World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:9, 2011

Approximation Incremental Training Algorithm Based on a Changeable Training Set

Yi-Fan Zhu, Wei Zhang, Xuan Zhou, Qun Li, Yong-Lin Lei

Abstract—The quick training algorithms and accurate solution procedure for incremental learning aim at improving the efficiency of training of SVR, whereas there are some disadvantages for them, i.e. the nonconvergence of the formers for changeable training set and the inefficiency of the latter for a massive dataset. In order to handle the problems, a new training algorithm for a changeable training set, named Approximation Incremental Training Algorithm (AITA), was proposed. This paper explored the reason of nonconvergence theoretically and discussed the realization of AITA, and finally demonstrated the benefits of AITA both on precision and efficiency.

Keywords—support vector regression, incremental learning, changeable training set, quick training algorithm, accurate solution procedure

I. Introduction

DURING the past decades, Support Vector Machine (SVM) [1] has been successfully applied in the field of machine learning, such as the pattern recognition [2]–[4], regression and approximation (referred to as support vector regression, SVR) [5], etc. The theoretical foundation of SVR is statistical learning theory (SLT), which is a specific theory for studying learning machine under the small sample condition [1]. The key part of SLT is the introduction of the structural risk minimization (SRM) principle to control the capacity of learning machine for considering both the asymptotic performance and the capacity of obtaining the optima result under an limited information condition. Based on the SLT, SVR possesses many advantages, e.g. no local optima (convex problem with a unique solution), good ability of generalization, intrinsic regularization [6], and the sparseness of support vectors (SV) as well as the robustness to outliers using ε-insensitive loss function. The studies have shown that SVR can obtain good performance of approximation or prediction in various applications including function approximation [6], [7], prediction [8] and other simulation applications [9].

The basic idea of SVR is as follows: select a function $\Phi(\cdot)$ to map the training set from original input space to high-dimension feature space $\mathcal{H}$ and construct an optimal linear regression function in $\mathcal{H}$ as follows:

$$f(x) =< w, \Phi(x) >_{\mathcal{H}} +b \tag{1}$$

where $< \cdot, \cdot >_{\mathcal{H}}$ ($< \cdot, \cdot >$ for simplicity) denotes the inner product in $\mathcal{H}$, and $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ are the weight vector and bias respectively. In order to meet the SRM principle, the w should be as small as possible to ensure the flatness of function f. Note that it's finally equivalent to solve a constrained optimization problem [5] as follows:

$$\begin{aligned} &\min_{w,b,\xi,\xi^*} \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \\ s.t. &\begin{cases} f(x_i) - y_i \le \varepsilon + \xi_i & , \quad i = 1, ..., N \\ y_i - f(x_i) \le \varepsilon + \xi_i^* & , \quad i = 1, ..., N \\ \xi_i, \xi_i^* \ge 0 & , \quad i = 1, ..., N \end{cases} \end{aligned} \tag{2}$$

where the minimization of the first term is to improve the generalization capacity and the second term is to reduce the error, constant $C > 0$ determines the trade-off between these two minimizations and $\varepsilon \ge 0$ is a priori chosen constant to control the noise tolerances. Finally, the following regression result can be trained using duality theory and Karush-Kuhn-Tucker (KKT) conditions with introducing kernel function K:

$$\begin{aligned} f(x) &= \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\, K(x_i, x) + b \\ &= \sum_{i \in \mathcal{SV}}(\alpha_i - \alpha_i^*)\, K(x_i, x) + b \end{aligned} \tag{3}$$

where $\mathcal{SV}$ is the index set of SVs. It's obvious that standard SVR training algorithm can always obtain the global optimal solution since the regression problem is transformed into a convex quadratic programming with a unique solution, and avoids the local optimal in neural networks.

However, the standard SVR training algorithm is confronted with a serious problems, i.e. it's incapable of dealing with mass samples since the limitation of storage and computation of Hessian matrix. Therefore, many improved training algorithms with decomposition (referred as quick training algorithm, QTA) are proposed. The main ideas for these quick training algorithms are classified as following two catalogues:

1) One is "Increasing", which takes Chunking algorithm [1] as representation. It obtained the SV set and corresponding Lagrange multipliers by selecting working set and subsequently to update the working set. The M samples, which possess the most serious violations with the KKT conditions, are selected into the working set and repeat the process above until all of the samples meet the KKT conditions.

Yi-Fan Zhu is with the College of Information Systems and Management, NUDT, Changsha,Hunan, China (e-mail: nudtzyf@hotmail.com).
Wei Zhang is with the College of Information Systems and Management, National University of Defense Technology (NUDT), Changsha, Hunan, 410073, China (e-mail: the_ant@163.com).
Xuan Zhou is with the College of Information Systems and Management, National University of Defense Technology (NUDT), Changsha, Hunan, 410073, China
Qun Li is with the College of Information Systems and Management, National University of Defense Technology (NUDT), Changsha, Hunan, 410073, China
Yong-Lin Lei is with the College of Information Systems and Management, National University of Defense Technology (NUDT), Changsha, Hunan, 410073, China

2) The other is "decomposing", whose examples are Osuna [10] and SMO [11]. The main idea of these algorithms is that the big quadratic programming is decomposed into manageable small ones over part of the data and any sample in the working set is replaced by the sample which is in the non-working set and disobeys the KKT conditions. The process is repeated until all the samples meet the KKT conditions.

However, these QTAs are working with a fixed working set and the generalization analysis of SVR are dealing with the new test samples, which possess no impact on the generalization analysis theoretically since they do not belong to the training set. Whereas when the training set is changeable, there are little effective approaches for these algorithms. There are two familiar approaches. One is to discard all previous result completely and train SVR with new training set, and the other is to train SVR constrainedly based on the new training set, while there is a same disadvantage that these techniques may give an approximate solution, and may require many passes through the dataset to reach a reasonable level of convergence [12], [13]. Cauwenberghs *et al.* [13] proposed an accurate solution for incremental learning (referred as accurate solution procedure, ASP), which is to compute the impact on Lagrange coefficients and Support Vectors (SV) when appending or removing a training sample. Whereas ASP cannot be employed in regression problems directly as it is proposed for classification problems. Although Ma *et al.* [14] introduced this algorithm for regression analysis, it's ineffective for increasing dateset.

A new training algorithm, i.e. Approximation Incremental Training Algorithm (AITA) was proposed in this paper. The AITA is hybrid with the QTA and ASP and the performance of precision and efficiency are demonstrated by a synthetic problem under different test schemes.

Notations: all matrices are written with uppercase letters, e.g. A, and the ith column of a matrix A that is denoted A_i . Matrix $X = \left(x_i^j\right)_{1\le i\le N, 1\le j\le d} \in \mathbb{R}^{N\times d}$ denotes the sample matrix, where N is the amount of samples and d is the dimension of input. Letter in lowercase, e.g. $x \in \mathbb{R}^d$ denotes a sample and $X = (x_1, ..., x_N)^T$, where the ith sample is denoted $x_i = (x_i^1, ..., x_i^d) \in \mathbb{R}^d$. Vector $Y = (y_1, ..., y_N)^T \in \mathbb{R}^N$ is the output of X. $K(t, s)$ denotes the kernel in $\mathbb{R}^d \times \mathbb{R}^d$ and $\mathcal{K}$ is the corresponding Gram matrix, i.e. $\mathcal{K} = \mathcal{K}(X^T, X^T) = (K(x_i, x_j))_{N\times N}$. Kernel matrix $\mathcal{K}(t, X^T)$ denotes the $1 \times N$ matrix consists of the kernel values between some input $t = (t^1, \cdots, t^d) \in \mathbb{R}^d$ and the N samples in sample matrix $X \in \mathbb{R}^{N\times d}$, i.e.

$$\begin{aligned}\mathcal{K}(t, X^T) &= \mathcal{K}\left(t, \begin{pmatrix} x_1^1 & \cdots & x_{N1} \\ \vdots & \ddots & \vdots \\ x_1^d & \cdots & x_{Nd} \end{pmatrix}\right) \\ &= \begin{pmatrix} K(t, x_1), & \cdots, & K(t, x_N) \end{pmatrix}_{1\times N}\end{aligned} \tag{4}$$

$\forall g(x) : \mathbb{R}^d \mapsto \mathbb{R}, g(X^T) = (g(x_1), \cdots, g(x_N))^T$. $e = (1, \cdots, 1)^T \in \mathbb{R}^N$ denotes the vector whose components equal to 1 and $e_m = (1, \cdots, 1)^T \in \mathbb{R}^m$

*Defintion 1 (**Modified sample**):* A sample is called modified sample if it is different with the existing samples in training set when the SVR is being trained.

