ARTICLE

Efficient Stochastic Simulation Algorithm for Chemically Reacting Systems Based on Support Vector Regression

Xin-jun Peng[a,b*], Yi-fei Wang[c]

a. Department of Mathematics, Shanghai Normal University, Shanghai 200234, China
b. Scientific Computing Key Laboratory of Shanghai Universities, Shanghai 200234, China
c. Department of Mathematics, Shanghai University, Shanghai 200444, China

(Dated: Received on May 11, 2009; Accepted on July 7, 2009)

The stochastic simulation algorithm (SSA) accurately depicts spatially homogeneous well-stirred chemically reacting systems with small populations of chemical species and properly represents noise, but it is often abandoned when modeling larger systems because of its computational complexity. In this work, a twin support vector regression based stochastic simulations algorithm (TS^{3}A) is proposed by combining the twin support vector regression and SSA, the former is a well-known robust regression method in machine learning. Numerical results indicate that this proposed algorithm can be applied to a wide range of chemically reacting systems and obtain significant improvements on efficiency and accuracy with fewer simulating runs over the existing methods.

Key words: Chemically reacting system, Stochastic simulation algorithm, Machine learning, Support vector regression, Histogram distance

I. INTRODUCTION

Recently, the emerging discipline of system biology attempts to examine the various levels of regulation in an integrated method [1,2], often through using some mathematical models in conjunction with experiments. A most commonly used model is the chemical kinetic model, in which the behavior of a biological system is illustrated as a system of individual chemical reaction channels. A key problem in any chemical kinetic model is the stochastic simulation of chemically reacting systems. Since the molecule numbers of many key reactant species in biological cells are usually few, the discreteness and stochasticity are very important [3].

As an essentially exact numerical simulation method for well-stirred systems, Gillespie's stochastic simulation algorithm (SSA) [4,5] has been widely used in the simulation to find the probability density functions (PDFs) of species of biochemical systems. However, it is time-consuming for most actual problems since it keeps track of each reaction event in a reacting system. Several improved algorithms, including the next reaction method (NRM) [6], the optimized direct method (ODM) [7], and the sorting direct method (SDM) [8], have been proposed to help relieve some of the computational demands of SSA, but these methods are still time-consuming for practical applications. A much more efficient method is the final all possible steps (FAPS) approach [9], which produces significant gains in the efficiency of the statistical properties for the simulation of biochemical reactions.

To speed up discrete stochastic simulation, the tau-leaping method as an approximate simulation strategy has been proposed by Gillespie [10]. By using Poisson random numbers, the tau-leaping method leaps over many reactions without a significant loss of accuracy. The tau-leaping method makes a natural connection between SSA in the discrete stochastic regime and the explicit Euler method applied to the chemical Langevin equation in the continuous stochastic regime, and to the reaction rate equations (RREs) in the continuous deterministic regime. It seems likely that some forms of the tau-leaping method will be required to successfully simulate most biological systems. However, the original explicit tau-leaping method is not efficient in many situations. Several improvements for the tau-leaping method have recently been proposed. For instances, the binomial tau-leaping methods [11,12] and the modified binomial tau-leaping methods [13,14] have been proposed to avoid negative populations in simulations. The procedures for selecting leap time τ in tau-leaping methods have been proposed by Gillespie and Petzold [15] and Cao *et al.* [16]. The implicit, trapezoidal and adaptive tau-leaping methods [17−19] have also been developed to simulate chemically reacting systems with stiffness. The unbiased tau-leaping methods [20] are used to overcome the biases in tau-leaping methods. In short, the essence of these SSAs is to find the PDFs of species in well-stirred chemically reacting systems.

[*] Author to whom correspondence should be addressed. E-mail: xjpeng@shnu.edu.cn

DOI:10.1088/1674-0068/22/05/502-510

The support vector machine (SVM) proposed by Vapnik *et al.* is a novel approach for solving pattern recognition (binary) problems in machine learning field [21−23]. SVM brings along a bunch of advantages compared with other methods, some of which are: First, SVM is a quadratic programming problem (QPP), which assures that, compared with the other methods, such as Neural Networks [24], it is the unique (global) solution once its solution is obtained. Second, its sparsity assures a better generalization property. Third, SVM implements the structural risk minimization principle that minimizes the upper bound of the generalization error, which leads to be suitable for small sample-size problems. Fourth, SVM has a common ground/formulation for the class separable and inseparable problems as well as for linear and nonlinear problems. Last, it has clear geometric intuitions on classification problems [25]. Due to the above merits, SVM has been successfully used in many application fields [26], and extent to support vector regression (SVR) [22,23,27], which can be used for function and density estimation problems.

While SVR approach achieves the good generalization performance, the computational complexity of SVR may prove to be burdensome in real-time or near real-time applications because the solution of the convex optimization problem. Recently, we proposed a novel efficient SVR algorithm, called as twin support vector regression (TSVR) [28]. TSVR aims to generate two nonparallel functions such that each function determines the ε-insensitive down or up-bound of the unknown regressor. TSVR solves two smaller sized QPPs instead of solving large one in a classical SVR. However, the formulation of TSVR is totally different from that of SVR in one fundamental way. In TSVR we solve a pair of QPPs, whereas, in SVR, we solve a single QPP. Further, in SVR the QPP has two groups of constraints for all data points, but in TSVR, only one group of constraints for all data points are used in each QPP. This strategy of solving two smaller sized QPPs, rather than one large QPP, makes TSVR work faster than standard SVR.

