Kernel Least-Squares Temporal Difference Learning

Xin Xu[1,2], Tao Xie[1], Dewen Hu[2], and Xicheng Lu[1]

[1]School of Computer, National University of Defense Technology, Changsha, 410073, P.R.China

[2]College of Mechantronics Engineering and Automation, National University of Defense Technology, Changsha, 410073, P.R.China

xuxin_mail@263.net

Abstract

Kernel methods have attracted many research interests recently since by utilizing Mercer kernels, non-linear and non-parametric versions of conventional supervised or unsupervised learning algorithms can be implemented and usually better generalization abilities can be obtained. However, kernel methods in reinforcement learning have not been popularly studied in the literature. In this paper, we present a novel kernel-based least-squares temporal-difference (TD) learning algorithm called KLS-TD(λ), which can be viewed as the kernel version or nonlinear form of the previous linear LS-TD(λ) algorithms. By introducing kernel-based nonlinear mapping, the KLS-TD(λ) algorithm is superior to conventional linear TD(λ) algorithms in value function prediction or policy evaluation problems with nonlinear value functions. Furthermore, in KLS-TD(λ), the eligibility traces in kernel-based TD learning are derived to make use of data more efficiently, which is different from the recent work on Gaussian Processes in reinforcement learning. Experimental results on a typical value-function learning prediction problem of a Markov chain demonstrate the effectiveness of the proposed method.

Keyword: Reinforcement learning, Kernel methods, Temporal difference, Markov chain.

I. Introduction

Recently, kernel methods or kernel machines [5] are popularly studied to realize non-linear and non-parametric versions of conventional supervised or unsupervised machine learning algorithms. The main idea behind kernel machines is that inner products in a high-dimensional feature space can be represented by a Mercer kernel function so that conventional learning algorithms in linear spaces may be transformed to nonlinear algorithms without explicitly computing the inner products in high-dimensional feature spaces. This idea, which is usually called the "kernel trick", has been widely applied in various kernel-based supervised and unsupervised learning problems. In supervised learning, the most popular kernel machines include support vector machines (SVMs) and the Gaussian process model for regression, which have been applied to many classification and regression problems [5] [6] [11]. In unsupervised learning, kernel principal component analysis (KPCA) and kernel independent component analysis have also been studied by many researchers [5] [14].

Unlike supervised learning and unsupervised learning, reinforcement learning (RL) is another class of machine learning algorithms that aim to solve Markov decision problems (MDPs) without complete model information [4][7]. To compute the optimal or near-optimal policies of MDPs, RL algorithms usually estimate the value functions of MDPs by observing data generated from state transitions. Since no explicit teacher signals can be obtained in RL, the estimation of value functions is different from the function regression problem in supervised learning. Furthermore, the state spaces of real-world MDPs are usually large or continuous. Thus, the value function approximation (VFA) problem has been an important research topic in RL. One basic technique for VFA in RL is temporal-difference (TD) learning that updates value function estimations based on temporal differences, i.e., the differences between two estimations of one stochastic variable at two successive time steps. To improve data efficiency in stochastic MDPs, eligibility traces are usually employed as an efficient mechanism in TD learning. Over the past ten years, TD learning algorithms with eligibility traces, which are usually called TD(λ) algorithms, have been widely studied in the literature[8]. For MDPs with large or continuous state spaces, TD(λ) algorithms with linear function approximators were popularly employed and some convergence results have been obtained [9][10].

As a class of linear regression algorithms, least-squares (LS) methods have also been studied in linear TD learning algorithms so that the learning efficiency of TD(λ) can be improved. For some recent work on linear TD(λ) based on LS methods, please refer to [1] and [12]. Although linear LS-TD learning has better performance than conventional linear TD(λ), nonlinear approximators such as neural networks have to used in TD learning when the approximation ability of linear approximators is inadequate. However, for TD learning with nonlinear function approximators, least-squares methods can not be employed directly and it has been shown that TD(λ) algorithms with conventional nonlinear approximators may diverge in simple cases [10].

To realize efficient and convergent TD learning with nonlinear function approximators, this paper presents a class of kernel-based least-squares TD learning algorithms with eligibilities, which is called KLS-TD(λ). The idea of KLS-TD(λ) is to make use of Mercer kernel functions to implement least-squares TD learning in a high-dimensional nonlinear feature space produced by a kernel-based feature mapping. Thus, compared to conventional linear TD(λ) and LS-TD(λ) algorithms, better performance of approximation accuracy can be obtained for KLS-TD(λ) in nonlinear VFA problems.

