[6] A. S. Poznyak, W. Yu, E. N. Sanchez, and J. P. Perez, "Nonlinear adaptive trajectory tracking using dynamic neural networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1402–1411, Nov. 1999.
[7] G. A. Rovithakis, "Tracking control of multi-input affine nonlinear dynamical systems with unknown nonlinearities using dynamical neural networks," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 179–189, Apr. 1999.
[8] S. S. Ge and C. Wang, "Adaptive neural control of uncertain MIMO nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 15, no. 3, pp. 674–692, May 2004.
[9] J. Q. Huang and F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time delay," *IEEE Trans. Neural Netw.*, vol. 14, no. 2, pp. 377–389, Mar. 2003.
[10] E. B. Kosmatopoulos, M. M. Polycarpou, M. A. Christodoulou, and P. A. Ioannou, "High-order neural network structures for identification of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 422–431, Mar. 1995.
[11] M. M. Polycapou and M. J. Mears, "Stable adaptive tracking of uncertain systems using nonlinearly parameterized on-line approximator," *Int. J. Control*, vol. 70, no. 3, pp. 363–384, 1998.
[12] X. M. Ren, A. B. Rad, P. T. Chan, and W. L. Lo, "Identification and control of continuous-time nonlinear systems via dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 478–486, Jun. 2003.
[13] D. Wang and J. Wang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 195–202, Jan. 2005.
[14] C. F. Hsu, C. M. Lin, and T. T. Lee, "Wavelet adaptive backstepping control for a class of nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1175–1183, Sep. 2006.
[15] Z. Man, H. R. Wu, S. Liu, and X. Yu, "A new adaptive backpropagation algorithm based on Lyapunov stability theory for neural networks," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1580–1590, Nov. 2006.
[16] *Automatica (Special Issue on Neural Network Feedback Control)*, vol. 37, Aug. 2001.
[17] A. Datta and J. Ochoa, "Adaptive internal model control: Design and stability analysis," *Automatica*, vol. 32, no. 2, pp. 261–266, 1996.
[18] W. Yu and X. Li, "Some new results on system identification with dynamic neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 412–417, Mar. 2002.
[19] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 69–79, Jan. 2000.
[20] E. B. Kosmatopoulos and M. A. Christodoulou, "High-order neural networks for robot contact surface shape," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 451–455, Jun. 1997.
[21] J. T. Spooner and K. M. Passino, "Stable adaptive control using fuzzy systems and neural networks," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 339–359, Aug. 1996.
[22] M. Zhang, S. Xu, and J. Fulcher, "Neuron-adaptive higher order neural—Network models for automated financial data modeling," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 188–204, Jan. 2002.
[23] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Upper Saddle River, N.J.: Prentice-Hall, 1996.
[24] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain*. Piscataway, NJ: IEEE Press, 2001.

Working Set Selection Using Functional Gain for LS-SVM

Liefeng Bo, Licheng Jiao, and Ling Wang

***Abstract*—The efficiency of sequential minimal optimization (SMO) depends strongly on the working set selection. This letter shows how the improvement of SMO in each iteration, named the functional gain (FG), is used to select the working set for least squares support vector machine (LS-SVM). We prove the convergence of the proposed method and give some theoretical support for its performance. Empirical comparisons demonstrate that our method is superior to the maximum violating pair (MVP) working set selection.**

***Index Terms*—Fast algorithm, least squares support vector machine (LS-SVM), sequential minimal optimization (SMO).**

I. Introduction

Support vector machines (SVMs) [1] are powerful tools for classification and regression. Least squares support vector machine (LS-SVM) [2] is a variant of SVMs which replaces the hinge loss function with the squared loss function. When no bias term is used in the LS-SVM formulation, similar expressions are obtained as with kernel ridge regression [3] and Gaussian processes regression [4].