This work focuses on the essence of SSAs that is to find the PDF of species in well-stirred chemically reacting systems. Note that the work of approximating the PDF of species can be viewed as the density estimation or regression problem. Intuitively, we can use SVR to approximate the PDF of species in well-stirred chemically reacting systems. However, as the above discussion, SVR is also time-consuming for real application. To improve the simulation speed, in this work we combine TSVR into the stochastic simulations algorithm, we present a novel method for approximating the PDF of species, termed as the twin support vector regression based stochastic simulations algorithm (TS^{3}A). For brief, we denote S^{3}A as the Support vector regression based Stochastic Simulations Algorithm in order to validate the performance of our TS^{3}A. Numerical simulation results indicate that this proposed TS^{3}A can be apply to a wide range of chemically reacting systems and obtain significant improvements on efficiency and accuracy with fewer simulating runs over the existed SSAs. Besides, it also derives better accuracy and efficiency than S^{3}A.

II. STOCHASTIC SIMULATION ALGORITHMS

Consider a well-stirred system of M chemically reacting channels $\{R_1, \ldots, R_M\}$ with N molecular species $\{S_1, \ldots, S_N\}$. The dynamical state vector $X(t)=(X_1(t), \ldots, X_N(t))^T$ describes the state of the N molecular species $\{S_1, \ldots, S_N\}$ in the system, where $X_i(t)$ is the number of molecules of species S_i at time t. We assume that, in general, the system is well-stirred and in thermal equilibrium. Each R_j is characterized by its corresponding propensity function $a_j(x)$ and state change vector $v_j=(v_{1j}, \ldots, v_{Nj})^T$, where $a_j(x)\mathrm{d}t$ is the probability, given $X(t)=x$, that one reaction R_j will occur in the next infinitesimal time interval $[t, t+\mathrm{d}t)$ and v_{ij} is the change in the number of species S_i due to one R_j reaction.

Gillespie's exact SSA generates two uniformly distributed random numbers in the interval (0, 1), r_1, and r_2, for each step. Assuming

$$a_0(x) = \sum_{j=1}^{M} a_j(x) \tag{1}$$

the time of the next reaction to occur is given by $t+\tau$, where τ is given by

$$\tau = \frac{1}{a_0(x)} \ln \frac{1}{r_1} \tag{2}$$

and the index j of the next selected reaction is the smallest integer in $\{1, \ldots, M\}$ such that

$$\sum_{j'=1}^{j} a_{j'}(x) > r_2 a_0(x) \tag{3}$$

Then the system is updated by $X(t+\tau)=x+v_j$. This process repeats until the desired end time T or other condition is reached.

Because it must precede one reaction at a time, SSA may be very slow for many practical problems. To speed up stochastic simulation, Gillespie proposed an approximate tau-leaping simulation method. The basic idea of tau-leaping method is to ask the question: How many times does each reaction channel fire in each subinterval? In each step, the tau-leaping method proceeds with many reactions. This is achieved at the cost of some accuracy. Define $K_j(\tau; x, t)$ as the number of times, given $X(t)=x$, that reaction channel R_j will fire in the interval $[t, t+\tau)$, $j=1, \ldots, M$. The tau-leaping method assumes the leap condition: for the current state x, require τ

DOI:10.1088/1674-0068/22/05/502-510

being small enough that the change in the state during $[t, t{+}\tau)$ will be so small that no propensity function will suffer an appreciable change in its value. $K_j(\tau; x, t)$ is then well approximated by the Poisson random variable with mean and variance $a_j(x)\tau$:

$$K_j(\tau; x, t) = P(a_j(x)\tau), \quad (j = 1, \dots, M) \tag{4}$$

The basic tau-leaping method proceeds as follows: choose a value for τ that satisfies the leap condition. Generate for each j=1, ..., M a sample value k_j of the Poisson random variable $P(a_j(x)\tau)$, and update the state by

$$X(t+\tau) = x + \sum_{j=1}^{M} k_j v_j \tag{5}$$

It has been found that when some consumed reactant species are present in small numbers, the original explicit tau-leaping method may drive some reactant populations negative. Several strategies have been proposed to circumvent this problem. Tian and Burrage [11], and Chatterjee *et al.* [12], proposed to replace the unbounded Poisson random numbers K_j with bounded binomial random numbers. But the bounds are often overly restrictive in many chemically reacting systems. Some improvements to the binomial tau-leaping methods have been presented [13,14]. Using the original Poisson tau-leaping method, Cao *et al.* also proposed a nonnegative Poisson tau-leaping algorithm that resolves this difficulty and establishes a smooth connection with the exact SSA [29].

In order to implement the tau-leaping method efficiently, Gillespie and Petzold formulated a procedure to quickly determine the largest value of τ that is compatible with the leap condition [15]. In a more recent work, Cao *et al.* developed an improvement for the tau selection formula [16]. More recently, to avoid the dilemma of the tau-leaping methods whether a preselected τ, without knowing at least an upper bound on the number of reactions that will occur in the next leap, well satisfies the leap condition or not, the K-leaping, R-leaping, and L-leaping methods have been introduced [30−32], in which the firing number of all reaction channels during a leap is first calculated, and the firing number of each reaction channel is determined by a multi-nominal random variable. To further improve the simulation accuracies of tau-leaping methods, Xu and Cai discussed the unbiased tau-leaping methods by using the Taylor expansions of the means and variance matrixes [20].