II. Approximation Incremental Training Algorithm

A. Problem Analysis

It's known that the predicted value is irrelevant to the sample which is not SV (NoSV), since the regression function (3) is actual the linear combination of kernel functions of SVs. This property is called "sparseness", which determines that the complex of computation of SV is related to the amount of SV other than the dimension of input space. Therefore, there is no impact on the training result if the modified samples are NoSVs, even though the training set is changeable. It's known that the SVs are determined automatically by solving a convex quadratic programming with linear constraints, that is the programming (2), and its necessary and sufficient conditions are Karush-Kuhn-Tucker(KKT) conditions as shown in following lemma.

Lemma 1: Let Q be the Hessian matrix of convex quadratic programming (2) and any feasible solution α is the optimal solution if and only if all the samples must meet the KKT conditions which are determined by Lagrange multipliers α.

For further details can be seen in [11]. It's obvious that the properties of Lagrange multipliers $\alpha^{(*)}$ and their relations with SVs as follows:

Theorem 1: Let $\bar{\alpha}^{(*)} = (\bar{\alpha}_1^{(*)}, ..., \bar{\alpha}_N^{(*)})^T$ be the optimal solutions of programming (2), for $\forall i \in \{1, 2, ..., N\}$, then $\bar{\alpha}_i\bar{\alpha}_i^* = 0, \bar{\xi}_i\bar{\xi}_i^* = 0$.

Proof: According to the KKT conditions and complementary slackness conditions, the programming (2) can be transformed into its dual problem (a matrix form) as follows:

$$\begin{aligned}&\min \tfrac{1}{2}(\alpha - \alpha^*)^T\mathcal{K}(X^T, X^T)(\alpha - \alpha^*) \\ &+\varepsilon e^T(\alpha + \alpha^*) - Y^T(\alpha - \alpha^*)\end{aligned} \tag{5}$$

$$s.t. \quad \begin{cases} e^T(\alpha - \alpha^*) = 0 \quad , \quad \alpha \ge 0, \alpha^* \ge 0 \\ Ce - \alpha \ge 0 \quad , \quad Ce - \alpha^* \ge 0 \end{cases} \tag{6}$$

Then there are $\bar{\lambda}^{(*)}, \bar{\xi}^{(*)}$ such that the optimal $\bar{\alpha}^{(*)}$ make the following equations hold:

$$\begin{aligned}\mathcal{K}(X^T, X^T)(\bar{\alpha} - \bar{\alpha}^*) + \varepsilon e - Y + \bar{b}e - \bar{\lambda} + \bar{\xi} = 0 \\ \mathcal{K}(X^T, X^T)(\bar{\alpha} - \bar{\alpha}^*) - \varepsilon e - Y + \bar{b}e + \bar{\lambda}^* - \bar{\xi}^* = 0 \\ \bar{\lambda}, \bar{\lambda}^* \ge 0, \bar{\xi}, \bar{\xi}^* \ge 0\end{aligned} \tag{7}$$

and meet the complementary slackness conditions

$$\begin{aligned}\bar{\lambda}^T\bar{\alpha} &= 0 \quad , \quad \bar{\lambda}^{*T}\bar{\alpha}^* &= 0 \\ \bar{\xi}^T(Ce - \bar{\alpha}) &= 0 \quad , \quad \bar{\xi}^{*T}(Ce - \bar{\alpha^*}) &= 0\end{aligned} \tag{8}$$

1) If $\bar{\alpha}_i = 0$ or $\bar{\alpha}_i^* = 0$, then $\bar{\alpha}_i\bar{\alpha}_i^* = 0$. According to (8), if $\bar{\alpha}_i = 0$ or $\bar{\alpha}_i^* = 0$, then the corresponding $\bar{\xi}_i = 0$ or $\bar{\xi}_i^* = 0$, which achieves our assertion.

2) If $0 < \bar{\alpha}_i < C$, according to (7) and (8), then

$$\bar{\alpha}_i(\varepsilon + \bar{\xi}_i + y_i - f(x_i)) = 0 \tag{9}$$

$$(C - \bar{\alpha}_i)\bar{\xi}_i = 0 \tag{10}$$

and $\bar{\xi}_i = 0$, $\varepsilon + \bar{\xi}_i + y_i - f(x_i) = 0$, that is

$$y_i - f(x_i) + \bar{\xi}_i = -\varepsilon < \varepsilon \tag{11}$$

According to the constraints of programming (2), i.e. $y_i - f(x_i) \leq \varepsilon + \bar{\xi}_i^*$, then $\bar{\xi}^* = 0$. And according to (7), then

$$\bar{\alpha}_i^*(\varepsilon + \bar{\xi}_i^* - y_i + f(x_i)) = 0 \tag{12}$$

Substituting (11) into (12), then $\bar{\alpha}_i^* = 0$, which achieves the assertion. In the same way that for $0 < \bar{\alpha}_i^* < C$, then $\bar{\alpha}_i = 0, \bar{\xi}_i \bar{\xi}_i^* = 0$.

3) If $\bar{\alpha}_i = C$, then $\bar{\xi}_i \geq 0$ according to (10). There are two cases, that is,

- If $\bar{\xi}_i = 0$, then $y_i - f(x_i) = -\varepsilon < \varepsilon$ according to (9). The proof is as same as 2)
- if $\bar{\xi}_i > 0$, then $y_i - f(x_i) = -\varepsilon - \bar{\xi} < -\varepsilon$ according to (9). Therefore, according to the constraints of programming (2) , that is $y_i - f(x_i) \leq \varepsilon + \bar{\xi}_i^*$, we have $\bar{\xi}_i^* = 0$. And according to (12), then $\bar{\alpha}_i^* = 0$.

As the cases above, for $\bar{\alpha}_i = C$, then $\bar{\alpha}_i^* = 0, \bar{\xi}_i^* = 0$. With the similar proof, if $\alpha_i^* = C$, the theorem will also hold. ■

According to the proof of theorem 1, the KKT conditions of SVR can be summarized as follows:

$$\alpha_i^{(*)} = 0 \quad \Rightarrow \quad |y_i - f(x_i)| \leq \varepsilon \tag{13}$$

$$\alpha_i^{(*)} \in (0, C) \quad \Rightarrow \quad |y_i - f(x_i)| = \varepsilon \tag{14}$$

$$\alpha_i^{(*)} = C \quad \Rightarrow \quad |y_i - f(x_i)| \geq \varepsilon \tag{15}$$

where the samples (x_i, y_i) of (13), (14) and (15) are called NoSV, Normal Support Vector (NSV) and Boundary Support Vector (BSV) respectively, where 1) NoSVs locate at the inside or boundary of the ε-bound, 2) NSVs locate on the boundary and 3) BSVs locate at the outside or boundary respectively [3]. The NSVs and BSVs are called by a joint name, i.e. Support Vector (SV)[1].