There are some recent work on kernel methods in reinforcement learning such as Gaussian processes model in TD(0) learning [3] and Gaussian processes model in policy iteration learning [2]. However, previous work did not study the eligibility traces in TD learning, which has been recognized as an important factor for the performance of TD learning prediction in stochastic problems. Furthermore, the kernel-based least-squares TD learning method studied in this paper is different from Gaussian processes in that no variance parameters of observation noise are required as *a priori*. Therefore, the main contributions of the paper include two aspects. One is the kernelized version of LS-TD learning, which can greatly improve the performance in estimating nonlinear value functions of Markov chains. The other is to derive the mechanism of kernel-based eligibility traces in nonlinear feature spaces. Experimental results on nonlinear VFA for Markov chains show that the KLS-TD(λ) algorithm has better performance in approximation precision of value functions than previous linear TD(λ) and LS-TD(λ).

This paper is organized as follows. In Section 2, an introduction on TD learning as well as linear TD(λ) algorithms and LS-TD(λ) algorithms is given. The KLS-TD(λ) algorithm is presented in Section 3. In Section 4, learning prediction experiments on a typical Markov chain with nonlinear value functions are described to illustrate the effectiveness of the proposed algorithm. And Section 5 draws conclusions.

II. Temporal Difference Learning for Prediction

Learning prediction of a stationary-policy MDP is to compute the value functions of the MDP without any prior model and it is a central problem in RL since optimal policies of MDPs are usually based on the estimation of value functions, which is usually called policy evaluation. Nevertheless, learning prediction in RL is different from that in supervised learning since there is no teacher signals for value function estimation except that all the stochastic state transitions have been observed for many times. As pointed out by Sutton [7][8], the prediction problem in supervised learning is single-step while value function prediction in reinforcement learning is a multi-step prediction problem. To realize multi-step prediction, a learning system must predict outcomes that depend on a future sequence of decisions. Therefore, the theory and algorithms of multi-step learning prediction for RL have received much attention and lots of research work has been carried out in the literature [4] [8].

Among the proposed multi-step learning prediction methods, temporal-difference (TD) learning [8] is one of the most popular methods. Some recent results on TD learning include the linear TD(λ) algorithm and the LS-TD(λ) algorithm, etc.

In this section, a brief discussion on the conventional linear TD(λ) algorithm and the LS-TD(λ) algorithm will be given. First of all, some mathematical notations are presented as follows.

Consider a Markov chain with states in a finite or countable infinite space S. The states of the Markov chain can be indexed as $\{1,2,\ldots,n\}$, where n is possibly infinite. Although the algorithms and results in the sequel are applicable to Markov chains with general state spaces, the discussion in this paper will be restricted within the cases with a countable state space to simplify the notation. The extension to Markov chains with a general state space only requires the translation of the matrix notation into operator notation.

Let the trajectory generated by the Markov chain be denoted by $\{x_t \,|t=0,1,2,\ldots;\ x_t \in S\}$. For each state transition from x_t to x_{t+1}, a scalar reward r_t is defined. The value function of each state is defined as follows:

$$V(i) = E\{\sum_{t=0}^{\infty} \gamma^t r_t \,|x_0 = i\} \tag{1}$$

where $0<\gamma\leq 1$ is a discount factor.

In TD(λ), there are two basic mechanisms which are the temporal difference and the eligibility trace, respectively. Temporal differences are defined as the differences between two successive estimations and have the following form

$$\delta_t = r_t + \gamma \widetilde{V}_t(x_{t+1}) - \widetilde{V}_t(x_t) \tag{2}$$

where x_{t+1} is the successive state of x_t, $\widetilde{V}(x)$ denotes the estimate of value function $V(x)$ and r_t is the reward received after the state transition from x_t to x_{t+1}.

As discussed in [8], the eligibility trace can be viewed as an algebraic trick to improve learning efficiency without recording all the data of a multi-step prediction process. This trick is originated

from the idea of using a truncated reward sum of Markov chains. In TD learning with eligibility traces, an *n*-step truncated return is defined as

$$R_t^n = r_t + \gamma r_{t+1} + ... + \gamma^{n-1} r_{t+n-1} + \gamma^n \tilde{V}_t(s_{t+n}) \tag{3}$$

For an absorbing Markov chain whose length is *T*, the weighted average of truncated returns is

$$R_t^\lambda = (1-\lambda)\sum_{n=1}^{T-t-1} \lambda^{n-1} R_t^n + \lambda^{T-t-1} R_T \tag{4}$$

where $0 \le \lambda \le 1$ is a decaying factor and

$$R_T = r_t + \gamma r_{t+1} + ... + \gamma^T r_T \tag{5}$$

R_T is the Monte-Carlo return at the terminal state. In each step of TD(λ), the update rule of value function estimation is determined by the weighted average of truncated returns defined above, i.e.,

$$\Delta \tilde{V}_t(s_i) = \alpha_t (R_t^\lambda - \tilde{V}_t(s_i)) \tag{6}$$

where α_t is a learning factor.