III. TWIN SUPPORT VECTOR REGRESSION

Here, we introduce TSVR [28] in brief. Let the samples to be trained be denoted by a set of l row vector A_i, i=1, 2, ..., l in the n-dimensional real space $\mathbf{R}^n$, where the i-th sample $A_i{=}(A_{i1}, A_{i2}, \dots, A_{in})$. Also let $A{=}(A_1; A_2; \dots; A_l)$ and let $Y{=}(y_1; y_2; \dots; y_l)$ denote the response vector of training samples, where $y_i{\in}\mathbf{R}$.

A. Linear TSVR

TSVR finds two functions $f_1(x){=}w_1{}^T x{+}b_1$ and $f_2(x){=}w_2{}^T x{+}b_2$, each one determines the ε-insensitive up- and down-bound regressor. Specially, given the training data points (A, Y), the function $f_1(x)$ determines the ε_1-insensitive down-bound regressor, while the function $f_2(x)$ determines the ε_2-insensitive up-bound regressor. The end regressor is decided by the average of these two functions. TSVR is obtained by solving the following pair of QPPs:

$$\begin{aligned} &\min \ \frac{1}{2}\|Y - e\varepsilon_1 - (Aw_1 + eb_1)\|^2 + C_1 e^T \xi \\ &\text{s.t.} \ \ Y - (Aw_1 + eb_1) \geq e\varepsilon_1 - \xi, \quad \xi \geq 0, \end{aligned} \tag{6}$$

$$\begin{aligned} &\min \ \frac{1}{2}\|Y + e\varepsilon_2 - (Aw_2 + eb_2)\|^2 + C_2 e^T \eta \\ &\text{s.t.} \ \ (Aw_2 + eb_2) - Y \geq e\varepsilon_2 - \eta, \quad \eta \geq 0, \end{aligned} \tag{7}$$

where $C_1, C_2{>}0$, $\varepsilon_1, \varepsilon_2{\geq}0$ are parameters, ξ and η are slack vectors, and e is the vector of ones of appropriate dimensions.

To derive the dual QPPs of TSVR, we introduce the Lagrange functions. By introducing the Karush-Kuhn-Tucker (KKT) necessary and sufficient optimality conditions for Eq.(6), we obtain the dual QPP for Eq.(6) as follows:

$$\begin{aligned} \max \ & -\frac{1}{2}\alpha^T G(G^T G)^{-1} G^T \alpha + \\ & f^T G(G^T G)^{-1} G^T \alpha - f^T \alpha \\ \text{s.t.} \ & 0 \leq \alpha \leq C_1 e \end{aligned} \tag{8}$$

where α is the Lagrange vector, $G{=}[A \ \ e]$, and $f{=}Y{-}e\varepsilon_1$. Optimizing Eq.(8) leads to

$$\begin{bmatrix} w_1{}^T & b_1 \end{bmatrix}^T = (G^T G)^{-1} G^T (f - \alpha) \tag{9}$$

Similarly, we obtain the dual of Eq.(7) as

$$\begin{aligned} \max \ & -\frac{1}{2}\gamma^T G(G^T G)^{-1} G^T \gamma - \\ & h^T G(G^T G)^{-1} G^T \gamma - h^T \gamma \\ \text{s.t.} \ & 0 \leq \gamma \leq C_2 e \end{aligned} \tag{10}$$

where γ is the Lagrange vector, $G{=}[A \ e]$, $h{=}Y{+}e\varepsilon_2$. After optimizing it, we obtain

$$\begin{bmatrix} w_2{}^T & b_2 \end{bmatrix}^T = (G^T G)^{-1} G^T (h + \gamma) \tag{11}$$

Note that TSVR is comprised of a pair of QPPs such that each one determines the one of up- or down-bound function by using only one group of constraints compared with SVR. Hence, TSVR gives rise to two smaller

DOI:10.1088/1674-0068/22/05/502-510

sized QPPs, is approximately four times faster than SVR in theory. Once the two dual QPPs are optimized, the two up- and down-bound functions are obtained. Then the estimated regressor is constructed by as follows:

$$f(x) = \frac{1}{2}\left(w_1 + w_2\right)^T x + \frac{1}{2}\left(b_1 + b_2\right) \tag{12}$$

B. Kernel TSVR

In order to extend to the nonlinear case, we consider the following kernel-generated functions instead of linear ones

$$\begin{aligned} f_1(x) &= K(x^T, A^T)w_1 + b_1 \\ f_2(x) &= K(x^T, A^T)w_2 + b_2 \end{aligned} \tag{13}$$

where K is an appropriately chosen kernel, such as the Gaussian kernel. Note that if the linear kernel is used the both linear functions are the special ones of Eq.(13). As the linear case, we construct a pair of optimization problems as follows:

$$\begin{aligned} \min\ & \frac{1}{2}\left\|Y - e\varepsilon_1 - [K(A, A^T)w_1 + eb_1]\right\|^2 + C_1 e^T \xi \\ \text{s.t.}\ & Y - [K(A, A^T)w_1 + eb_1] \geq e\varepsilon_1 - \xi,\ \ \xi \geq 0 \end{aligned} \tag{14}$$

$$\begin{aligned} \min\ & \frac{1}{2}\left\|Y + e\varepsilon_2 - [K(A, A^T)w_2 + eb_2]\right\|^2 + C_2 e^T \eta \\ \text{s.t.}\ & [K(A, A^T)w_2 + eb_2] - Y \geq e\varepsilon_2 - \eta,\ \ \eta \geq 0 \end{aligned} \tag{15}$$