From the analysis above, the following corollary holds:

Corollary 1: Given $\bar{\alpha}^{(*)} = (\bar{\alpha}_1^{(*)}, ..., \bar{\alpha}_N^{(*)})^T$ be the optimal solution of programming (5), $f(x)$ be the regression function, for any $i \in \{1, \cdots, N\}$:

1) if $|y_i - f(x_i)| < \varepsilon$, then $\bar{\alpha}_i^{(*)} = 0$;
2) if $|y_i - f(x_i)| > \varepsilon$, then $\bar{\alpha}_i^{(*)} = C$

Proof: For 1), if $|y_i - f(x_i)| < \varepsilon$, then

$$\begin{aligned} 0 &¡ \bar{\xi}_i ¡ \varepsilon + \bar{\xi}_i + y_i - f(x_i) ¡ 2\varepsilon + \bar{\xi}_i \\ 0 &¡ \bar{\xi}_i^* ¡ \varepsilon + \bar{\xi}_i^* - y_i + f(x_i) ¡ 2\varepsilon + \bar{\xi}_i^* \end{aligned} \tag{16}$$

According to (9) and (12) we have $\bar{\alpha}_i = 0, \bar{\alpha}_i^* = 0$;

For 2), utilizing reduction to absurdity, suppose $|y_i - f(x_i)| > \varepsilon$, but the corresponding $\bar{\alpha}_i^{(*)} \in [0, C)$. According to the proof of theorem 1, if $\bar{\alpha}_i^{(*)} \in [0, C)$, then $|y_i - f(x_i)| \leq \varepsilon$. it's obvious that such identification is at odds with the assumption, that is $\bar{\alpha}_i^{(*)} = C$, which completes the proof. ■

Note that, without consideration of computing b and with assumption of $\beta = \alpha - \alpha^*$ (without loss of generality, suppose $\beta \geq 0$), the primal dual problem (5) can be transformed into the following expression according to the result of theorem 1, i.e. $\alpha_i \alpha_i^* = 0$:

$$\begin{aligned} \min F = &\tfrac{1}{2}\beta^T \mathcal{K}(X^T, X^T)\beta + \varepsilon e^T |\beta| - Y^T \beta \\ &s.t. \quad e^T \beta = 0 \end{aligned} \tag{17}$$

The expression of (17) is quite similar to the quadratic programming for classification, while employing $\hat{\beta}_i = y_i \beta_i$ and supposing $\varepsilon = 0$ will make the similarity more obvious. The only difference between them is without limitation of $\hat{\beta}_i > 0$, while $\alpha_i > 0$ was required for classification. Therefore, the first derivative of F to β_i is as follows:

$$\begin{aligned} h_i &= \frac{\partial F}{\partial \beta_i} = \sum_{j=1}^{N} K(x_i, x_j)\beta_j + \varepsilon - y_i \\ &= f(x_i) + \varepsilon - y_i \begin{cases} \geq 0 \ ; \ |y_i - f(x_i)| \leq \varepsilon \\ = 0 \ ; \ |y_i - f(x_i)| = \varepsilon \\ \geq 0 \ ; \ |y_i - f(x_i)| \geq \varepsilon \end{cases} \end{aligned} \tag{18}$$

It's easy to conclude the relations between the three types of samples (that is NoSV, NSV and BSV) and the first derivative of F as well as the geometrical properties of different samples (see figure 1 and table I) according to (13)∽ (15).

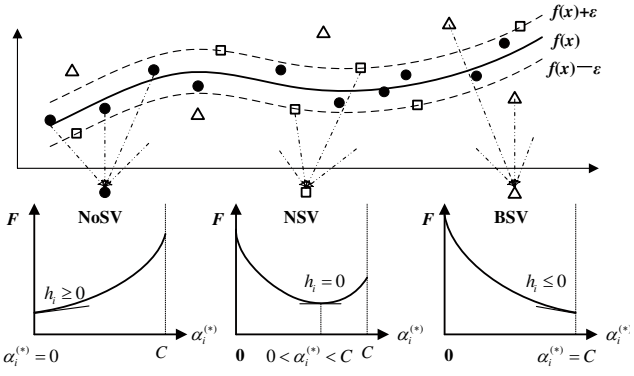

Fig. 1. Relations between three types of samples and first derivative of objective function

TABLE I
GEOMETRICAL PROPERTIES OF DIFFERENT TYPES OF SAMPLES

Type Name		$\bar{\alpha}_i^{(*)}$	$\bar{\xi}_i^{(*)}$	$s_i^{(*)a}$
NoSV		0	0	0
Normal SV	in-bound SV	$(0, C)$	0	0
Boundary SV	off-bound SV	C	$(0, \infty)$	$(0, \infty)$

a: $s_i^{(*)}$ denotes the corresponding expression of complementary slackness conditions of Lagrange multipliers $\alpha_i^{(*)}, \xi_i^{(*)}$, that is $s_i := \alpha_i(\varepsilon + \xi_i + y_i - f(x_i))$ or $s_i^* := \alpha_i^*(\varepsilon + \xi_i^* - y_i + f(x_i))$.

It's noted that the optimal solution may be more than 1 since the SV kernel just requires the Gram matrix $\mathcal{K} = \mathcal{K}(X^T, X^T) = (K(x_i, x_j))_{N \times N}$ be semidefinite. According to lemma 1, if and only if $\alpha^{(*)}$ make any training sample x meet the KKT conditions, $\alpha^{(*)}$ are the optimal solutions of

[1]the SVs corresponding to (14) and (15) are also called on-boundary SV and off-boundary SV or utilizing terms "in-bound" and "bound support vector" respectively.

programming (2). In other words, if there is a modified sample s, it's required that s is a NoSV to ensure that $\alpha^{(*)}$ are also the optimal Lagrange multipliers trained for the new training set, that is s obeys the primal KKT conditions. Therefore, the following theorem holds:

Theorem 2: Given primal training set $\mathcal{N}_0 = (X_0, Y_0)$ and suppose the training result be as follows:

$$f_0(x) = \mathcal{K}(x, X_0^T)(\overline{\alpha}_0 - \overline{\alpha}_0^*) + \bar{b} \tag{19}$$

and let $g(x)$ be the follows judgement function:

$$g(x) = y - f_0(x) \tag{20}$$

Suppose $s = (x_s^T, y_s)$ be a modified sample, where $x_s = (x_s^1, \ldots, x_s^d)^T$ is input, y_s is the corresponding output, then the following assertions hold, that is

1) if $|g(x_s)| = |y_s - f_0(x_s)| < \varepsilon$, then s obeys the primal KKT conditions and the regression function dose not change as well as the s is a NoSV.
2) if $|g(x_s)| = |y_s - f_0(x_s)| > \varepsilon$, then s disobeys the primal KKT conditions and the regression function definitely change while the s will change to be a SV.

Proof: Firstly, we proof the case that the modified sample s is appended. Without loss of generality, Let $\mathcal{N} = (\mathcal{N}_0^T, s)^T = \begin{pmatrix} X_0 & Y_0 \\ x_s^T & y_s^T \end{pmatrix}_{(N+1)\times(N+1)} = (X, Y), X = \begin{pmatrix} X_0 \\ x_s^T \end{pmatrix}, Y = \begin{pmatrix} Y_0 \\ y_s^T \end{pmatrix}$, $\alpha^{(*)} = (\alpha_0^{(*)T}, \alpha_s^{(*)}) \in \mathbb{R}^{N+1}$, where $\alpha_0^{(*)T} \in \mathbb{R}^N$ denote the first N Lagrange multipliers, whose corresponding training set is $\mathcal{N}_0$. Suppose $\alpha_s^{(*)}$ be a new Lagrange multiplier corresponding to s, and

$$\mathcal{K}(X^T, X^T) = \begin{pmatrix} \mathcal{K}(X_0^T, X_0^T) & \mathcal{K}(X_0^T, x_s) \\ \mathcal{K}(x_s, X_0^T) & \mathcal{K}(x_s, x_s) \end{pmatrix}_{(N+1)\times(N+1)}$$