The update equation (6) can be used only after the whole trajectory of the Markov chain is observed. To realize incremental or online learning, eligibility traces are defined for each state as follows:

$$z_{t+1}(s_i) = \begin{cases} \gamma\lambda\, z_t(s_i) + 1, & \text{if } s_i = s_t \\ \gamma\lambda\, z_t(s_i), & \text{if } s_i \neq s_t \end{cases} \tag{7}$$

The online TD(λ) update rule with eligibility traces is

$$\tilde{V}_{t+1}(s_i) = \tilde{V}_t(s_i) + \alpha_t \delta_t z_{t+1}(s_i) \tag{8}$$

where δ_t is the temporal difference at time step *t*, which is defined in (2) and $z_0(s)=0$ for all *s*.

A. Linear TD(λ) Algorithm

In linear TD(λ) algorithms, value functions are represented as

$$\tilde{V}(x) = \phi^T(x) W = \sum_{j=1}^{n} \phi_j(x) w_j \tag{9}$$

where $W = [w_1, w_2, ..., w_n]$ is the weight vector and $\phi(x) = [\phi_1(x), \phi_2(x), ..., \phi_n(x)]$ is a linear basis function.

The update rule for linear TD(λ) algorithms is as follows

$$W_{t+1} = W_t + \alpha_t (r_t + \gamma \phi^T(x_{t+1}) W_t - \phi^T(x_t) W_t) \vec{z}_{t+1} \tag{10}$$

where the eligibility trace vector $\vec{z}_t(x)$ is defined as

$$\vec{z}_{t+1} = \gamma\lambda\vec{z}_t + \phi(x_t) \tag{11}$$

B. Least-squares TD(λ) Algorithm

In [10], the above linear TD(λ) algorithm is proved to converge with probability 1 under certain assumptions and the limit of convergence *W** is also derived, which satisfies the following equation.

$$E_0[A(X_t)]W^* - E_0[b(X_t)] = 0 \tag{12}$$

where $X_t = (x_t, x_{t+1}, z_{t+1})$ (t=1,2,…) form a Markov process, $E_0[\cdot]$ stands for the expectation with respect to the unique invariant distribution of $\{X_t\}$, and $A(X_t)$, $b(X_t)$ are defined as

$$A(X_t) = \vec{z}_t(\phi^T(x_t) - \gamma\phi^T(x_{t+1})) \tag{13}$$

$$b(X_t) = \vec{z}_t r_t \tag{14}$$

The LS-TD(λ) algorithm proposed in [1] computes the weight vector *W* by solving equation (12) directly, i.e.,

$$W_{LS-TD(\lambda)} = A_T^{-1} b_T = (\sum_{t=1}^{T} A(X_t))^{-1} (\sum_{t=1}^{T} b(X_t)) \tag{15}$$

As studied in [1] and [12], LS-TD(λ) and RLS-TD(λ) algorithms converge to the same limit as conventional linear TD(λ) algorithms but have better data efficiency. Nevertheless, for Markov chains with nonlinear value functions, both linear TD(λ) and LS-TD(λ) algorithms may degrade significantly in approximation precision since improper selection of linear basis functions can not approximate nonlinear value functions with high accuracy.

III. Kernel Least-Squares TD Learning

In this Section, we will present the kernel least-squares TD learning algorithm, i.e., KLS-TD(λ), that introduces a Mercer kernel function to realize nonlinear least-squares TD learning in a kernel-based high-dimensional feature space *F*. The feature space *F* may have very high or even infinite dimensions and the feature mapping associated with *F* can be denoted as

$$\Phi : R^n \to F \tag{16}$$

where R^n is the original state space, and $\Phi(x)$ is the feature vector.