If define H=$[K(A, A^T)\ e]$, then we obtain the dual QPPs as follows:

$$\begin{aligned} \max\ & -\frac{1}{2}\alpha^T H(H^T H)^{-1}H^T\alpha + \\ & f^T H(H^T H)^{-1}H^T\alpha - f^T\alpha \\ \text{s.t.}\ & 0 \leq \alpha \leq C_1 e \end{aligned} \tag{16}$$

$$\begin{aligned} \max\ & -\frac{1}{2}\gamma^T H(H^T H)^{-1}H^T\gamma - \\ & h^T H(H^T H)^{-1}H^T\gamma - h^T\gamma \\ \text{s.t.}\ & 0 \leq \gamma \leq C_2 e \end{aligned} \tag{17}$$

After optimizing them, we obtain the following end regressor:

$$f(x) = \frac{1}{2}K(x, A^T)(w_1 + w_2) + \frac{1}{2}(b_1 + b_2) \tag{18}$$

$$\begin{aligned} [w_1{}^T\ b_1]^T &= (H^T H)^{-1}H^T(f - \alpha) \\ [w_2{}^T\ b_2]^T &= (H^T H)^{-1}H^T(h + \gamma) \end{aligned} \tag{19}$$

IV. TWIN SUPPORT VECTOR REGRESSION BASED STOCHASTIC SIMULATIONS ALGORITHM

Recall that the essence of SSAs is to find the PDFs of molecular species in well-stirred systems. Obviously, we can combine some regression tools to estimate the PDF with some simulation samples. However, most regression tools need a lot of samples in order to derive the good approximation for PDF, which causes to have to make so many SSAs' runs. Fortunately, theories and applications show that SVM is a small-sample machine learning tool. In other words, it only needs very few samples in order for a better separate hyperplane (Classification) or regressor (Regression).

We present an efficient simulation algorithm for chemically reacting systems by integrating TSVR and SSA, called as twin support vector regression based stochastic simulation algorithm (TS^{3}A). TS^{3}A is a simple simulation algorithm, which only combines TSVR with SSA in order to derive efficiently the PDF of molecular species. TS^{3}A can be summarized as follow. ALGORITHM 1: (i) Set the initial populations of all species, the end time T and the number of runs k. (ii) Make k runs of SSA. Prepare the training data points using the data preparation method. (iii) Train TSVR with suitable parameters. (iv) Use the trained TSVR regressor to estimate the PDF of species.

A. Training data set preparation

A key step in TS^{3}A is to prepare the training data set since one SSA run only obtains a state of the molecular species at time T. For the molecular species, our data preparation method includes the following three steps: (i) Compute the frequency f_i of each state x_i derived by k runs of SSA. (ii) Generate the training samples (x_i, f_i). (iii) Rescale (x_i, f_i) as A_i=$(x_i$−$\bar{x})/\text{std}(x)$, and y_i=$f_i/\max_j\{f_j\}$, where $\bar{x}$, std(x), and $\max_j\{f_j\}$ represent the mean, standard deviation of state variable, and the maximum frequency of state, respectively.

We point out that the 3th step in this data preparation method is important to obtain a better PDF estimation. This is because for many cases the frequency f_i's are often very small while the range of state is large.

B. TSVR training and parameter selection

After the positive and negative training examples are gathered, some parameters should be initialized firstly, such as the type of kernel function, its associated parameter, the regularization parameter C_1 and C_2 in the structural risk function, and the up- and down-bound parameters ε_1 and ε_2. In our work, we set C_1=C_2 and ε_1=ε_2 in order to reduce the burden of parameter selection. To optimize these parameters, we applied 10-fold cross validation method to get the optimal parameters. Once the best parametric setting (*i.e.*, the type of the kernel function and its associated parameter) is determined, the TSVR is retrained using all the available samples in the training set to obtain the final form of regressor. The resulting regression function will be used to the approximation of PDF of molecular species in the system.

C. Some remarks for TS^{3}A

There are some remarks for TS^{3}A we have to point out when applying it to estimate the PDF of chemical species.

Remark 1: Note that TSVR is only a regression tool, which means it may be possible to output the negative probability in some states. For this case we set the predicted state probability as zero for brief.

Remark 2: It is important to evaluate the performance of a given algorithm. In this work we use the histogram distance [16] as the criterion of generalization performance of TS^{3}A.

Remark 3: Generally, it is impossible to obtain the theory PDFs of chemical species based on the chemical master equations (CMEs) in most chemically reacting systems. For these cases, in order to evaluate the performance of TS^{3}A, we first calculate the standard histograms of the populations of one molecular species at the end of simulation from enough repeated SSA runs, and make a series of runs of Gillespie's SSA and our TS^{3}A, then compare the histogram distance [16] between the results of SSA and our TS^{3}A. For these cases, we think it is a good approximate result if the histogram distance is less than 0.05 compared with the standard histograms [16].

Remark 4: To evaluate the performance of our TS^{3}A, we compare it with the support vector regression based stochastic simulation algorithm (S^{3}A). S^{3}A is constructed as ALGORITHM 1, in which only TSVR is replaced by SVR.