Then the quadratic programming is as follows:

$$\begin{aligned} \min F = & \tfrac{1}{2}(\alpha - \alpha^*)^T \mathcal{K}(X^T, X^T)(\alpha - \alpha^*) \\ & + \varepsilon(e^T, 1)(\alpha + \alpha^*) - Y^T(\alpha - \alpha^*) \\ s.t. & \begin{cases} (e^T, 1)(\alpha - \alpha^*) = 0 \\ C(e^T, 1)^T - \alpha \geq 0 \\ C(e^T, 1)^T - \alpha^* \geq 0 \\ \alpha \geq 0, \alpha^* \geq 0 \end{cases} \end{aligned} \tag{21}$$

Actually, the objective function of the programming (21) can be transformed to the following function,

$$\begin{aligned} F &= \frac{1}{2}(\alpha_0 - \alpha_0^*)^T \mathcal{K}(X_0^T, X_0^T)(\alpha_0 - \alpha_0^*) \\ &+ \varepsilon e^T(\alpha_0 + \alpha_0^*) - Y_0^T(\alpha_0 - \alpha_0^*) \\ &+ \frac{1}{2}(\alpha_s - \alpha_s^*)^T \mathcal{K}(x_s, X_0^T)(\alpha_0 - \alpha_0^*) \\ &+ \frac{1}{2}(\alpha_0 - \alpha_0^*)^T \mathcal{K}(X_0^T, x_s)(\alpha_s - \alpha_s^*) \\ &+ \frac{1}{2}(\alpha_s - \alpha_s^*)^T \mathcal{K}(x_s, x_s)(\alpha_s - \alpha_s^*) \\ &+ \varepsilon(\alpha_s + \alpha_s^*) - y_s(\alpha_s - \alpha_s^*) \end{aligned} \tag{22}$$

and the corresponding constraints are:

$$\begin{aligned} e^T(\alpha_0 - \alpha_0^*) + (\alpha_s - \alpha_s^*) = 0,\ \alpha_0 \geq 0, \alpha_0^* \geq 0 \\ Ce - \alpha_0 \geq 0, C - \alpha_s \geq 0,\ \alpha_s \geq 0, \alpha_s^* \geq 0 \\ Ce - \alpha_0^* \geq 0, \quad C - \alpha_s^* \geq 0 \end{aligned} \tag{23}$$

Suppose the optimal solutions of programming (21) are $\hat{\alpha}^{(*)} = (\hat{\alpha}_1^{(*)}, \cdots, \hat{\alpha}_N^{(*)}, \hat{\alpha}_s^{(*)})$. According to the KKT conditions, there are some Lagrange multipliers $\hat{b}, \hat{\xi}^{(*)}$ such that the following complementary slackness conditions hold:

$$\hat{\alpha}_i(\varepsilon + \hat{\xi}_i - g(x_i)) = 0 \tag{24}$$

$$\hat{\alpha}_i^*(\varepsilon + \hat{\xi}_i^* + g(x_i)) = 0 \tag{25}$$

Thus, for 1), if $|g(x_s)| = |y_s - f_0(x_s)| < \varepsilon$, i.e. $\varepsilon + \hat{\xi}_s - g(x_s) > 0$, then $\hat{\alpha}_s = 0$. In the same way, $\hat{\alpha}_s^* = 0$ holds. Substituting it into (22) and (23), such that the terms, which contain $\alpha_s \pm \alpha_s^*$, equal to 0. Therefore, the programming (21) is equivalent to programming (5). Thus, the optimal solutions $\hat{\alpha}^{(*)} = (\bar{\alpha}^{(*)T}, 0)^T$, where $\bar{\alpha}^{(*)}$ are the optimal solutions of programming (5). In other words, if $|g(x_s)| < \varepsilon$, sample $s = (x_s, y_s)$ obeys the KKT conditions of primal programming (5), thus the regression function does not change and s is a NoSV.

For 2), firstly suppose s obeys the KKT conditions of primal programming (5), then s is appended to obtain a new training set, denoted by $\mathcal{N}$ for solving the programming. The corresponding optimal solution of s should be $\alpha_s^{(*)} = 0$. According to the theorem 1 and corollary 1, then $|y_s - f_0(x_s)| \leq \varepsilon$. It's noted that the result is inconsistent with the assumption $|y_s - f_0(x_s)| > \varepsilon$. Thus, s disobeys the KKT conditions of primal programming (5).

Secondly, suppose the regression function does not be effected by the modified sample s, though s disobeys the KKT conditions of primal programming (5). In other words, s is a NoSV for new programming (21). Thus, the corresponding Lagrange multiplier $\alpha_s^{(*)} = 0$. According to the proof of 1), if $\alpha_s^{(*)} = 0$, then programming (21) is equivalent to primal programming (5) and consequently the regression function trained as well as KKT conditions would not change. However, s will disobey the KKT conditions of programming (21) consequentially since it disobeys the KKT conditions of programming (5). It's inconsistent with the lemma 1. Therefore, the regression function will change and s is a SV.

If the modified sample is removed, the proof is in the same way as above, which would not be discussed due to the limited space. ■

From the proof above, the following corollary holds,

Corollary 2: Any modified sample x obeys the KKT conditions of primal programming if and only if x meets $|g(x)| < \varepsilon$.

The training results, which are trained with all the samples and just with a set of samples which disobey the KKT conditions, are same by utilizing the theorem 2 and corollary 2. Therefore, for any modified sample s:

1) If s obeys the primal KKT conditions, it means that the SV set contains the information of s and it's unnecessary to train it any more.
2) If s disobeys the primal KKT conditions, it means that there is a lack of information of s and the training with s is necessary.

Obviously, the prior evaluation, which modified samples may disobey the KKT conditions, will simplify the training with modified samples. Based on the idea, lots of quick training algorithms (QTA), e.g. Chunking [1], Osuna [10] and SMO [11] etc., are proposed. However, there is a notable disadvantage of these QTAs, i.e. they are lack of convergence [12]. Because the theorem 2 can just ensure that the modified samples will possess no impact on the training result if they obey the primal KKT conditions. Whereas, there exists some cases that the modified samples would turn to be SVs when they disobey the KKT conditions, while it may make the original NoSVs turn to be SVs and the converse may be also true. Therefore, this case will make the QTAs iterate all the time.

As noted in previous section, an accurate solution procedure (ASP) for incremental learning was proposed by Cauwenberghs *et al.* [13]. The algorithm aims to compute the accurate impact of any modified sample on Lagrange coefficients and SVs, which was introduced into regression analysis by Ma *et al.* [14]. However, the main shortcoming is its inefficiency.

Therefore, this paper will propose an approximation incremental training algorithm (AITA) based on the ASP. It can be considered to be a tradeoff of the QTA and ASP as follows:

- Take the nonconvergence of QTA into account. AITA will consider the cases on which the existing samples would be turned to be a SV or NoSV due to the modified samples, which is the idea of ASP.
- Overcome the inefficiency of ASP. AITA allows a spot of samples could disobey the primal KKT conditions as long as the loss of precision is tolerant as QTA does [12]. It can dramatically improve the efficiency of training by only computing the accurate change of samples which are found most likely to change.

B. Approximation Incremental Training Algorithm

1) Overview of Basic Process: The basic process of AITA is as follows:

Setp 1: Firstly verify the modified sample $s_c = (x_s, y_s)$ (suppose s denote its index) whether it meets the primal KKT conditions according to the theorem 2. It's unnecessary to append it into the new training set if it obeys the KKT conditons, otherwise turn to Setp 2.

Setp 2: Let A denote the indexes of the primal training set. The training set is divided into the following three parts, i.e.