LS-SVM is formulated as convex quadratic programming with equality constraint; hence, its solution is obtained by solving a set of linear equations. Although this problem is, in principle, solvable, in practice it is intractable for a large data set by the classical techniques, e.g., Gaussian elimination, because their computational complexity usually scales cubically with the size of training samples. To make LS-SVM applicable to large scale problems, Suykens *et al.* [5] presented a conjugate gradient (CG) algorithm. Chu *et al.* [6] gave an improved conjugate gradient algorithm. Keerthi and Shevade [7] proposed a sequential minimal optimization (SMO) algorithm where the maximum violating pair (MVP) is selected as the working set. Jiao *et al.* [8] developed a fast sparse approximation algorithm for LS-SVM. Empirical comparisons [6], [7] have shown that SMO is more efficient than CG and improved CG for the large scale data sets.

Inspired by [9] and [10], we present an improved working set selection using functional gain (FG) for LS-SVM. It selects the variable pair leading to a great functional gain as the working set. Although the working set selection using functional gain is first proposed for SVMs, intuitively, it is more natural for LS-SVM since it does not suffer from the boundary effects caused by inequality constraints ensuring the sparsity in SVMs. We prove that it achieves a greater or equal functional gain than the MVP method. Experiments show that the proposed method significantly reduces the training time of LS-SVM for large C values.

II. Working Set Selection Using FG

Consider a classification or regression problem with training samples $\{\mathbf{x}_i, y_i\}_{i=1}^{\ell}$ where $\mathbf{x}_i$ is the input sample and y_i is the corresponding target. Note that the variables in bold face denote the vector. In the

Manuscript received November 11, 2006; revised February 12, 2007; accepted February 17, 2007. This work was supported by the National Natural Science Foundation of China under Grant 60372050 and the National Defense Preresearch Foundation of China under Grant A1420060172.

The authors are with the Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China (e-mail: blf0218@163.com).

Digital Object Identifier 10.1109/TNN.2007.899715

feature space, LS-SVM takes the form $y = \mathbf{w}^T\varphi(\mathbf{x}) + b$ where the nonlinear mapping $\varphi(\mathbf{x})$ maps the input data into a high-dimensional feature space. To obtain a linear predictor, LS-SVM solves the following optimization problem:

$$\min \left\{ \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{2}\sum_{i=1}^{\ell} e_i^2 \right\}$$
$$\text{s.t.} \quad y_i = \mathbf{w}^T\varphi(\mathbf{x}_i) + b + e_i, \qquad i = 1, \ldots, \ell \tag{1}$$

where $C > 0$ is the regularization parameter. Its Wolfe dual problem is

$$\max \left\{ D(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i\alpha_j\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) - \sum_{i=1}^{\ell}\frac{\alpha_i^2}{2C} + \sum_{i=1}^{\ell}\alpha_i y_i \right\}$$
$$\text{s.t.} \quad \sum_{i=1}^{\ell}\alpha_i = 0. \tag{2}$$

The form $\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ in (2) is often replaced with a so-called positive–definite kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$, which can be expressed as the inner product of two vectors in some feature space and, therefore, can be used in LS-SVM. The Lagrangian for (2) is

$$\max_{\boldsymbol{\alpha},b} \left\{ D(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{\ell}\frac{\alpha_i^2}{2C} + \sum_{i=1}^{\ell}\alpha_i y_i + \beta\sum_{i=1}^{\ell}\alpha_i \right\}. \tag{3}$$

Define

$$F_i = F_i(\boldsymbol{\alpha}) = -\sum_j \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{C}\alpha_i + y_i. \tag{4}$$

The Karush–Kuhn–Tucker (KKT) conditions for the dual problem are

$$F_i + \beta = 0, \qquad \text{for } i = 1, 2, \ldots, \ell. \tag{5}$$

Keerthi and Shevade [7] suggested using SMO algorithm to solve (2). Its flowchart is shown in algorithm 1.

Algorithm 1: SMO algorithm for (2)

1) Set $k = 0, \boldsymbol{\alpha}^k = 0$, and $\mathbf{F}^k = \mathbf{F}(\boldsymbol{\alpha}^k) = \mathbf{y}$.

2) If the stop criterion is satisfied, stop. If not, select $p_1 = \arg\max_i(F_i^k)$ and $p_2 = \arg\min_i(F_i^k)$.