Remark 5: TS^{3}A and S^{3}A effectively combine the advantage that SVM is a small-sample machine learning tool when estimating the PDF of molecular species. Hence, an important merit of TS^{3}A and S^{3}A is that they can derive good PDF approximations of chemical species in chemically reacting systems with fewer runs of SSA.

V. EXPERIMENTS AND DISCUSSION

To demonstrate the performance of our TS^{3}A, the independent double-channel model [31], the decaying-dimerizing model [10,18,20,31], the stiff decaying-dimerizing model [10,17,19], and the Schlögl model [16,17] are simulated. To assess the accuracy of our TS^{3}A, we first make enough repeated SSA runs to make the histogram nearly smooth and regard this histogram as the real distribution, then we compute the histogram distances between SSA, TS^{3}A, S^{3}A, and the real distribution after a series of runs. Since the histogram distance provides a measure of the error, we compare the number of runs of these two algorithms with the same histogram distance. All simulations are run in MATLAB [33] on a personal computer with a 2.26 GHz CPU and 1 Gbyte memory running WINDOWS XP.

A. Independent double-channel model

This simple reaction system [31] includes two reaction channels:

$$\begin{aligned} S_1 &\xrightarrow{c_1} \phi \\ S_2 + S_3 &\xrightarrow{c_2} S_4 \end{aligned} \tag{20}$$

The two reaction channels are actually independent. If the initial number of S_1 molecules at t=0 is $X_1(0)$=$\bar{x}_1$, the PDF of $X_1(t)$ can be obtained from its CME as

$$\begin{aligned} p[X_1(t) = x] = {} & \frac{\hat{x}_1!}{x!(\hat{x}_1 - x)!} \cdot \\ & \left(e^{-c_1 t}\right)^x \left(1 - e^{-c_1 t}\right)^{\hat{x}_1 - x} \\ & (x = 0, \ \ldots, \ \hat{x}_1) \end{aligned} \tag{21}$$

Hence, we can compare the histograms of $X_1(t)$ generated from SSA, TS^{3}A, and S^{3}A with Eq.(21). In our work, we simulate this model using the rate constants c_1=1, c_2=10^{-4}, and the initial conditions X_1=X_2=3×10^3, X_3=10^4, and X_4=0. We make a series of simulations with SSA, TS^{3}A, and S^{3}A in time interval [0,2], and then calculate the practical histogram of $X_1(2)$ and the histogram distances of SSA, TS^{3}A, and S^{3}A of $X_1(2)$ by using the histogram distance formula.

Figure 1 depicts the histogram distance versus run number. It has been seen that both TS^{3}A and S^{3}A outperform Gillespie's exact SSA, since they produce smaller histogram distances given the same run numbers. In other words, both TS^{3}A and S^{3}A take less CPU time for a given histogram distance than SSA, since the CPU time for training TSVR and SVR is very small (less than 1 s). For instance, both TS^{3}A and SS^{3}A need less then 10^3 runs to obtain a histogram distance 0.05, while SSA needs about 11×10^3 runs in order to derive the same histogram distance. Meanwhile, both TS^{3}A and S^{3}A still derive smaller histogram distances than SSA after 10^5 runs. Besides, it has been seen that our TS^{3}A obtains much smaller histogram distances than S^{3}A given the same run number. For instance, the histogram distance of our TS^{3}A is less then 0.03 given 10^3 SSA runs. Considering that the CPU time for training TSVR is four times longer than SVR, we can claim that TS^{3}A obtains far better performance than S^{3}A and SSA.

B. Decaying-dimerizing model

This simple decaying-dimerizing model [10,18,20,31] includes three reactant species and four reactant channels, which are

$$\begin{aligned} S_1 &\xrightarrow{c_1} \phi \\ S_1 + S_1 &\xrightarrow{c_2} S_2 \\ S_2 &\xrightarrow{c_3} S_1 + S_1 \\ S_2 &\xrightarrow{c_4} S_3 \end{aligned} \tag{22}$$

DOI:10.1088/1674-0068/22/05/502-510

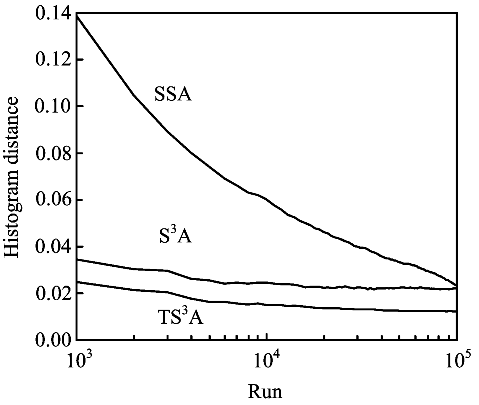

FIG. 1 Histogram distances of $X_1(t)$ at t=2 s versus run number for the double-channel model Eq.(20).

In these reactions, a decay-prone monomer S_1 reversibly dimerizes to an unstable form S_2, which can convert to a stable form S_3. The propensity functions of the decaying-dimerizing model are:

$$\begin{aligned} a_1(x) &= c_1x_1 \\ a_2(x) &= \frac{1}{2}c_2x_1(x_1-1) \\ a_3(x) &= c_3x_2 \\ a_4(x) &= c_4x_2 \end{aligned} \tag{23}$$

Detailed simulations of this system can be found in Ref.[10] based on the following reaction rate c_1=1, c_2=0.002, c_3=0.5, and c_4=0.04.

