Bias Term *b* in SVMs Again

Te Ming Huang, Vojislav Kecman

School of Engineering, The University of Auckland, Auckland, New Zealand
e-mail: v.kecman@auckland.ac.nz, huangjh@win.co.nz

Abstract: The paper discusses and presents the use and calculation of the explicit bias term *b* in the support vector machines (SVMs) within the Iterative Single training Data learning Algorithm (ISDA). The approach proposed can be used for both nonlinear classification and nonlinear regression tasks. Unlike the other iterative methods in solving the SVMs learning problems containing the huge data sets, such as sequential minimal optimization (SMO) and its variants that must use at least two training data pairs, the algorithms shown here use the single training data based iteration routine for solving QP learning problem. In this way the various 2nd order heuristics in choosing the data for an updating is avoided. This makes the proposed ISD learning method remarkably quick. The algorithm can also be thought off as an application of a classic Gauss-Seidel (GS) coordinate ascent procedure and its derivative known as the successive over-relaxation (SOR) algorithm in SVMs learning from huge data sets subject to both the box constraints and the equality ones. (The later coming from minimizing the primal objective function in respect to the bias term *b*). The final solution in a dual domain is not an approximate one, but it is the optimal set of dual variables which would have been obtained by using any of existing and proven QP problem solvers if they only could deal with huge data sets.

1. Introduction

The development of the iterative learning schemes is the only way for the SVMs' learning when the training data set is huge (say more than 5,000 data pairs). It is the mainstream research field in the learning from empirical data by support vector machines. Recently, we (Kecman, Vogt, Huang, 2003) have shown that the kernel AdaTron (Anlauf, Biehl, 1989; Frieß, Cristianini, Campbell, 1998; Veropoulos, 2001) and SMO (Platt, 1999; Vogt, 2002), when the positive definite kernels are used without bias, are equal procedures. Even more, both are equal to the classic Gauss-Seidel (i.e., SOR) algorithm. We have also shown that such iterative 'single data based' learning algorithm (ISDA) converges to the optimal solution under the box constraints. In a matrix notation, the solution to both problems above is obtained by an iterative solving of the linear system of equations $\mathbf{K\alpha = f}$ subject to corresponding box constraints (see the details in the paper). Before presenting iterative algorithms with bias term, we discuss some recent presentations of the bias *b* utilization. It is well known that for positive definite kernels there is no need for bias *b* (Kecman, 2001). However, one can use it and this means implementing a different kernel. In (Poggio et all, 2001) it was also shown that when using positive definite kernels, one can choose between two types of solutions for both classification and regression. The first one uses the model without bias term (i.e., $f(\mathbf{x}) = \sum_{j=1}^{l} w_j K(\mathbf{x}, \mathbf{x}_j)$), while the second SVM uses

an explicit bias b. For the second one $f(\mathbf{x}) = \sum_{j=1}^{l} w_j K(\mathbf{x}, \mathbf{x}_j) + b$ and it was shown that $f(\mathbf{x})$ is a function resulting from a minimization of the functional shown below

$$I[f] = \sum_{j=1}^{l} V(y_j, f(\mathbf{x}_j)) + \lambda \|f\|_{K*}^2 \tag{1}$$

where $K^* = K - a$ (for an appropriate constant a) and K is an original kernel function (more details can be found in the mentioned report). This means that by adding a constant term to a positive definite kernel function K, one obtains the solution to the functional $I[f]$ where K^* is a conditionally positive definite kernel. Interestingly, similar type of model was also presented in (Mangasarian, Musicant, 1999). However, their formulation is done for the classification problems only. They reformulated the optimization by adding the $b^2/2$ term to the cost function $\|\mathbf{w}\|^2/2$. This is equivalent to an addition of 1 to the original kernel matrix $\mathbf{K}$. As a result, they changed the original classification dual problem to the optimization of the following one

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (K(\mathbf{x}_i, \mathbf{x}_j) + 1) \tag{2}$$