1) index set of NoSV, that is

$$NoSV = \{i \mid \text{if sample}(x_i^T, y_i)\text{is NoSV}\} \tag{26}$$

2) index set of NSV, that is

$$NSV = \{i \mid \text{if sample } (x_i^T, y_i)\text{is NSV}\} \tag{27}$$

3) index set of BSV, that is

$$BSV = \{i \mid \text{if sample } (x_i^T, y_i)\text{is BSV}\} \tag{28}$$

Obtain all the samples within the radius $\Delta\delta = |y_s - f(x_s)| + \varepsilon$ of the s_c and denote the index set by I as well as let its potency be n_i.

Setp 3: Check the change of the corresponding samples with index in NSV caused by s_c and apply the change to all the samples with index in $C = I \cup NSV$. Determine whether the samples with index in C would turn to SV or NoSV and keep the samples with index in $A - C$ unchanged.

2) the Realization of AITA: Given the primal training set be $\mathcal{N} = \{(x_i^T, y_i)\}_{i=1}^N$, $s_c = (x_s^T, y_s)$ denote a modified sample and $|g(x_s)| = |y_s - f(x_s)| > \varepsilon$. Firstly suppose $NoSV, NSV, BSV$ be the indexes sets obtained by training with $\mathcal{N}$, and let theirs potencies be n_p, n_q, n_l. The basic idea of AITA is to modify the corresponding Lagrange multiplier β_s of modified sample s_c with an increment $\Delta\beta_s$, and to make sure that s_c will obey the KKT conditions in a limited iteration and all the samples with the indexes in $S = NSV \cup \{s\}$ always obey the KKT conditions in each iteration. Therefore, the relationship between $\Delta\beta_s$ and the change of KKT conditions are discussed firstly.

At first, let $\beta_s = 0$, then change (increase or decrease) the β_s gradually with $\Delta\beta_s$ under the constraint of KKT conditions (see (13)$\sim$(18)). In each incremental step, the corresponding increment of samples with indexes in $D = I \cup NSV$, i.e. $\Delta\beta_i, (i \in S)$, should ensure the samples meet the KKT conditions. In order to make sure that s_c also meet the KKT conditions, it's necessary that for $\forall i \in S$ such that

$$\Delta h_i = K(x_i, x_s)\Delta\beta_s + \sum_{j \in NSV} K(x_i, x_j)\Delta\beta_j + \Delta b \tag{29}$$

$$0 = \Delta\beta_s + \sum_{j \in NSV} \Delta\beta_j \tag{30}$$

where Δb denotes the increment of bias. According to the KKT conditions, for all the sample with indexes $i \in NSV$, the corresponding $h_i \equiv 0$. Thus, the equations (29) and (30) can be rewritten as follows:

$$\begin{aligned} \sum_{j \in NSV} K(x_i, x_j)\Delta\beta_j + \Delta b &= -K(x_i, x_s)\Delta\beta_s \\ \textstyle\sum_{j \in NSV} \Delta\beta_j &= -\Delta\beta_s \end{aligned} \tag{31}$$

without loss of generality , suppose $NSV = \{s_1, \cdots, s_{n_p}\}$, the corresponding sample set be X_{NSV}, then equation(31) can be transformed into the following matrix form:

$$\begin{aligned} \begin{pmatrix} 0 & e^T \\ e & \mathcal{K}(X_{NSV}^T, X_{NSV}^T) \end{pmatrix} \cdot \begin{pmatrix} \Delta b \\ \Delta\beta_{NSV} \end{pmatrix} \\ = -\begin{pmatrix} 1 \\ \mathcal{K}(X_{NSV}^T, x_s) \end{pmatrix} \Delta\beta_s \end{aligned} \tag{32}$$

where $\Delta\beta_{NSV} = (\Delta\beta_{s_1}, \cdots, \Delta\beta_{s_{n_p}})^T$. Since $\mathcal{K}(X_{NSV}^T, X_{NSV}^T)$ is semi-definite, let matrix $\mathcal{R}$ be as follows:

$$\mathcal{R} = \begin{pmatrix} 0 & e^T \\ e & \mathcal{K}(X_{NSV}^T, X_{NSV}^T) \end{pmatrix}^{-1} \tag{33}$$

and substituting into (32), i.e.

$$\begin{aligned} \begin{pmatrix} \Delta b \\ \Delta\beta_{NSV} \end{pmatrix} &= -\mathcal{R} \cdot \begin{pmatrix} 1 \\ \mathcal{K}(X_{NSV}^T, x_s) \end{pmatrix} \cdot \Delta\beta_s \\ &= \Gamma \cdot \Delta\beta_s = \begin{pmatrix} \gamma \\ \gamma_{NSV} \end{pmatrix} \cdot \Delta\beta_s \end{aligned} \tag{34}$$

where

$$\Gamma = \begin{pmatrix} \gamma \\ \gamma_{NSV} \end{pmatrix} = -\mathcal{R} \cdot \begin{pmatrix} 1 \\ \mathcal{K}(X_{NSV}^T, x_s) \end{pmatrix} \quad (35)$$

denotes the Coefficient Sensitivity of the change of multipliers [13], where $\gamma_{NSV} = (\gamma_1, \cdots \gamma_{n_p})^T$. It's obvious that $\mathcal{R}$ above will change as the NSV changes, and the update strategy will be discussed later. In ASP, the corresponding balance condition of Coefficient Sensitivity is as follows:

$$\Delta b = \gamma \Delta\beta_s \quad (36)$$

$$\Delta\beta_j = \gamma_j \Delta\beta_s, \quad \forall j \in A \quad (37)$$

and let $\gamma_j \equiv 0, (\forall j \notin NSV)$. In AITA, the index set of sample which will be checked is reduced to I. Note that the samples with indexes in I meet the balance condition above. Therefore suppose $I' = I - I \cap NSV = \{i \mid i \in I, i \notin NSV\}$, i.e. the indexes in I rather than in NSV, and suppose n_t denote its potency and $I' = \{t_1, \cdots, t_{n_t}\}$ be the index set and $X_{I'}$ be the corresponding sample set. According to (18),(29),(30) and (35), then

$$\begin{pmatrix} \Delta h_{t_1} \\ \vdots \\ \Delta h_{t_{n_t}} \end{pmatrix} = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_{t_{n_t}} \end{pmatrix} \Delta\beta_s = \Phi \Delta\beta_s \quad (38)$$

where

$$\begin{aligned} \Phi &= \begin{pmatrix} K(x_{t_1}, x_s) \\ \vdots \\ K(x_{t_{n_t}}, x_s) \end{pmatrix} \quad (39) \\ &+ \begin{pmatrix} 1 & K(x_{t_1}, x_{s_1}) & \cdots & K(x_{t_1}, x_{s_{n_p}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_{t_{n_t}}, x_{s_1}) & \cdots & K(x_{t_{n_t}}, x_{s_{n_p}}) \end{pmatrix} \\ &\cdot \begin{pmatrix} \gamma \\ \gamma_{NSV} \end{pmatrix} = \mathcal{K}(X_{I'}, x_s) + (e, \mathcal{K}(X_{I'}, X_{NSV})) \cdot \Gamma \end{aligned}$$

is called Margin Sensitivity [13]. It can be transformed to the component form as follows:

$$\phi_i = \begin{cases} K(x_i, x_s) + \sum_{j \in NSV} K(x_i, x_j)\gamma_j + \gamma, & \forall i \in I' \\ 0, & \forall i \in NSV \end{cases} \quad (40)$$

Especially, if $NSV = \emptyset$, i.e. there are no NSV, equation (38) can be simplified to $\Delta h_j = \Delta b, \forall j \in I'$ according to (29) and (30). Given any $\Delta\beta_s$, the $\beta_i (i \in NSV)$ and b can be updated according to (35), and $h_i (i \in I')$ can also be updated according to (35). It' noted that the samples with indexes in I' and NSV would not change if $\Delta\beta_s$ is as small as possible, while Δh_i and consequently the samples would change if $|\Delta\beta_s|$ is gradually increasing since equation (38) holds. According to the derivative process of (34) and (38), they will hold only if the NSV remains unchanged. Therefore, the next step is to determine the bound of $\Delta\beta_s$ to keep the NSV unchanged or determine the termination condition.