In our simulations, we simulate the decaying-dimerizing model with the same values of reaction rates and the following initial populations $X_1(0)$=4150, $X_2(0)$=39565, $X_3(0)$=3445. We first make 10^5 runs with SSA in order to derive the approximate smooth PDFs of species in time interval [0, 10] since it is impossible to obtain the theoretical PDFs using CMEs, and make a series of simulations with SSA, TS^{3}A and S^{3}A in this time interval [0, 10]. We then calculate the histogram distances between the histogram of 10^5 SSA runs and the histograms of a series of SSA, TS^{3}A, and S^{3}A runs.

Figure 2 shows the histogram distances of $X_1(10)$, and $X_2(10)$ versus the run number derived by these three methods. It has been seen that both TS^{3}A and S^{3}A outperform Gillespie's exact SSA again for the same reason. For the species S_1, both methods only need less then 10^3 runs to derive a histogram distance 0.05, while SSA needs about 10^4 runs. The similar phenomenon shows in the simulation results of species S_2, in which both methods need fewer runs than SSA in order to derive the small histogram distance 0.05. It thus can claim again that TS^{3}A and S^{3}A take less CPU time than SSA for a given histogram distance. Similarly, the simulation results also show that our TS^{3}A obtains better performance than S^{3}A, in which our TS^{3}A obtains less histogram distance than S^{3}A given the same SSA runs. It has been also seen that, in Fig.2, the histogram distance of SSA becomes smaller than those of TS^{3}A and S^{3}A after about 5.5×10^4 and 4×10^4 SSA runs for the species S_1. This is because the practical PDF of species S_1 in simulations is only approximated by only 10^5 SSA runs, which causes the histogram distance of SSA to become zero at last, but as the machine approaches, the histogram distances of TS^{3}A and S^{3}A can not become zero. It can not claim that TS^{3}A and S^{3}A are worse compared with SSA under this meaning. In fact, the histogram distances of TS^{3}A and S^{3}A for S_1 only about 0.02 after 10^5 runs. Similarly, we have the same conclusion on the simulation of S_2. All in all, TS^{3}A and S^{3}A, especially the former, obtain better performances than SSA.

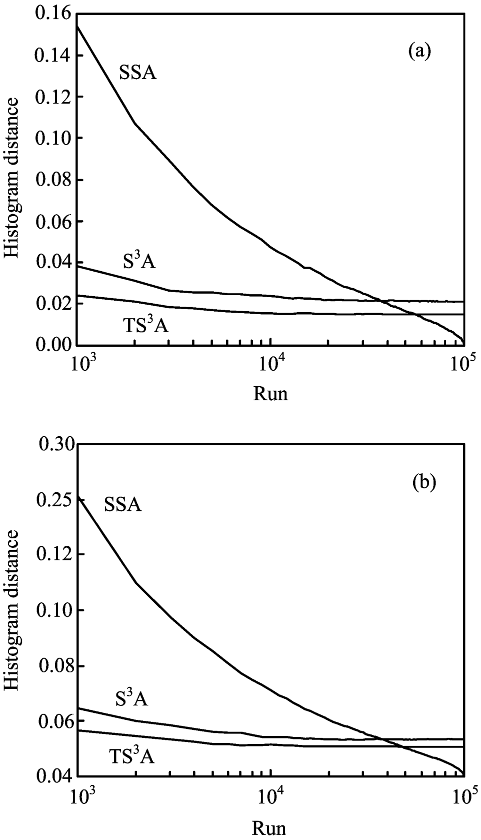

FIG. 2 Histogram distances of $X_1(t)$ (a) and $X_2(t)$ (b) at t=10 s versus run number for the decaying-dimerizing model Eq.(22).

C. Stiff decaying-dimerizing model

The stiff decaying-dimerizing model [10,17,19] has the same reaction channels compared to the ordinary decaying-dimerizing model Eq.(22), but has different reaction rates, *i.e.*, in the stiff decaying -dimerizing model, we choose values for the reaction rates c_1=1, c_2=10, c_3=10^3, and c_4=0.1, which will render the problem stiff,

DOI:10.1088/1674-0068/22/05/502-510

and set the initial conditions $X_1(0)$=400, $X_2(0)$=798, and $X_3(0)$=0 in the simulations. This model is well studied in recent work. We notice that X_1 and X_2 vary rapidly but X_3 varies slowly in the model. We make 11×10^4 runs with SSA in order to derive the approximate smooth PDFs of species in time interval [0, 0.2], and make a series of simulations with SSA, TS^{3}A and S^{3}A in this time interval [0, 0.2]. We then calculate the histogram distances between the histogram of 11×10^4 SSA runs and those of a series of SSA, T S^{3}A, and S^{3}A runs.