2. Iterative Single Data Algorithm (ISDA) for SVMs with Bias

In (Kecman, Vogt, Huang, 2003), for the SVMs' models when positive definite kernels are used without a bias term b, the learning algorithms for classification and regression (in a dual domain) were solved with box constraints only, originating from minimization of a primal Lagrangian in respect to the weights w_i. However, there remains an open question - how to apply the proposed ISD scheme for the SVMs that do use explicit bias term b. Such general nonlinear SVMs in classification and regression tasks are given below,

$$f(\mathbf{x}_i) = \sum_{j=1}^{l} y_j \alpha_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) + b = \sum_{j=1}^{l} w_j K(\mathbf{x}_i, \mathbf{x}_j) + b \tag{3a}$$

$$f(\mathbf{x}_i) = \sum_{j=1}^{l} (\alpha_j^* - \alpha_j) \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) + b = \sum_{j=1}^{l} w_j K(\mathbf{x}_i, \mathbf{x}_j) + b \tag{3b}$$

where $\Phi(\mathbf{x}_i)$ is the l-dimensional vector that maps n-dimensional input vector $\mathbf{x}$ into the feature space. (Note that for a classification model in (3a), we usually take the sign of $f(\mathbf{x})$ but this is of lesser importance now). For the SVMs' models (3), there are also *the equality constraints* originating from minimizing the primal objective function in respect to the bias b as given below,

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \qquad \text{(in a classification)}, \tag{4a}$$

and

$$\sum_{i=1}^{l} \alpha_i^* = \sum_{i=1}^{l} \alpha_i, \qquad \text{(in a regression)}. \tag{4b}$$

The motivation for developing the ISDA for the SVMs with an explicit bias term b originates from the fact that the use of an explicit bias b *seems to lead to the SVMs with less support vectors*. This fact can often be very useful for both the data (infor-

mation) compression and the speed of learning. Below, we present an iterative learning algorithm for the classification SVMs (3a) with an explicit bias *b*, subjected to the equality constraints (4a). (The same procedure is developed for the regression SVMs but due to the space constraints we do not go into these details here. However we give some relevant hints for the regression SVMs with bias *b*). The problem to solve is,

$$\min \frac{1}{2} \mathbf{w}^T \mathbf{w} \tag{5a}$$

$$\text{s.t. } y_i[\ \mathbf{w}^T \Phi(\mathbf{x}_i) + b\ ] \geq 1, \quad i = 1, \dots, l, \tag{5b}$$

which can be transformed into its dual form by minimizing the primal Lagrangian

$$L_p(\mathbf{w}, b, \mathbf{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^{l} \alpha_i \left\{ y_i \left[\mathbf{w}^T \Phi(\mathbf{x}_i) + b \right] - 1 \right\}, \tag{6}$$

in respect to $\mathbf{w}$ and b by using $\partial L_p / \partial \mathbf{w} = 0$ and $\partial L_p / \partial b = 0$, i.e., by exploiting

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i) \qquad \text{and} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0. \tag{7}$$

The standard change to a dual problem is to substitute $\mathbf{w}$ from (7) into the primal Lagrangian and this leads to a dual Lagrangian problem below,

$$L_d(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i y_i b \tag{8}$$

subject to the box constraints (9) and, in a standard SVMs formulation, also to the equality constraints (10) as given below

$$\alpha_i \geq 0, \quad i = 1, \dots, l \qquad \text{and} \qquad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{9}, (10)$$

There are *three major avenues* (procedures, algorithms) possible in solving the dual problem (8), (9) and (10).

The first one is the standard SVMs algorithm which imposes the equality constraints (10) during the optimization and in this way ensures that the solution never leaves a feasible region. In this case the last term in (8) vanishes. After the dual problem is solved, the bias term is calculated by using *unbounded* Lagrange multipliers α_i (Kecman, 2001; Schölkopf, Smola, 2002) as follows

$$b = \frac{1}{\#UnboundSVecs} \left(\sum_{i=1}^{\#UnboundSVecs} \left(y_i - \mathbf{w}^T \Phi(\mathbf{x}_i) \right) \right) \tag{11}$$

Note that in a standard SMO iterative scheme the minimal number of training data points enforcing (10) and ensuring staying in a feasible region is two.