When a kernel function *k* is selected, according to the Mercer Theorem, there is a feature mapping Φ that satisfies

$$k(x, y) = <\Phi(x), \Phi(y)> = \Phi^T(x)\Phi(y) \tag{17}$$

Thus, all the inner products in feature spaces can be replaced by the computation of kernel functions, which will greatly simplify the computational problem caused by high-dimensional feature spaces.

Based on the above idea of kernel methods, the KLS-TD(λ) algorithm approximates the value function of a Markov chain as follows:

$$\widetilde{V}(x) = \Phi^T(x)W \tag{18}$$

where $W = [w_1, w_2, ..., w_{\dim(F)}]^T$ and $\Phi(x)$ are both column vectors.

According to the Representer Theorem in [13], the weight vector W in (18) can be represented by the weighted sum of state feature vectors:

$$W = \sum_{i=1}^{t} \Phi(x_i)\alpha_i \tag{19}$$

where x_i (i=1,2,…,m) are the observed states and α_i (i=1,2,…,t) are the corresponding coefficients.

For TD(λ) learning with eligibility traces, the least-squares regression equation determined by (12) is

$$E_0[z_i(\Phi^T(x_i) - \gamma\Phi^T(x_{i+1}))]\sum_{i=1}^{t}\Phi(x_i)\alpha_i^* = E_0[z_i r(x_i)] \tag{20}$$

where $z_i = \gamma\lambda z_{i-1} + \Phi(x_i)$.

Since the unbiased estimations of expectation $E_0[\cdot]$ can be obtained as

$$E_0[y] = \frac{1}{t}\sum_{i=1}^{t} y_i \tag{21}$$

where y_i (i=1,2,…,t) are observed data of random variable y, the least-squares regression equation (20) can be expressed as follows

$$\sum_{i=1}^{t-1}[z_i(\Phi^T(x_i) - \gamma\Phi^T(x_{i+1})]\sum_{i=1}^{t}\Phi(x_i)\alpha_i^* = \sum_{i=1}^{t} z_i r_i \tag{22}$$

Let

$$H_t = \begin{bmatrix} 1 & -\gamma & 0 & 0 & 0 \\ 0 & 1 & -\gamma & 0 & 0 \\ 0 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & 1 & -\gamma \end{bmatrix} \tag{23}$$

$$\Phi_t = [\Phi(x_1) \quad \Phi(x_2) \quad ... \quad \Phi(x_t)]^T \tag{24}$$

Then we can rewrite the regression equation (22) as follows

$$Z_t H_t \Phi_t \Phi_t^T \alpha_t = Z_t R_t \tag{25}$$

where

$$Z_t = [z_1 \quad \Phi(x_2) + \gamma\lambda z_1 \quad ... \quad \Phi(x_t) + \gamma\lambda z_{t-1}] \tag{26}$$

$$\vec{\alpha}_t = [\alpha_1, \alpha_2, ..., \alpha_t]^T \tag{27}$$

From (25), an equivalent form is as follows:

$$\Phi_t Z_t H_t \Phi_t \Phi_t^T \vec{\alpha}_t = \Phi_t Z_t R_t \tag{28}$$

Denote $\overline{Z}_t = \Phi_t Z_t$, then we have

$$\overline{Z}_t = [k_{1t}, k_{2t} + \gamma\lambda k_{1t}, ..., k_{tt} + \gamma\lambda k_{(t-1)t}] \tag{29}$$

where

$$k_{it} = [k(x_1, x_i), k(x_2, x_i), ..., k(x_t, x_i)]^T \tag{30}$$

Based on (17), we can define

$$\begin{aligned} K_t &= \Phi_t \Phi_t^T \\ &= \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & ... & k(x_1,x_t) \\ k(x_2,x_1) & k(x_2,x_2) & ... & k(x_2,x_t) \\ & & ... & \\ k(x_t,x_1) & k(x_t,x_2) & ... & k(x_t,x_t) \end{bmatrix} \end{aligned} \tag{31}$$

Then the least-squares TD regression equation becomes

$$\overline{Z}_t H_t K_t \vec{\alpha}_t = \overline{Z}_t R_t \tag{32}$$

The kernel least-squares solution is

$$\vec{\alpha}_t = (\overline{Z} H_t K_t)^{-1} \overline{Z}_t R_t \tag{33}$$

From (18) and (19), the estimated value function of Markov chains can be given by

$$\tilde{V}(x) = \sum_{i=1}^{t} \alpha_t k(x, x_i) \tag{34}$$

Therefore, we can present the kernel least-squares TD learning algorithm with eligibility traces, i.e., KLS-TD(λ), for ergodic Markov chains in the following. As discussed in [10], there are two classes of Markov chains, i.e., ergodic and absorbing Markov chains. In the later discussion, we will extend the kernel LS-TD(λ) learning algorithm from ergodic Markov chains to absorbing Markov chains.

