 4$ and $M = 20$. It is clearly shown that the experimental curves in both figures approach the theoretical ones.

VI. Conclusions

In this paper, we analyzed Method 2 more quantitatively than [7], under the assumption that both the probability of the set of support vectors being updated and the decrease of the margin are proportional to the current margin. The disc approximation, we introduced here, makes it possible to evaluate their coefficients. The theoretical learning curves derived here agreed well the experimental results given by computer simulations.

Acknowledgment

This study is supported in part by a Grant-in-Aid for Scientific Research (15700130, 18300078) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

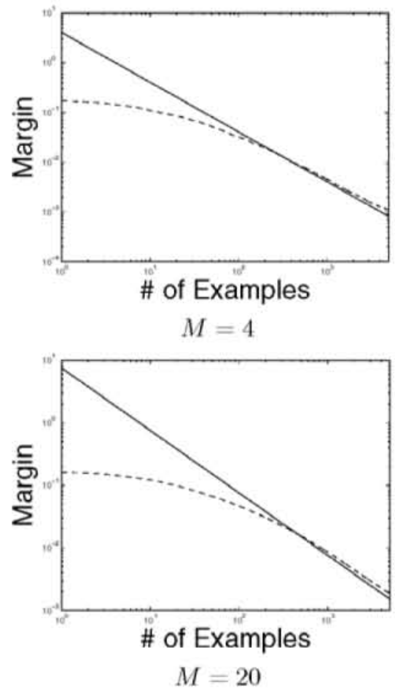

Fig. 7. Learning curves of Method 2.

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY: Springer-Verlag, 1995.

[2] ——, *Statistical Learning Theory*. New York, NY: John Wiley and Sons, 1998.

[3] B. Schölkopf, C. Burges, and A. J. Smola, *Advances in Kernel Methods: Support Vector Learning*. Cambridge, UK: Cambridge Univ. Press, 1998.

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge Univ. Press, 2000.

[5] A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds., *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 2000.

[6] D. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.

[7] K. Ikeda and T. Yamasaki, "Incremental support vector machines and their geometrical analyses," *Neurocomputing*, in press.

[8] R. Herbrich, *Learning Kernel Classifiers: Theory and Algorithms*. Cambridge, MA: MIT Press, 2002.

[9] K. Ikeda, "Geometrical properties of lifting-up in the nu support vector machines," *IEICE Trans. Information and Systems*, vol. E89-D, no. 2, pp. 847–852, 2006.

[10] K. Ikeda and S.-I. Amari, "Geometry of admissible parameter region in neural learning," *IEICE Trans. Fundamentals*, vol. E79-A, pp. 938–943, 1996.

[11] K. Ikeda and N. Murata, "Geometrical properties of nu support vector machines with different norms," *Neural Computation*, vol. 17, no. 11, pp. 2508–2529, 2005.