Firstly determine the sign of $\Delta\beta_s$, i.e. whether β_s be increased or decreased. It's found that $\beta_i = \alpha_i - \alpha_i^*$ and $\alpha_i \alpha_i^* = 0$, thus (13)-(15) can be transformed to the following equation:

$$\beta_i \begin{cases} = -C & ; y_i - f(x_i) \le -\varepsilon \\ \in (-C, 0) & ; y_i - f(x_i) = -\varepsilon \\ = 0 & ; -\varepsilon \le y_i - f(x_i) \le \varepsilon \\ \in (0, C) & ; y_i - f(x_i) = \varepsilon \\ = C & ; y_i - f(x_i) \ge \varepsilon \end{cases} \quad (41)$$

Therefore, according to (41), the sign of $\Delta\beta_s$ can be determined as follows:

$$q = sign(\Delta\beta_s) = -sign(y_s - f(x_s)) \quad (42)$$

Next is to determine the magnitude of $\Delta\beta_s$. Using the taxonomy in [13] for reference, it's necessary to estimate the following five cases for the possible four kind of samples with corresponding indexes in I (that is modified sample, NoSV, NSV, BSV) to determine the magnitude of $\Delta\beta_s$ according to (34) and (38).

1) for a modified sample s_c:
   - **C1**: $h_s \le 0$ and h_s is turned from $h_s < 0$ to $h_s = 0$, then s_c is turned to be a NSV.
   - **C2**: $|\beta_s| \le C$, and β_s is turned from $|\beta_s| < C$ to $|\beta_s| = C$, then $s_c$is turned to be a BSV.
2) for $s_i = (x_i^T, y_i), i \in NSV$:
   - **C3**: $0 \le |\beta_i| \le C$, if β_i is turned from $0 < |\beta_i| < C$ to $|\beta_i| = C$, then s_i is turned from a NSV to a BSV, and if $\beta_i = 0$, then s_i is turned from a NSV to a NoSV.
3) for $s_i = (x_i^T, y_i), i \in BSV$:
   - **C4**: $h_i \le 0$, and h_i is turned from $h_i \le 0$ to $h_i = 0$, then s_i is turned from a BSV to a NSV.
4) for $s_i = (x_i^T, y_i), i \in NoSV$:
   - **C5**: $h_i \ge 0$, and h_i is turned from $h_i \ge 0$ to $h_i = 0$, then s_i is turned from a NoSV to a NSV.

These cases will be discussed summarily:

a) for case **C1**: According to (38) and (40), $\Delta\beta_s = \frac{\Delta h_s}{\phi_s}$, thus the maximum increment of $\Delta\beta_s$is as follows:

$$C_1 = y_s - f(x_s) + q\varepsilon/\phi_s \quad (43)$$

b) for **C2**: according to β_s, the maximum increment of $\Delta\beta_s$ is as following equation since β_s is turned from $\beta_s < C$ to $\beta_s = C$:

$$C_2 = qC - \beta_s \quad (44)$$

c) for **C3**: according to (37), the maximum increment of $\Delta\beta_s$ will be discussed with the following cases for the different values of β_i and γ_i:

$$C_3 = \begin{cases} \frac{(C-\beta_i)}{\gamma_i}, & \text{if } 0 \le \beta_i < C \ , q\gamma_i > 0, \beta_i \text{turns to be } C \\ \frac{-(C+\beta_i)}{\gamma_i}, & \text{if } -C < \beta_i \le 0, q\gamma_i < 0, \beta_i \text{turns to be } -C \\ \frac{-\beta_i}{\gamma_i}, & \text{if } -C \le \beta_i < 0, q\gamma_i > 0, \beta_i \text{turns to be } 0 \\ \frac{-\beta_i}{\gamma_i}, & \text{if } 0 < \beta_i \le C \ , q\gamma_i < 0, \beta_i \text{turns to be } 0 \end{cases} \quad (45)$$

d) for **C4**: since (37), the maximum increment of $\Delta\beta_s$ is

$$C_4 = y_s - f(x_s) - sign(q\gamma_i)\varepsilon/\gamma_s \tag{46}$$

e) for **C5**: similarly to **C4**, the maximum increment of $\Delta\beta_s$ is

$$C_5 = y_s - f(x_s) + sign(q\gamma_i)\varepsilon/\gamma_s \tag{47}$$

Note that the $\Delta\beta_s$ computed above may be different from each other. Therefore, the upper bound can be determined as the minimum of all the $\Delta\beta_s$ computed when $\Delta\beta_s > 0$. In the same way, for $\Delta\beta_s < 0$, the lower bound is the corresponding maximum value. Moreover, it can be simplified to the following equation:

$$\Delta\beta_s^{\min} = q\min\{|C_1|, |C_2|, |C_3|, |C_4|, |C_5|\} \tag{48}$$

In the iteration of AITA, the $\mathcal{R}$ (see (33)) should be updated as long as NSV changes. Similarly with reference [13], $\mathcal{R}$ should be expanded to the following equation to append the index of the modified sample s_c, which is possible to turn to be a NSV, to NSV:

$$\mathcal{R} \leftarrow \begin{pmatrix} \mathcal{R} & 0e \\ 0e^T & 0 \end{pmatrix} + \frac{1}{\phi_s} \cdot \begin{pmatrix} \Gamma \\ 1 \end{pmatrix} \cdot \begin{pmatrix} \Gamma, & 1 \end{pmatrix} \tag{49}$$

where Γ, ϕ_i can be computed according to (35) and (40). It also can be adapted to append the index of any sample into NSV, meanwhile, $\mathcal{R}$ is still reversible [13]. If the indexes of some samples in index set NSV, just suppose the kth index will be removed from NSV and the corresponding sample is s_k, then the updating equation of $\mathcal{R}$ is as follows:

$$\begin{gathered} r_{ij} \leftarrow r_{ij} - r_{kk}^{-1} r_{ik} r_{kj} \\ \forall i, j \in N' = \{0, 1, \cdots, k-1, k+1, \cdots, n_p\} \end{gathered} \tag{50}$$

where index 0 corresponds to b. According to the analysis above, the AITA can be presented as follows:

Step 1: $\beta_s \leftarrow 0$, compute h_s. If $h_s > 0$, it means that s_c obeys the KKT conditions, thus append its index into $NoSV$, and terminate AITA, otherwise turn to Step 2.

Step 2: obtain sample indexes within $|g(x_s)| + \varepsilon$, i.e. I.

Step 3: Compute the sign q of $\Delta\beta_s$ according to (42).

Step 4: If s_c meets the KKT conditions, i.e. $\Delta\beta_s$ is determined by **C1** or **C2**, terminate AITA, otherwise turn to Step 5.

Step 5: Update Γ, Φ according to (35) and (39) respectively, compute the increments of $\Delta\beta_s$ corresponding to various cases according to (43)-(47), or compute its bound according to (48). Update $\beta_s, b, \beta_i, i \in NSV$ according to (34) and $h_i, i \in I'$ according to (38). Update various index set under the following cases according to the bound of $\Delta\beta_s$:

1) if the bound was computed by **C1**, then append the index of s_c into NSV and update $\mathcal{R}$ according to (49).
2) if the bound was computed by **C2**, then append the index of s_c into BSV
3) if the bound was computed by **C3**, and
   - if the corresponding $\beta_k = 0$ for the current sample s_k, then append the index k of s_k into $NoSV$ and update $\mathcal{R}$ according to (50).
   - if the corresponding $\beta_k = C$ for the current sample s_k, then append the index k of s_k into BSV and update $\mathcal{R}$ according to (50).
4) if the bound was computed by **C4**, then append the index k of s_k into NSV and update $\mathcal{R}$ according to (49).
5) if the bound was computed by **C5**, then append the index k of s_k into NSV and update $\mathcal{R}$ according to (49).