Figure 3 shows the histogram distances of $X_1(0.2)$ and $X_2(0.2)$, versus the run number. It can be seen that TS^{3}A and S^{3}A obtain the similar speed-up results. For instance, TS^{3}A only needs about 10^3 runs if gives a histogram distance 0.03, and S^{3}A needs about 3×10^3 runs for this histogram distance. But SSA needs about 15×10^3 runs for estimating the PDF of $X_1(0.2)$. These results mean our TS^{3}A accelerates the run speed about 15 times, and S^{3}A accelerates the run speed about 5 times in this case (The CPU time for training SVR or TSVR is less than a run of SSA for this stiff system). Similarly, given the same histogram distance 0.03, TS^{3}A only needs less 10^3 runs for estimating the PDF of $X_2(0.2)$. While S^{3}A needs 10^4 SSA runs, which is similar to SSA. These results confirm that TS^{3}A obtains an obvious improvement than S^{3}A. Note that for this system, it is very time-consuming to approximate the PDFs of species using SSA because of stiffness. To improve the efficiency of SSA on stiff systems, some methods have been discussed recently. However, these methods often have the large histogram distances. Hence we can conclude that our TS^{3}A is a better method for approximating the PDFs of chemical species than the other methods. The similar results appear in the histogram distances of TS^{3}A and S^{3}A of $X_2(0.2)$. In short, our TS^{3}A outperforms the other two methods because of its efficiency and accuracy. We also notice that the histogram distances of SSA become smaller than those of TS^{3}A and S^{3}A after about 45×10^3 and 22×10^3 runs for approximating the PDF of $X_1(0.2)$. This is because that in the experiments we take the approximate PDF of $X_1(0.2)$ as the practical one when evaluating the performances of these methods. The similar phenomenon is shown in the process of estimating the PDF of $X_2(0.2)$ as the same reason.

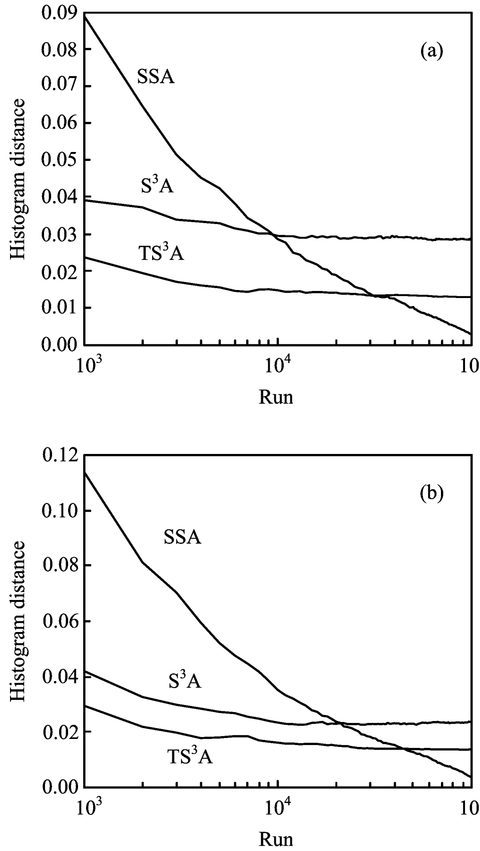

FIG. 3 Histogram distance of $X_1(t)$ (a) and $X_2(t)$ (b) at t=0.2 versus run number for the stiff decaying-dimerizing model.

D. Schlögl model

This Schlögl model is famous for its bistable steady-state distribution. The reactions are:

$$\begin{aligned} B_1 + 2X &\overset{c_1}{\rightarrow} 3X \\ 3X &\overset{c_2}{\rightarrow} B_1 + 2X \\ B_2 &\overset{c_3}{\rightarrow} X \\ X &\overset{c_4}{\rightarrow} B_2 \end{aligned} \tag{24}$$

where B_1 and B_2 denote buffered species whose respective molecular populations N_1 and N_2 are assumed to remain essentially constant over the time interval of interest. There is only one time varying species X. The state change vectors are v_1=v_3=1, v_2=v_4=−1, and the propensity functions are:

$$\begin{aligned} a_1(x) &= \frac{1}{2}c_1N_1x(x-1) \\ a_2(x) &= \frac{1}{6}c_2x(x-1)(x-2) \\ a_3(x) &= c_3N_2 \\ a_4(x) &= c_4x \end{aligned} \tag{25}$$

For some values of the parameters this model has two stable states, and that is the case for the parameter values we have chosen here: c_1=3×10^{-7}, c_2=10^{-4}, c_3=10^{-3}, c_4=3.5, N_1=10^5, and N_2=2×10^5.

We make 10^5 SSA runs from the initial state $X(0)$=250 to time t=4 in order to derive the approximate PDF of species X at time t=4, and then calculate the histogram distance between the histogram of 10^5 SSA runs and those of a series of SSA, TS^{3}A, and S^{3}A runs.

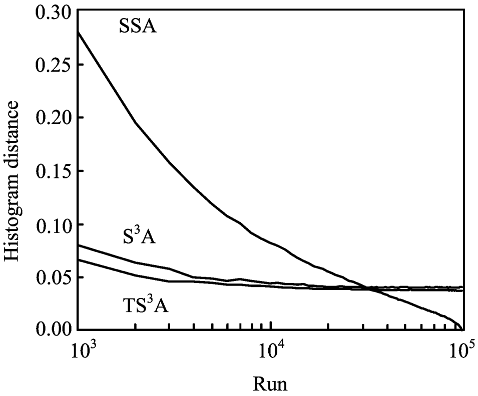

FIG. 4 Histogram distance of $X(t)$ at t=4 s versus run number for the Schlögl model Eq.(24).

Figure 4 lists the histogram distances obtained from these three methods of $X(4)$ versus the run number. The result shows that our TS^{3}A still derives the best performance among these methods, in which it needs the least CPU time for a given histogram distance. For example, TS^{3}A derives a histogram distance 0.05 after about 2.2×10^3 runs, but Gillespie's exact SSA and S^{3}A need about 25×10^3 and 4×10^3 runs, respectively. That is, TS^{3}A outperforms SSA and S^{3}A obviously in this model, which is similar to the former three models. We also notice that the histogram distance of SSA is smaller than TS^{3}A and S^{3}A after about 32×10^3 runs but both are stable at the histogram distance 0.04. We can use the same reason to interpret this phenomenon.