3) Solve the following subproblem with the variable t

$$t^{\text{opt}} = \arg\max_t \left\{ -\frac{1}{2}\begin{bmatrix} -t \\ t \end{bmatrix}^T \begin{bmatrix} k(\mathbf{x}_i, \mathbf{x}_i) + \frac{1}{C} & k(\mathbf{x}_i, \mathbf{x}_j) \\ k(\mathbf{x}_j, \mathbf{x}_i) & k(\mathbf{x}_j, \mathbf{x}_j) + \frac{1}{C} \end{bmatrix} \times \begin{bmatrix} -t \\ t \end{bmatrix} + \begin{bmatrix} -t \\ t \end{bmatrix}^T \begin{bmatrix} F_i^k \\ F_j^k \end{bmatrix} \right\}.$$

4) Set $\alpha_{p_1}^{k+1} = \alpha_{p_1}^k - t^{\text{opt}}$, $\alpha_{p_2}^{k+1} = \alpha_{p_2}^k + t^{\text{opt}}$, $F_i^{k+1} = F_i^k + t^{\text{opt}}k(\mathbf{x}_i, \mathbf{x}_{p_1}) - t^{\text{opt}}k(\mathbf{x}_i, \mathbf{x}_{p_2}), i \in \{1, \ldots, \ell\} \setminus \{p_1, p_2\}$, $F_{p_1}^{k+1} = F_{p_1}^k + t^{\text{opt}}(k(\mathbf{x}_{p_1}, \mathbf{x}_{p_1}) + (1)/(C)) - t^{\text{opt}}k(\mathbf{x}_{p_1}, \mathbf{x}_{p_2})$, $F_{p_2}^{k+1} = F_{p_2}^k + t^{\text{opt}}k(\mathbf{x}_{p_2}, \mathbf{x}_{p_1}) - t^{\text{opt}}(k(\mathbf{x}_{p_2}, \mathbf{x}_{p_2}) + (1)/(C))$, and $k = k + 1$; go back to step 2).

In the following, we will analyze the shortage of the MVP method and present our method. Suppose (α_i, α_j) is the current working variables and $\alpha_i^{k+1} = \alpha_i^k - t$. Together with the equality constraint, we have $\alpha_j^{k+1} = \alpha_j^k + t$. Thus, the functional gain of SMO in the current iteration can be written as the following:

$$D(\boldsymbol{\alpha}^{k+1}) - D(\boldsymbol{\alpha}^k) = g^k(i,j) = \max_t \left\{ -\frac{1}{2}\begin{bmatrix} -t \\ t \end{bmatrix}^T \times \begin{bmatrix} k(\mathbf{x}_i, \mathbf{x}_i) + \frac{1}{C} & k(\mathbf{x}_i, \mathbf{x}_j) \\ k(\mathbf{x}_j, \mathbf{x}_i) & k(\mathbf{x}_j, \mathbf{x}_j) + \frac{1}{C} \end{bmatrix} \times \begin{bmatrix} -t \\ t \end{bmatrix} + \begin{bmatrix} -t \\ t \end{bmatrix}^T \begin{bmatrix} F_i^k \\ F_j^k \end{bmatrix} \right\} \tag{7}$$

Solving (6), we get

$$t^{\text{opt}} = \frac{F_j^k - F_i^k}{\frac{2}{C} + \mu(i,j)} \tag{7}$$

where $\mu(i,j) = k(\mathbf{x}_i, \mathbf{x}_i) + k(\mathbf{x}_j, \mathbf{x}_j) - k(\mathbf{x}_i, \mathbf{x}_j) - k(\mathbf{x}_j, \mathbf{x}_i)$. Substituting t^{opt} into (6), we have

$$g^k(i,j) = \frac{\left(F_j^k - F_i^k\right)^2}{2\left(\frac{2}{C} + \mu(i,j)\right)}. \tag{8}$$

The key observation is that the maximum violating pair only maximizes the numerator in (8) without considering the effect of $\mu(i,j)$, possibly leading to a small gain. For a very small C, the effect of $\mu(i,j)$ can be ignored, so the MVP method is suitable. However, for a very large C, $\mu(i,j)$ plays an important role in the functional gain. Ideally, we want to select the variable pair maximizing (8). Unfortunately, this needs to evaluate (8) for all $\ell(\ell-1)/2$ possible variable pairs, which incurs a high computational cost. A simple alternative is to fix one variable and find the other by maximizing (8). This results in the following working set selection.