Below, we show *two possible ways* how the ISDA works for the SVMs containing an explicit bias term too. In *the first method*, the cost function (5a) is augmented with the term $0.5kb^2$ (where $k \geq 0$). Note that this step is related to solving the dual problem

by penalty method where a decrease in k leads to the stronger imposing of equality constraints (see comments below). After forming the primal Lagrangian as well as using $\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \Phi(\mathbf{x}_i)$ and $b = \frac{1}{k} \sum_{i=1}^{l} \alpha_i y_i$ (coming from $\partial L_p / \partial \mathbf{w} = 0$ and $\partial L_p / \partial b = 0$) one arrives to the dual problem not containing the explicit bias b. Actually, the optimization of a dual Lagrangian is reformulated for the SVMs with a bias b by applying 'tiny' changes only to the original matrix $\mathbf{K}$. For the nonlinear classification problems ISDA stands for an iterative solving of the following linear system

$$\mathbf{K}_k \boldsymbol{\alpha} = \mathbf{1}_l \tag{12a}$$

$$\text{s.t. } 0 \le \alpha_i \le C, \quad i = 1, ..., l \tag{12b}$$

where $K_k(\mathbf{x}_i, \mathbf{x}_j) = y_i y_j (K(\mathbf{x}_i, \mathbf{x}_j) + 1/k)$, $\mathbf{1}_l$ is an l-dimensional unity vector and C is a penalty factor equal to infinity for a hard margin classifier. Note that during the updates of α_i, the bias term b must not be used because it is implicitly incorporated within the $\mathbf{K}_k$ matrix. Only after the solution vector $\boldsymbol{\alpha}$ in (12) is found, the bias b should be calculated either by using *unbounded* Lagrange multipliers α_i as given in (11), or by implementing the equality constraints from $\partial L_p / \partial b = 0$ and given as

$$b = \frac{1}{k} \sum_{j=1}^{\#SVecs} \alpha_j y_j . \tag{13}$$

Note, however, that all the Lagrange multipliers, meaning both bounded (clipped to C) and unbounded (smaller than C) must be used in (13). Both equations, (11) and (13), result in the same value for the bias b. Thus, using the SVMs with an explicit bias term means that in the ISDA proposed above original kernel is changed, i.e., another kernel function is used. This means that the alpha values will be different for each k chosen, and so will be the value for b. However, the final SVM as given in (3) is produced by original kernels. Namely, $f(\mathbf{x})$ is obtained by adding the sum of weighted original kernel values and corresponding bias term b.

The first method presented above and aimed at an extending of the ISDA to the SVMs with bias is related to the classic (quadratic) penalty methods for solving optimization problems with equality constraints. Namely, the addition of $0.5kb^2$ to (5a) changes the last term of (8) to $\frac{1}{2k} \left\| \sum_{i=1}^{l} \alpha_i y_i \right\|_2^2$, which is equivalent to applying a penalty parameter of $1/k$ to the L_2 norm of the equality constraint (10). As a result, for a large value of $1/k$, the solution will have a small L_2 norm of (10). In other words, as k approaches zero a bias b converges to the solution of the standard QP method that enforces the equality constraints. However, we do not use the ISDA with small parameter k values here, because the condition number of the matrix $\mathbf{K}_k$ increases as $1/k$ rises. Furthermore, the strict fulfillment of (10) may not be needed in obtaining a good SVM. Here, in classifying the MNIST data with Gaussian kernels, the value $k = 10$ proved to be a very good one justifying all the reasons for its introduction (fast learning, small number of support vectors and good generalization).