Algorithm 1: KLS-TD(λ) for ergodic Markov chains

1: Given:
- A termination criterion for the algorithm;
- A kernel function $k(.,.)$ and the parameter λ;

2: Initialize:
(2.1) Let t=0.
(2.2) Set the initial state x_0 of the Markov chain.

3: Loop:
(3.1) For the current state x_t, observe the state transition from x_t to x_{t+1} and the reward $r(x_t, x_{t+1})$.
(3.2) Whenever updated estimations are desired, apply equations (23), (29) and (31) to compute $H_t, \overline{Z}_t, K_t$, respectively, use equation (33) and (34) to compute the coefficients and value function estimations.
(3.3) t=t+1.
until the termination criterion is satisfied.

The update equation (23) for H_t is only valid for ergodic Markov chains since for absorbing states in absorbing Markov chains, the update equation becomes

$$H_t = \begin{bmatrix} 1 & -\gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -\gamma & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\gamma \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{35}$$

where the states corresponding to the 3th and 5th rows are two absorbing states. When entering these absorbing states, the Markov chains are reset to some initial states.

Based on the above analysis, the KLS-TD(λ) algorithm for absorbing Markov chains can be described as follows.

Algorithm 2: KLS-TD(λ) for absorbing Markov chains

1: Given:
- A termination criterion for the algorithm;
- A kernel function $k(.,.)$ and the parameter λ;

2: Initialize:
(2.1) Let t=0.
(2.2) Set the initial state x_0.

3: Loop:
(3.1) For the current state x_t,
- If x_t is an absorbing state, using equation (35) to update H_t, $r(x_t)=r_T$, where r_T is the terminal reward.
- Otherwise, observe the state transition from x_t to x_{t+1} and the reward $r(x_t, x_{t+1})$, using equation (23) to update H_t.

(3.2) If x_t is an absorbing state, re-initialize the process by setting x_{t+1} to an initial state.
(3.3) Compute equation (33) and (34) whenever updated estimations are desired.
(3.4) $t=t+1$.

IV. Experimental Results

In this section, an illustrative example of value function prediction for Markov chains is given to show the effectiveness of the proposed KLS-TD(λ) algorithm. In the learning prediction experiments, a stochastic Markov chain with nonlinear value functions is considered, which is a variant of the Hop-World problem studied in [1].

As shown in Figure 1, the Hop-World problem is a 13-state Markov chain with an absorbing state. However, the value functions of the Markov chain in this example are nonlinear while the Hop-World problems discussed in [1] and [12] are linear with respect to some pre-selected basis functions.

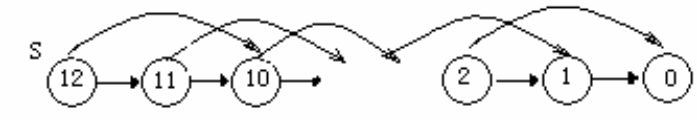


Fig.1. The non-linear Hop-World problem

In Fig. 1, state 12 is the initial state for each trajectory and state 0 is the absorbing state. Each non-absorbing state has two possible state transitions with transition probability 0.5. The true value function for state i ($0 \le i \le 12$) is defined as

$$V(i) = \frac{\sin(i/12.0)}{(i/12.0 + 0.1)} \tag{36}$$

In our experiments, the KLS-TD(λ) algorithm and conventional linear TD(λ) algorithms are used to solve the above value function prediction problem without knowing the model of the Markov chain. The basis function for the linear TD(λ) algorithms is chosen as $[x, x^2]$, where x is the normalized state of the Markov chain.

For the KLS-TD(λ) algorithm, RBF (Radius Basis Function) kernel functions are used and the width parameter for RBF kernels is manually selected as σ=0.2.In the experiments, a trial is defined as the period from the initial state 12 to the terminal state 0. The performance of the algorithms is evaluated by the averaged root mean squared (RMS) error of value-function predictions over all the 13 states. For each parameter setting, the RMS error is averaged over 20 independent Monte-Carlo runs.

Table 1 shows the RMS prediction errors of KLS-TD(λ) and conventional linear TD(λ) algorithms with manually optimized parameter settings. N is the sample number of the underlying Markov chain. From Table 1, it is clearly shown that the proposed KLS-TD(λ) has much better prediction accuracy than previous linear TD(λ) algorithms under different settings of parameter λ.