Algorithm 2: FGWSS (working set selection using functional gain)

1) Select $v_1 = \arg\max_i(\text{abs}(F_i^k))$.

2) Select $v_2 = \arg\max_i(g^k(v_1, i))$.

Some theoretical properties of the proposed working set selection are the following.

Theorem 1: For the same $\mathbf{F}^k$, FGWSS always gives a greater or equal functional gain than the MVP method.

Proof: There are two possibilities in step 1) of FGWSS. One is $v_1 = \arg\max_i(F_i^k) = p_1$ and the other is $v_1 = \arg\max_i(-F_i^k) = \arg\min_i(F_i^k) = p_2$. For the former case, we have

$$g^k(v_1, v_2) = \max_i(g^k(v_1, i)) \geq g^k(v_1, p_2) = g^k(p_1, p_2). \tag{9}$$

For the latter case, we have

$$g^k(v_1, v_2) = \max_i(g^k(v_1, i)) \geq g^k(v_1, p_1) = g^k(p_2, p_1) = g^k(p_1, p_2). \tag{10}$$

This completes the proof of Theorem 1.

Theorem 2: The sequence $\{\boldsymbol{\alpha}^k\}$ generated by SMO using FGWSS converges to the global optimal solution of (2).

TABLE I
COMPARISONS OF THE THREE ALGORITHMS ON THE MEDIUM AND LARGE SCALE DATA SETS. KERNEL DENOTES THE NUMBER OF KERNEL EVALUATIONS (WITHOUT CONSIDERING THE CACHE) WITH EACH UNIT CORRESPONDING TO 10^8 EVALUATIONS. DUAL DENOTES THE DUAL OBJECTIVE FUNCTION VALUE. TIME DENOTES THE TRAINING TIME WITH EACH UNIT CORRESPONDING TO 1 s