III. Experiment and Result Analysis

A. Synthetic Problem

In order to demonstrate the existence of the transformations between SV and NoSV and the impact of the location of modified samples on these transformations, a first-order function $y = f(x) = (1 + x + x^2)e^{0.5x^2}$ was selected, where a data set of $N = 20$ training points in which the input data point x is picked uniformly from the interval $x \in [-4, 4]$, and the targets are generated by an additive white noise (the true values and observations are denoted by solid squares and $+$ respectively as shown in fig.2(a)). Meanwhile, five modified samples, which were labeled with Loc. i ($i = 1, \cdots, 5$) from left to right (denoted by hollow circle as shown in fig.2(a)), were selected according to the geometrical characters of $f(x)$. The corresponding characters are as follows:

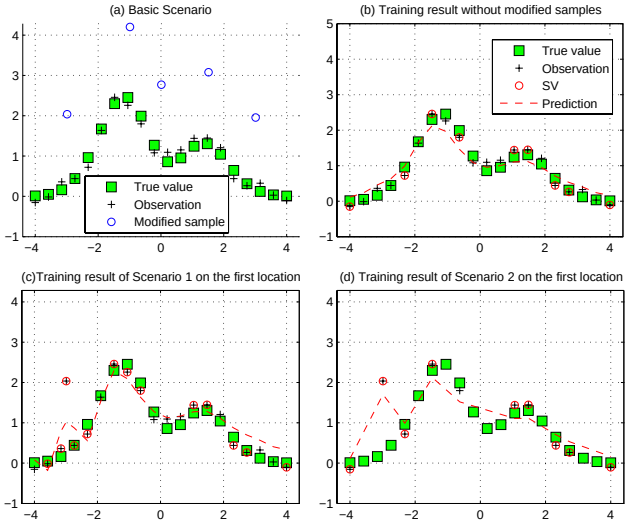

Fig. 2. Training results of basic scenario and two test scenarios on Loc. 1

1) **Loc. 1**: the derivative of $f(x)$ at Loc. 1 is larger than 0 and it locates at an area which is smooth and changes rapidly on both sides.
2) **Loc. 2**: the derivative of $f(x)$ at Loc. 2 approximates to 0 and it locates at an area which contains the local maximums and changes rapidly on both sides.
3) **Loc. 3**: the derivative of $f(x)$ at Loc. 3 approximates to 0 and it locates at an area which contains the local minimums and changes differently on both sides.
4) **Loc. 4**: the derivative of $f(x)$ at Loc. 4 approximates to 0 and it locates at an area which contains the local maximums and changes smoothly on both sides.

5) **Loc. 5**: the derivative of $f(x)$ at Loc. 5 is smaller than 0 and it locates at an area which is smooth and changes smoothly on both sides.

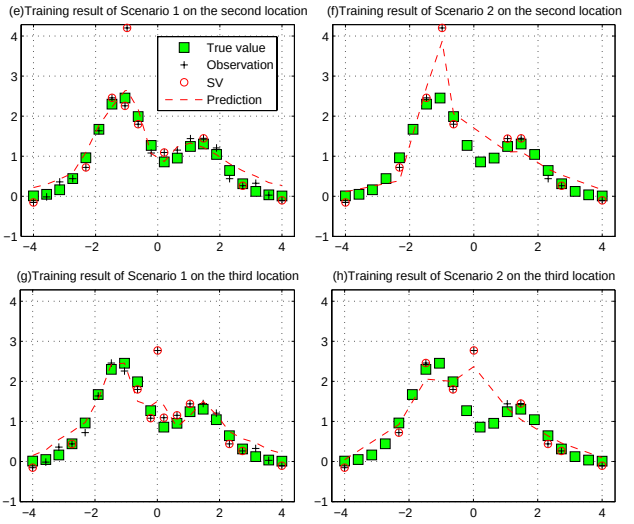

Fig. 3. Training results of two test scenarios on Loc. 2 and 3

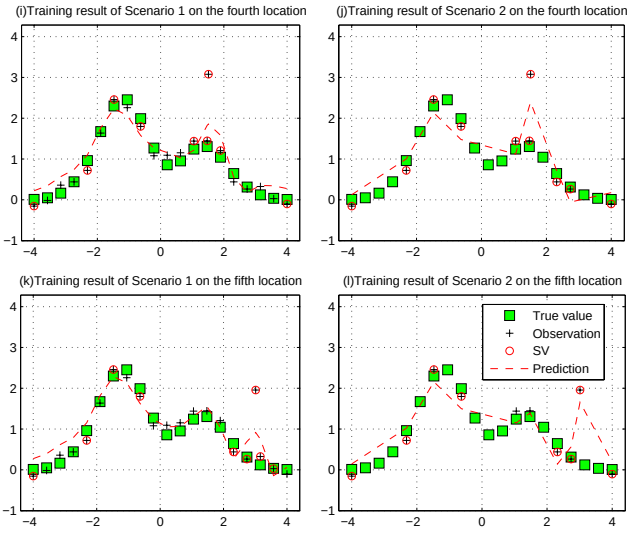

Fig. 4. Training results of two test scenarios on Loc. 4 and 5

B. Test scenario and simulation result

Meanwhile, three different scenarios as follows are designed for systematic comparisons on each location.

1) **Basic Scenario** Neglect modified samples, where $\oplus$ denotes the SVs trained (whose training set is S) and dashed denotes the prediction curve (as shown in fig.2(b))
2) **Test Scenario 1** Train the SVR with all the training set that the modified samples are appended to the primal training set.
3) **Test Scenario 2** Train the SVR with a new training set which consists of the modified samples and S.

It's noted that the following conclusions can obtained from fig.2 to fig.4.

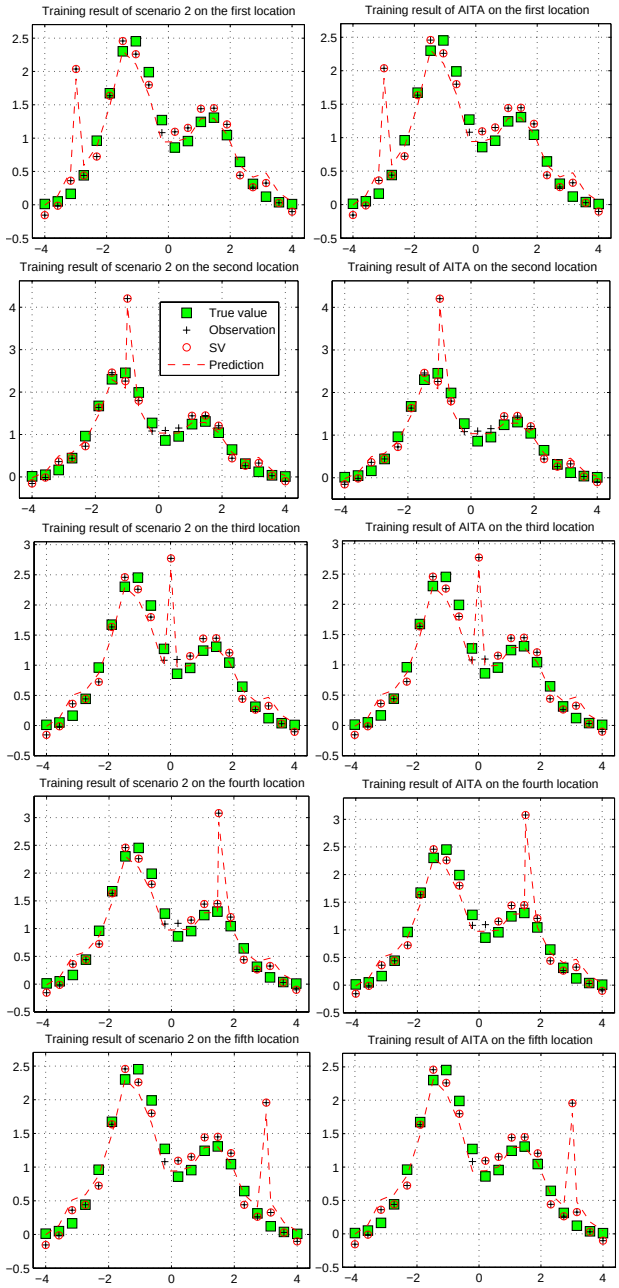

Fig. 5. Comparison results between AITA and the test scenario 2 on the five locations

1) It's found that the offset of the curve approximates 0 as the distance, which is apart from the modified sample, increasing according to the comparison of the two test scenarios with the basic scenario.
2) It's noted that the impact of the modified sample on the change of SV beside the modified sample according to the comparison of the test scenario 1 with the basic scenario.