Zhou *et al.* proposed a novel FAPS approach by considering all situations at the final step in order to improve the simulation speed of SSA [9]. To further accelerate the efficiency of our TS^{3}A, we can combine our TSVR and FAPS algorithms.

VI. CONCLUSION

Stochastic simulation of chemically reacting system is a topic of current interest, since discreteness and stochasticity are important in systems formed by living cells where some key reactant molecules may be present in small numbers. Gillespie's SSA is an essentially exact numerical simulation method for well-stirred systems and is widely used in the simulation of biochemical systems. But because SSA keeps track of every reaction event, it is impractical for many realistic problems, in spite of recently significant improvements.

Motivated by the recent work of Gillespie for time acceleration of Monte Carlo methods in well-mixed systems, a series of tau-leaping methods were presented, which included binomial tau-leaping, nonnegative tau-leaping, implicit and trapezoidal tau-leaping, and unbiased tau-leaping methods and so on. An important merit of these methods is that they improve the simulation efficiency with acceptant accuracies. However, these algorithms have certain deficiencies in their applications.

In this work, we derives an efficient stochastic simulation algorithm by combing the twin support vector regression (TSVR), called as the twin support vector regression based stochastic simulation algorithm (TS^{3}A). To compare the performance of TS^{3}A, it also introduces the support vector regression based stochastic simulation algorithm (S^{3}A). The numerical examples indicate that the presented TS^{3}A is much more efficient than S^{3}A and SSA, especially than the latter. In other words, it needs much fewer CPU time for a given histogram distance compared with SSA and S^{3}A. The further work we should do includes how to extend it into the simulations for chemically reacting systems with delays and tau-leaping methods; another is how to realize it as software.

VII. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No.30871341), the National High-Tech Research and Development Program of China (No.2006AA02-Z190), the Shanghai Leading Academic Discipline Project (No.S30405), and the Natural Science Foundation of Shanghai Normal University (No.SK200937).

[1] N. Fedoroff and W. Fontana, Science. **297**, 5584 (2002).
[2] T. Ideker, T. Galitski, and L. Hood, Annu. Tev. Genom. Human Genet. **2**, 343 (2001).
[3] M. Kærn, T. C. Elston, W. J. Blake, and J. J. Collins, Nat. Tev. Genet. **6**, 451 (2005).
[4] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[5] D. T. Gillespie, J. Chem. Phys. **81**, 2340 (1977).
[6] M. Gibson and J. Bruck, J. Chem. Phys. **104**, 1876 (2000).
[7] Y. Cao, H. Li, and L. R. Petzold, J. Comput. Phys. **121**, 4059 (2004).
[8] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova, Comput. Biol. Chem. **30**, 39 (2006).
[9] W. Zhou, X. J. Peng, Z. L. Yan, and Y. F. Wang, Appl. Math. Mech. **3**, 379 (2008).
[10] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[11] T. H. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).
[12] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis, J. Chem. Phys. **122**, 024112 (2005).
[13] X. J. Peng and Y. F. Wang, J. Chem. Phys. **126**, 224109 (2007).
[14] M. Pettigrew and H. Resat, J. Chem. Phys. **126**, 084101 (2007).
[15] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).
[16] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **124**, 044109 (2006).

DOI:10.1088/1674-0068/22/05/502-510

[17] Y. Cao and L. Petzold, *Proceedings of Foundations of Systems Biology in Engineering*, (FOSBE 2005), 149, (2005).

[18] Y. Cao, D. T. Gillespie, and L. Petzold, J. Chem. Phys. **126**, 224101 (2007).

[19] M. Rathinam, Y. Cao, L. Petzold, and D. Gillespie, J. Chem. Phys. **119,** 12784 (2003).

[20] Z. Y. Xu and X. D. Cai, J. Chem. Phys. **128**, 154112 (2008).

[21] V. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge: Cambridge University Press, (2002).

[22] V. N. Vapnik, *The Natural of Statistical Learning Theory*, New York: Springer, (1995).

[23] V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, (1998).

[24] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, (1996).

[25] K. P. Bennett and E. J. Bredensteiner, *Proceedings of 17th Int. Conf. Machine Learning*, P. Langley, Ed. San Mateo, CA, 57 (2000).

[26] E. Osuna, R. Freund, and F. Girosi, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan: Pattern Recogn, 130, (1997).

[27] S. Mukherjee, E. Osuna, and F. Girosi, *Proceedings of IEEE Workshop on Neural Networks and Signal Processing*, FL: Amelia Island, 24, (1997).

[28] X. J. Peng, Neural Networks, doi:10.1016/j.neunet.2009.07.002, (2009).

[29] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **123**, 054104 (2005).

[30] A. Auger, P. Chatelain, and P. Koumoutsakos, J. Chem. Phys. **125**, 084103 (2006).

[31] X. D. Cai, Z. Y. Xu, J. Chem. Phys. **126**, 074102 (2007).

[32] X. J. Peng and Y. F. Wang, Appl. Math. Mech. **28**, 1361 (2007).

[33] MATLAB, The MathWorks, Inc., 1994-2001, http://www.mathworks.com.

DOI:10.1088/1674-0068/22/05/502-510