$\log_{10}(C)$	Adult4, 4781 samples with $\sigma^2 = 10$								
	CG			SMO using MVP			SMO using FG		
	Kernel	Dual	Time	Kernel	Dual	Time	Kernel	Dual	Time
-3	0.914	1.697	10.375	0.787	1.697	15.188	0.785	1.697	16.313
-2	1.828	14.026	11.047	0.710	14.026	13.750	0.710	14.026	15.109
-1	4.112	112.854	13.047	0.671	112.854	14.391	0.667	112.854	14.750
0	10.282	956.486	18.391	0.837	956.485	15.657	0.744	956.485	15.442
1	28.332	7296.708	34.047	3.242	7296.703	26.625	2.129	7296.704	24.078
2	84.768	42222.787	83.485	18.036	42222.749	46.234	8.703	42222.763	42.328
3	261.614	197013.627	237.141	76.768	197013.432	94.281	29.189	197013.527	80.234
Sum	391.850	247618.184	407.553	101.051	247617.946	225.766	**42.927**	247618.056	**208.234**
$\log_{10}(C)$	Adult7, 16100 samples with $\sigma^2 = 10$								
	CG			SMO using MVP			SMO using FG		
	Kernel	Dual	Time	Kernel	Dual	Time	Kernel	Dual	Time
-3	15.551	5.208	396.734	9.453	5.208	353.422	9.450	5.208	362.718
-2	31.101	41.458	756.218	8.003	41.457	295.531	8.030	41.457	303.484
-1	75.162	356.490	1774.000	7.257	356.490	258.593	7.264	356.490	265.203
0	204.750	3210.464	4767.391	9.583	3210.463	340.844	8.379	3210.463	303.469
1	572.783	27045.671	13250.562	38.893	27045.652	1224.922	24.082	27045.658	749.485
2	1775.368	189686.214	40946.766	262.862	189686.050	6033.843	114.094	189686.115	1794.765
3	5720.054	1135750.883	132282.828	1537.397	1135749.808	24855.703	524.150	1135750.292	10718.719
Sum	8394.769	1356096.387	194175.499	1873.447	1356095.126	33362.858	**695.447**	1356095.683	**15497.843**
$\log_{10}(C)$	Bank8fh, 8192 samples with $\sigma^2 = 10$								
	CG			SMO using MVP			SMO using FG		
	Kernel	Dual	Time	Kernel	Dual	Time	Kernel	Dual	Time
-3	2.684	0.521	12.891	1.744	0.521	13.907	1.749	0.521	15.735
-2	4.696	4.423	14.609	1.777	4.423	14.078	1.776	4.423	15.657
-1	10.064	23.794	19.282	1.677	23.794	13.922	1.673	23.794	15.401
0	16.773	147.247	25.078	1.756	147.246	13.688	1.560	147.246	14.547
1	41.597	1311.812	46.765	6.247	1311.780	28.812	2.145	1311.806	17.329
2	103.993	12692.571	100.782	49.052	12692.430	101.515	6.689	12692.512	32.422
3	299.232	124027.392	270.141	474.486	124025.938	687.594	52.285	124026.634	122.844
Sum	479.040	138207.759	489.548	536.740	138206.152	873.516	**67.876**	138206.935	**233.940**
$\log_{10}(C)$	House8l, 22784 samples with $\sigma^2 = 1$								
	CG			SMO using MVP			SMO using FG		
	Kernel	Dual	Time	Kernel	Dual	Time	Kernel	Dual	Time
-3	41.525	0.440	475.625	13.232	0.440	176.937	13.230	0.440	184.375
-2	77.859	3.318	879.391	12.190	3.318	162.906	12.199	3.318	169.829
-1	155.720	24.535	1744.047	10.794	24.535	144.766	10.823	24.535	150.406
0	358.155	198.157	3993.969	14.492	198.154	196.266	10.461	198.155	145.484
1	887.601	1741.838	9879.125	57.794	1741.804	761.437	17.461	1741.822	219.093
2	2522.656	15831.875	28037.407	470.266	15831.489	5767.265	94.658	15831.664	635.016
3	7573.160	145684.423	84162.797	4546.251	145680.445	46062.765	893.784	145681.703	5005.063
Sum	11616.677	163484.586	129171.361	5125.019	163480.186	53271.342	**1052.605**	163481.636	**6509.266**

Proof: Combining (7) and (8), we have

$$D(\boldsymbol{\alpha}^{k+1}) - D(\boldsymbol{\alpha}^k) = \frac{(t^{\mathrm{opt}})^2\left(\frac{2}{C} + \mu(i,j)\right)}{2}. \tag{11}$$

The positive–definite kernel function implies $\mu(i,j) \geq 0$. Together with $\|\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\|_2^2 = 2(t^{\mathrm{opt}})^2$, we have the following:

$$D(\boldsymbol{\alpha}^{k+1}) - D(\boldsymbol{\alpha}^k) \geq \frac{\|\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\|_2^2}{2C}. \tag{12}$$

Inequality (12) implies that $\{D(\boldsymbol{\alpha}^k)\}$ is a decreasing sequence. Together with $D(\boldsymbol{\alpha}^k) > -\infty$, we have that $\{D(\boldsymbol{\alpha}^k)\}$ converges. Applying (12) again, we get that $\{\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\}$ converges to 0.

Since $D(\boldsymbol{\alpha})$ is a positive–definite quadratic form, the set $\{\boldsymbol{\alpha} \,|\, D(\boldsymbol{\alpha}) \geq D(\boldsymbol{\alpha}^0)\}$ is a compact set. $\{\boldsymbol{\alpha}^k\}$ lies in this set, so it is a bounded sequence. Let $\bar{\boldsymbol{\alpha}}$ be the limit point of any convergent subsequence $\{\boldsymbol{\alpha}^k\}, k \in \Gamma$. Since there are only a finite number of variables, there exists at least one working set $\{v_1, v_2\}$ which occurs infinitely in this subsequence. Let $\Gamma^* \subseteq \Gamma$ be the set of the superscripts corresponding to $\{v_1, v_2\}$; then, we have

$$F_{v_1}(\bar{\boldsymbol{\alpha}}) - F_{v_2}(\bar{\boldsymbol{\alpha}}) = \lim_{k \to \infty, k \in \Gamma^*} (F_{v_1}(\boldsymbol{\alpha}^k) - F_{v_2}(\boldsymbol{\alpha}^k)). \tag{13}$$