The second method in implementing the ISDA for SVMs with the bias term b is to work with original cost function (5a) and keep imposing the equality constraints during the iterations as suggested in (Veropoulos, 2001). The learning starts with $b = 0$ and after each epoch the bias b is updated by applying a secant method as follows

$$b^k = b^{k-1} - \omega^{k-1} \frac{b^{k-1} - b^{k-2}}{\omega^{k-1} - \omega^{k-2}} \tag{14}$$

where $\omega = \sum_{i=1}^{l} \alpha_i y_i$ represents the value of equality constraint after each epoch. In the case of regression SVMs, equation (14) is used by implementing the corresponding regression's equality constraints, namely $\omega = \sum_{i=1}^{l} \left(\alpha_i - \alpha_i^* \right)$. This is different from (Veropoulos, 2001) where an iterative update after each data pair is proposed. In our SVMs regression experiments such an updating led to an unstable learning. Also, in an addition to changing expression for ω, both the **K** matrix, which is now (2*l*, 2*l*) matrix, and the right hand side of (12a) which becomes (2*l*, 1) vector, should be changed too and formed as given in (Kecman, Vogt, Huang, 2003).

3. Performance of an ISD Learning Algorithm and Comparisons

To measure the relative performance of different ISDAs, we ran all the algorithms with RBF Gaussian kernels on a MNIST dataset with 576-dimensional inputs (Dong et all, 2003), and compared the performance of our ISD algorithm with LIBSVM V2.4 (Chang et all, 2003) which is one of the fastest and the most popular SVM solvers at the moment based on the SMO type of an algorithm. The MNIST dataset consists of 60,000 training and 10,000 test data pairs. To make sure that the comparison is based purely on the nature of the algorithm rather than on the differences in implementation, our encoding of the algorithms are the same as LIBSVM's ones in terms of caching strategy (LRU–Least Recent Used), data structure, heuristics for shrinking and stopping criterions. The only significant difference is that instead of two heuristic rules for selecting and updating two data points at each iteration step aiming at the maximal improvement of the dual objective function, our ISDA selects the worse KKT violator only and updates its α_i at each step.

Also, in order to speed up the LIBSVM's training process, we modified the original LIBSVM routine to perform faster by reducing the numbers of complete KKT checking without any deterioration of accuracy. All the routines were written and compiled in Visual C++ 6.0, and all simulations were run on a 2.4 GHz P4 processor PC with 1.5 Gigabyte of memory under the operating system Windows XP Professional. The shape parameter σ^2 of an RBF Gaussian kernel and the penalty factor C are set to be 0.3 and 10 (Dong, J.X. et all, 2003). The stopping criterion τ and the size of the cache used are 0.01 and 250 Megabytes. The simulation results of different ISDA against both LIBSVM are presented in tables 1 and 2, and in a figure 1. The first and the second column of the tables show the performance of the original and modified LIBSVM respectively. The last three columns show the results for single data point learning algorithms with various values of constant $1/k$ added to the kernel matrix in (12a). For $k = \infty$, ISDA is equivalent to the SVMs without bias term, and for $k = 1$, it is the same as the classification formulation proposed in (Mangasarian and Musicant, 1999).

Table 1: Simulation time for different algorithms

	LIBSVM original	**LIBSVM modified**	**Iterative single data algorithm (ISDA)** $k = 10$	$k = 1$	$k = \infty$
Class	Time(sec)	Time(sec)	Time(sec)	Time(sec)	Time(sec)
0	1606	885	**794**	800	1004
1	740	**465**	491	490	855
2	2377	1311	**1181**	1398	1296
3	2321	1307	**1160**	1318	1513
4	1997	1125	**1028**	1206	1235
5	2311	1289	**1143**	1295	1328
6	1474	818	**754**	808	1045
7	2027	1156	**1026**	2137	1250
8	2591	1499	**1321**	1631	1764
9	2255	1266	**1185**	1410	1651
Time Increase	**+95.3%**	**+10.3%**	**0**	**+23.9%**	**+28.3%**