Table 1. Performance comparison of KLS-TD(λ) and TD(λ)

Algorithm/RMS	λ=0, N=100	λ=1, N=100	λ=0, N=200
KLS-TD(λ)	**0.036**	**0.065**	**0.006**
TD(λ)	0.385	0.263	0.293

V. Conclusion

This paper studies a new class of LS-TD learning algorithms based on kernel methods. By deriving the eligibility traces in the kernel-induced feature space, the KLS-TD(λ) algorithm is proposed for value function prediction both in ergodic and absorbing Markov chains. Compared to the previous linear TD(λ) and LS-TD(λ) algorithms, KLS-TD(λ) has significant values in nonlinear approximation abilities. Furthermore, KLS-TD(λ) is the first kernel-based TD learning algorithm with eligibility traces, which is different from the recently proposed kernel TD learning algorithms using Gaussian process models [3]. More theoretical and experimental analysis on the KLS-TD(λ) algorithm as well as extensions to learning control problems is our ongoing work.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grants 60303012 and 60234030, and Chinese Post-Doctor Science Foundation under Grant 200403500202.

References

[1] J. A. Boyan. "Technical update: least-squares temporal difference learning." *Machine Learning*, vol. 49, pp. 233-246, 2002.

[2] C. E. Rasmussen, M. Kuss, "Gaussian processes in reinforcement learning." In: *Proc. of the 2003 Neural Information Processing Systems*, NIPS-2003, 2003.

[3] Y. Engel, S. Mannor and R. Meir, "Bayes meets bellman: the Gaussian Process approach to temporal difference learning." In: *Proc. of the 20th International Conference on Machine Learning*, ICML-03 , 2003.

[4] L.P. Kaelbling, M.L. Littman, A.W. Moore, "Reinforcement learning: a survey." *Journal of Artificial Intelligence Research*, vol. 4, pp.237-285, 1996.

[5] B. Schölkopf, A.: Smola, "Learning with Kernels." Cambridge, MA: MIT Press, 2002.

[6] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A.J. Smola, "Input space vs feature space in kernel-based algorithms." *IEEE Transactions on Neural Networks*, vol.10, no.3, pp.1000 – 1017, 1999.

[7] R. Sutton, A. Barto, "Reinforcement learning, an introduction." Cambridge MA, MIT Press, 1998.

[8] R. Sutton, "Learning to predict by the method of temporal differences." *Machine Learning*, vol.3, no.1, pp.9-44,1988.

[9] V. Tadić, "On the convergence of temporal-difference learning with linear function approximation," *Machine Learning*, vol.42, no.3, pp.241-267, 2001.

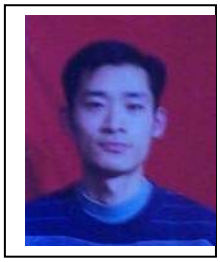
[10] J.N. Tsitsiklis, B.V. Roy, "An analysis of temporal difference learning with function approximation," *IEEE Transactions on Automatic Control,* vol.42, no.5, pp. 674-690, 1997.

[11] V. Vapnik, "Statistical Learning Theory," NewYork:Wiley Interscience, 1998.

[12] X. Xu, H. He, D.W Hu., "Efficient reinforcement learning using recursive least-squares methods," *Journal of Artificial Intelligence Research*, vol. 16, pp. 259–292, 2002.

[13] G. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, pp. 82 – 95, 1971.

[14] F. R. Bach, M. I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol.3, pp.1-48, 2002.



Dr. Xin Xu received the Bachelor degree in Electrical Engineering from Department of Automatic Control, National University of Defense Technology (NUDT), P.R.China, in 1996. In 2002, he obtained the PhD degree in Electrical Engineering from College of Mechantronics Engineering and Automation (CMEA), NUDT, P.R.China. From 2003 to 2004, he was a post-doctor fellow in School of Computer, NUDT. Now he is an associate professor at CMEA, NUDT, P.R.China. His research interests include reinforcement learning, data mining, intelligent control, autonomic Grid computing and computer security.

Dr. Tao Xie received his PhD degree in Aerospace Engineering from Department of Aerospace Engineering, National University of Defense Technology (NUDT), P.R.China, in 1998. From 1999 to 2001, he was a post-doctor fellow in School of Computer, NUDT, P.R.China. Now he is an associate professor in School of Computer, NUDT. His research interests include machine learning, evolutionary computation and computer security.