According to [7, Lemma 1], (13) can be decomposed into

$$F_{v_1}(\bar{\boldsymbol{\alpha}}) - F_{v_2}(\bar{\boldsymbol{\alpha}}) = \lim_{k \to \infty, k \in \Gamma^*} (A_1(k) + A_2(k) + A_3(k)) \tag{14}$$

where $A_1(k) = F_{v_1}(\boldsymbol{\alpha}^k) - F_{v_1}(\boldsymbol{\alpha}^{k+1}), A_2(k) = F_{v_1}(\boldsymbol{\alpha}^{k+1}) - F_{v_2}(\boldsymbol{\alpha}^{k+1})$,and $A_3(k) = F_{v_2}(\boldsymbol{\alpha}^{k+1}) - F_{v_2}(\boldsymbol{\alpha}^k)$. From step 4) of SMO, we know

$$A_2(k) = 0. \tag{15}$$

Since $\{\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k\}$ converges to 0, $\lim_{k \to \infty, k \in \Gamma^*} A_1(k) = 0$ and $\lim_{k \to \infty, k \in \Gamma^*} A_3(k) = 0$. Thus, we get

$$F_{v_1}(\bar{\boldsymbol{\alpha}}) - F_{v_2}(\bar{\boldsymbol{\alpha}}) = 0. \tag{16}$$

According to Theorem 1, we have

$$\frac{\left(F_{v_1}(\boldsymbol{\alpha}^k)-F_{v_2}(\boldsymbol{\alpha}^k)\right)^2}{2\left(\frac{2}{C}+\mu(v_1,v_2)\right)} \geq \frac{\left(F_i(\boldsymbol{\alpha}^k)-F_j(\boldsymbol{\alpha}^k)\right)^2}{2\left(\frac{2}{C}+\mu(i,j)\right)} \qquad \forall i,j \in \{1,\ldots,\ell\}. \tag{17}$$

Considering the limit of (17), we get

$$\begin{aligned}
&(F_i(\bar{\boldsymbol{\alpha}}) - F_j(\bar{\boldsymbol{\alpha}}))^2 \\
&\quad = \left(\lim_{k\to\infty, k\in\Gamma^*}(F_i(\boldsymbol{\alpha}^k) - F_j(\boldsymbol{\alpha}^k))\right)^2 \\
&\quad \leq \frac{2+C\mu(i,j)}{2+C\mu(v_1,v_2)}\left(\lim_{k\to\infty, k\in\Gamma^*}(F_{v_1}(\boldsymbol{\alpha}^k) - F_{v_2}(\boldsymbol{\alpha}^k))\right)^2 \\
&\quad = \frac{2+C\mu(i,j)}{2+C\mu(v_1,v_2)}(F_{v_1}(\bar{\boldsymbol{\alpha}}) - F_{v_2}(\bar{\boldsymbol{\alpha}}))^2 \\
&\quad = 0 \qquad \forall i,j \in \{1,\ldots,\ell\}.
\end{aligned} \tag{18}$$