3) It's found that the more SVs near the modified sample, the more extent to which the curve changes according to the comparison of the test scenario 1 with scenario 2.

In order to illustrate the advantage of AITA, we still take the test function used in fig.2 to fig.4 as an example. Fig.5 shows the comparison results between the test scenario 2 and AITA on the five locations above. For the precision, AITA is superior to that of the test scenario 2 from fig.5, especially on the rougher locations, i.e. Loc. 2, 3 and 4. Moreover, it's found that the amounts of SV trained by AITA and the test scenario 1 are approximately equivalent according to the comparison between fig.5 and fig.2∼4, while the training samples are less than that of the test scenario 1.

Next, the following two comparison schemes are set to illustrate the training efficiency of the four scenarios, i.e. basic scenario, test scenario 1, test scenario 2 and AITA.

1) **Efficiency Scheme 1** It's different about the potency of training set (there are twelve schemes from 5 to 300 respectively) to evaluate the change of average training times when there is an appended modified sample (the additive noise is set bigger to make sure the sample will be appended into training set) in each five locations.
2) **Efficiency Scheme 2** The primal training set keeps unchanged (150 samples) and evaluate the impact of different kind of modified sample (which may be appended into or keep away from the training set) with different amounts (there are eight schemes from 5 to 100 respectively) on the total training time.

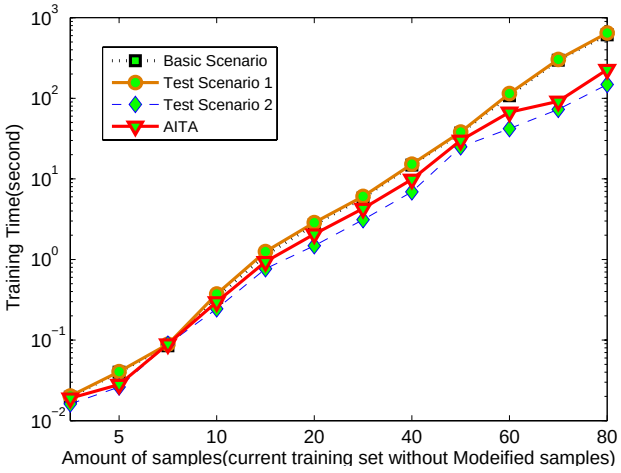

Fig. 6. Comparison results under the Efficiency Scheme 1

Fig.6 and 7 show that the comparison results under the efficiency schemes, that is

1) It's found that the efficiency of AITA is superior to that of the test scenario 2 while sightly inferior to that of the test scenario 1 owing to the corresponding potencies of the training sets when the amount of modified samples is less according to the fig.6 and 7. However, fig.5 shows that the test scenario 2 is inferior to AITA in the performance of precision.

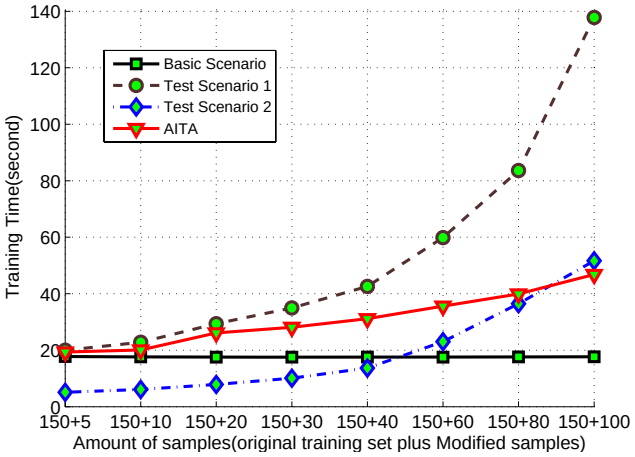

Fig. 7. Comparison results under the Efficiency Scheme 2

2) It's noted that the training time is increasing with the amount of the modified samples, while the increasing rate is different, i.e. that of the test scenario 1 is more than others according to fig.7 when the amount of modified samples are more and of complicated composition. Moreover, the training time of AITA increases slower than the test scenario 1 and 2 and less than these test scenarios finally with the increment of sample size, because some of these modified samples may meet the KKT conditions and do not attach themselves to the computation of kernel matrix.

According to the analysis of precision and efficiency, AITA can improve the training efficiency while preserve a better precision performance. It's illustrated that it can enhance the training algorithm with an outstanding accumulative learning ability for a changeable training set effectively.

IV. Conclusion

This paper proposed a new algorithm for training SVR quickly under a changeable dataset. It's known that there are significant problems in the conventional quick training algorithms (QTA), e.g. Chunking, SMO *et al.*, and the accurate solution procedure (ASP) for incremental learning. These problems result in the lack of convergence or efficiency. Therefore, this paper is concerned with suggesting ways to improve the training algorithm, i.e. approximation incremental training algorithm (AITA), by hybrid the ideas of QTA and ASP. It learns from the advantages of QTA and ASP, that is improving training efficiency, avoiding nonconvergence. Furthermore, the basic process and realization of AITA are presented. Finally, some comparison schemes are used to demonstrate the correctness of the idea of AITA and illustrate its performance. The numerical results indicate that AITA can indeed improve the quality of SVR in fitting precision and efficiency. It shows great potential for AITA to the accumulative learning ability for a changeable dataset which usually appears in the practical applications.

ACKNOWLEDGMENT

The authors would like to gratefully acknowledge the financial support of the National Defense Pre-Research Foundation of China (Grant No. 9140C640505), the National Natural Science Foundation of China (No. 60974073 and No. 60974074).

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[2] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.
[3] B. Schölkopf, C. J. Burges, and A. J. Smola, *Advances in Kernel Methods - Support Vector Learning*. Cambridge, England: The MIT Press, 1999.
[4] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge: MIT Press, 2002.
[5] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
[6] G. Bloch, F. Lauer, G. Colin, and Y. Chamaillard, "Support vector regression from simulation data and few experimental samples," *Information Sciences*, vol. 178, pp. 3813–3827, 2008.
[7] J.-B. Gao, S. R. Gunn, and C. J. Harris, "Mean field method for the support vector machine regression," *Neurocomputing*, vol. 50, pp. 391–405, 2003.
[8] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," in *Artificial Neural Networks ICANN'97*, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds., vol. 1327. Berlin: Springer Lecture Notes in Computer Science, 1997, pp. 999–1004.
[9] D. Odapally, "Structural optimization using femlab and smooth support vector regression," Ph.D. dissertation, University of Texas, 2006.
[10] E. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130–136.
[11] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods-Support Vector Learning*, B. Sch?lkopf, C. J. Burges, and A. J. Smola, Eds. Cambridge, England: MIT Press, 1999.
[12] W. Zhou, "Kernel-based learning machines," Ph.D. dissertation, Xi'an Electronic and Science University, 2003.
[13] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," *Machine Learning*, vol. 44, no. 13, pp. 409–415, 2001.
[14] J. Ma, J. Theiler, and S. Perkins, "Accurate online support vector regression," *Neural Computation*, vol. 15, no. 11, pp. 2683–2703, 2003.