Table 2: Number of support vectors for each algorithm

|	**LIBSVM original**	**LIBSVM modified**	**Iterative single data algorithm (ISDA)** $k = 10$	$k = 1$	$k = \infty$
Class	# SV (BSV)	# SV (BSV)	# SV (BSV)	# SV (BSV)	# SV (BSV)
0	2172 (0)	2172 (0)	2132 (0)	2162 (0)	**2682** (0)
1	1440 (4)	1440 (4)	1453 (4)	1429 (4)	**2373** (4)
2	3055 (0)	3055 (0)	3017 (0)	3047 (0)	**3327** (0)
3	2902 (0)	2902 (0)	2897 (0)	2888 (0)	**3723** (0)
4	2641 (0)	2641 (0)	2601 (0)	2623 (0)	**3096** (0)
5	2900 (0)	2900 (0)	2856 (0)	2884 (0)	**3275** (0)
6	2055 (0)	2055 (0)	2037 (0)	2042 (0)	**2761** (0)
7	2651 (4)	2651 (4)	2609 (4)	3315 (4)	**3139** (4)
8	3222 (0)	3222 (0)	3226 (0)	3267 (0)	**4224** (0)
9	2702 (2)	2702 (2)	2756 (2)	2733 (2)	**3914** (2)
Average # of SV	**2574**	**2574**	**2558**	**2639**	**3151**

BSV = Bounded SVs

Table 1 illustrates the running time for each algorithm. The ISDA with $k = 10$ was the quickest and required the shortest average time (T_{10}) to complete the training. The average time needed for the original LIBSVM is almost $2T_{10}$ and the average time for a modified version of LIBSVM is 10.3 % bigger than T_{10}. This is contributed mostly to the simplicity of the ISD algorithm. One may think that the improvement achieved is minor, but it is important to consider the fact that approximately more than 50% of the CPU time is spent on the final checking of the KKT conditions in all simulations. During the checking, the algorithm must calculate the output of the model at each datum in order to evaluate the KKT violations. This process is unavoidable if one wants to ensure the solution's global convergence, i.e. that *all the data* do satisfy the KKT conditions with precision τ indeed. Therefore, the reduction of time spent on iterations is approximately double the figures shown. Note that the ISDA slows down

for $k < 10$ here. This is a consequence of the fact that with a decrease in k there is an increase of the condition number of a matrix $\mathbf{K}_k$, which leads to more iterations in solving (12). At the same time, implementing the no-bias SVMs, i.e., working with $k = \infty$, also slows the learning down due to an increase in the number of support vectors needed when working without bias b.

Table 2 presents the numbers of support vectors selected. For the ISDAs, the numbers reduce significantly when the explicit bias term b is included. One can compare the numbers of SVs for the case without the bias b ($k = \infty$) and the ones when an explicit bias b is used (cases with $k = 10$ and $k = 1$). Because identifying less support vectors speeds the overall training definitely up, the SVMs implementations with an explicit bias b are faster than the version without bias.

In terms of a generalization, or a performance on a test data set, all algorithms had very similar results and this demonstrates that the ISDAs produce models that are as good as the standard QP, i.e., SMO based, algorithms. The percentages of the errors on the test data are shown in figure 1. Notice the extremely low error percentages on the test data sets for all numerals.

Figure 1: The percentage of the error on the test data
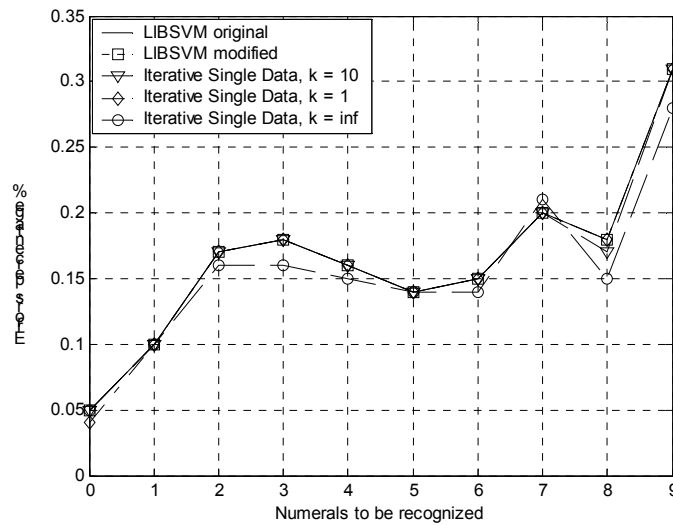
4. Conclusions