Equation (18) implies $F_1(\bar{\boldsymbol{\alpha}}) = F_2(\bar{\boldsymbol{\alpha}}), \ldots, = F_\ell(\bar{\boldsymbol{\alpha}})$. From the KKT conditions, $\bar{\boldsymbol{\alpha}}$ is the global optimal solution of (2). Since $D(\boldsymbol{\alpha})$ is strictly convex, (2) has a unique global optimal solution and we denote it as $\boldsymbol{\alpha}^*$. Assume that $\{\boldsymbol{\alpha}^k\}$ does not converge to $\boldsymbol{\alpha}^*$. Then, $\forall \epsilon > 0$, there exists an infinite subset $\Gamma^{'}$ such that $\|\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^*\| > \epsilon, \forall k \in \Gamma^{'}$. Because $\{\boldsymbol{\alpha}^k\}, \forall k \in \Gamma^{'}$ is a compact set, there is a convergent subsequence. Without loss of generality, we assume its limit to be $\bar{\boldsymbol{\alpha}}$. Thus, we have $\|\bar{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*\| > \epsilon$. Since $\bar{\boldsymbol{\alpha}}$ is the global optimal solution of (2), this contradicts that $\boldsymbol{\alpha}^*$ is the unique global optimal solution. The proof of Theorem 2 is completed.

III. Empirical Study

In order to evaluate the performance of the proposed method, we compare it with SMO and improved CG on four benchmark data sets. All the three algorithms are implemented in VC++ 6.0 and are run on a personal computer with 2.4-GHz processors, 1.5-GB memory and Windows XP operation systems. The size of the cache is set to 800 MB. The optimization process is terminated when the maximal violation of the KKT conditions is within 0.001.[1] For the regression data sets, both the input and the output are scaled into the interval $[-1, 1]$.

The Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/2\sigma^2)$ is used to construct LS-SVM. For Adult4 and Adult7 data sets, the values of σ^2 are the same as in [11]. For Bank8fh and House8l data sets, the values of σ^2 are determined by the tenfold cross validation on a small random subset.

Table I reports the number of kernel evaluations and the training time of CG, SMO using MVP, and SMO using FG. As we can see, our method beats its competitors and achieves the better performance on the cases we have studied. Our method significantly outperforms CG, in particular, for the large scale data sets. At the small C values, our method exhibits similar performance with the MVP method; however, at the large C values, our method significantly outperforms the MVP method. The discussions below (8) explain the reason. Note that for medium scale problems, the whole kernel matrix can be fitted into the cache, so the real number of kernel evaluations is at most ℓ^2, which explains why the training time does not match the number of kernel evaluations shown in Table I.

[1]The source code for our method is available at http://see.xidian.edu.cn/graduate/lfbo/. The classification data sets Adult4 and Adult7 come from http://www.reasearch.microsoft.com/~jplatt/smo.html. The regression data sets Bank8fh and House8l can be accessed at http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html

IV. Conclusion

In this letter, we have proposed a new method for selecting the working set for LS-SVM and proved its asymptotic convergence. Our method effectively utilizes the functional gain information and achieves fast convergence. Empirical comparisons demonstrate that the new algorithm is significantly faster than other existing algorithms.

Acknowledgment

The authors would like to thank the reviewers for their helpful comments.

References

[1] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[2] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
[3] C. Saunders, A. Gammerman, and K. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. 15th Int. Conf. Mach. Learn.*, 1998, pp. 515–521.
[4] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems 8*. Cambridge, MA: MIT Press, 1996, pp. 514–520.
[5] J. A. K. Suykens, L. Lukas, P. Van Dooren, B. De Moor, and J. Vandewalle, "Least squares support vector machine classifiers: A large scale algorithm," in *Proc. Euro. Conf. Circuit Theory Design*, 1999, pp. 839–842.
[6] W. Chu, C. J. Ong, and S. S. Keerthy, "An improved conjugate gradient method scheme to the solution of least squares SVM," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 498–501, Mar. 2005.
[7] S. S. Keerthi and S. K. Shevade, "SMO for least squares SVM formulations," *Neural Comput.*, vol. 15, pp. 487–507, 2003.
[8] L. C. Jiao, L. F. Bo, and L. Wang, "Fast sparse approximation for least squares support vector machine," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 685–697, May 2007.
[9] T. Glasmachers and C. Igel, "Maximum-gain working set selection for SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1437–1466, 2006.
[10] R. E. Fan, P. H. Chen, and C. J. Lin, "Working set selection using second order information for training support vector machines," *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, 2005.
[11] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," *Advance in Kernel Methods—Support Vector Learning*, pp. 185–208, 1999.