We demonstrate the use, the calculation and the effect of incorporating an explicit bias term b in the SVMs trained with the ISDA. The simulation results show that models generated by ISDAs (either with or without the bias term b) are as good as the standard QP (i.e., SMO) based algorithms in terms of a generalization performance. Moreover, ISDAs with an appropriate k value are faster than the standard SMO algorithms on large scale classification problems ($k = 10$ worked particularly well in all our simulations using Gaussian RBF kernels). This is due to both the simplicity of

ISDAs and the decrease in the number of SVs chosen after an inclusion of an explicit bias b in the model. The simplicity of ISDAs is the consequence of the fact that the equality constraints (4) do not need to be fulfilled during the training stage. In this way, the *second order heuristics is avoided* during the iterations. Thus, the ISDA is an extremely good tool for solving large scale SVMs problems containing huge training data sets because it is faster than, and it delivers 'same' generalization results as, the other standard QP (SMO) based algorithms. The fact that an introduction of an explicit bias b means solving the problem with different kernel suggests that it may be hard to tell in advance for what kind of previously unknown multivariable decision (regression) function the models with bias b may perform better, or may be more suitable, than the ones without it. As it is often the case, the real experimental results, their comparisons and the new theoretical developments should probably be able to tell one day. As for the single data based learning approach presented here, the future work will focus on the development of even faster training algorithms.

5. References

1. Anlauf, J. K., Biehl, M., The AdaTron - an adaptive perceptron algorithm. *Europhysics Letters*, 10(7), pp. 687–692, 1989
2. Frieß, T.-T., Cristianini, N., Campbell, I. C. G., The Kernel-Adatron: a Fast and Simple Learning Procedure for Support Vector Machines. In Shavlik, J., editor, *Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann, pp. 188–196, San Francisco, CA, 1998
3. Kecman, V., *Learning and Soft Computing, Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, The MIT Press, Cambridge, MA, http://www.support-vector.ws, 2001
4. Kecman, V., Vogt, M., Huang, T-M., On the Equality of Kernel AdaTron and Sequential Minimal Optimization in Classification and Regression Tasks and Alike Algorithms for Kernel Machines, Proc. of ESANN 2003, 11th European Symposium on Artificial Neural Networks, Bruges, Belgium, (downloadable from http://www.support-vector.ws), 2003
5. Platt, J.C., Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Ch. 12 in Advances in Kernel Methods – Support Vector Learning*, edited by B. Schölkopf, C. Burges, A. Smola, The MIT Press, Cambridge, MA, 1999
6. Schölkopf, B., Smola, A., *Learning with Kernels – Support Vector Machines, Optimization, and Beyond*, The MIT Press, Cambridge, MA, 2002
7. Veropoulos, K., *Machine Learning Approaches to Medical Decision Making*, PhD Thesis, The University of Bristol, Bristol, UK, 2001
8. Vogt, M., SMO Algorithms for Support Vector Machines without Bias, Institute Report, Institute of Automatic Control, TU Darmstadt, Darmstadt, Germany, (http://w3.rt.e-technik.tu-darmstadt.de/~vogt/), 2002
9. Poggio, T., Mukherjee, S., Rifkin, R., Rakhlin, A., Verri, A., *b,* CBCL Paper #198/AI Memo# 2001-011, Massachusetts Institute of Technology, Cambridge, MA, 2001
10. Mangasarian, O.L., Musicant, D.R., *Successive Overrelaxation for Support Vector Machines*, IEEE Trans. Neural Networks, 11(4), 1003-1008, 1999.
11. Dong, X., Krzyzak, A., Suen, C. Y., A fast SVM training algorithm, International Journal of Pattern Recognition and Artificial Intelligence, vol. 17, No. 3, pp. 367-384, 2003.
12. Chang, C., Lin, C., LIBSVM : a library for support vector machines, Available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2003.