FUZZY Q-LEARNING IN SVD REDUCED DYNAMIC STATE-SPACE

SZILVESZTER KOVÁCS*
Department of Information Technology, University of Miskolc,
Miskolc-Egyetemváros, Miskolc, H-3515, Hungary
szkovacs@iit.uni-miskolc.hu

Péter BARANYI*
Department of Telecommunication and Telematics, Technical University of Budapest,
Pázmány Péter sétány 1/d, B223, Budapest, H-1117, Hungary
baranyi@ttt.bme.hu

[Received May 2002 and accepted April 2003]

*Intelligent Integrated Systems Japanese Hungarian Laboratory,
Budapest University of Technology and Economics, Hungary

Abstract. Reinforcement Learning (RL) methods, surviving the control difficulties of the unknown environment, are gaining more and more popularity recently in the autonomous robotics community. One of the possible difficulties of the reinforcement learning applications in complex situations is the huge size of the state-value- or action-value-function representation [17]. The case of continuous environment (continuous valued) reinforcement learning could be even complicated, as the state-value- or action-value-functions are turning into continuous functions. In this paper we suggest a way for tackling these difficulties by the application of SVD (Singular Value Decomposition) methods [6], [19], [20].

Keywords: Reinforcement Learning, Fuzzy Q-Learning, Singular Value Decomposition

1. Introduction

Reinforcement learning methods are trial-and-error style learning methods adapting dynamic environment through incremental iteration. The principal ideas of reinforcement learning methods, the dynamical system state and the idea of "optimal return" or "value" function are inherited from optimal control and dynamic programming [7]. One common goal of the reinforcement learning strategies is to find an optimal policy by building the state-value- or action-value-function [17]. The state-value-function $V^{\pi}(s)$, is a function of the expected return (a function of the cumulative reinforcements), related to a given state $s \in S$ as a starting point, following a given policy π. Where the states of the learning agent are observable and the reinforcements (or rewards) are given by the environment. These rewards are the expression of the goal of the learning agent as a kind of evaluation follows the recent action (in spite of the instructive manner of error feedback based approximation techniques, like the gradient descent training). The policy is the description of the agent behaviour, in the form of mapping between the agent states and the corresponding suitable actions. The action-value function $Q^{\pi}(s,a)$ is a function of the expected return, in case of taking action $a \in A_s$ in a given state s, and then following a given policy π. Having the action-value-function, the optimal (greedy) policy, which always takes the optimal (the greatest estimated value) action in every state, can be constructed as [17]:

$$\pi(s) = \arg\max_{a \in A_s} Q^{\pi}(s,a) \cdot \tag{1}$$

(Where the function arg is standing for the indexes of the set of possible actions.)

Namely for estimating the optimal policy, the action-value function $Q^{\pi}(s,a)$ is needed to be approximated. In discrete environment (discrete states and discrete actions) it means, that at least $\sum_{s \in S} \|A_s\|$ element must be handled. (Where $\|A_s\|$ is the cardinality of the set of possible actions in state s.) Having a complex task to adapt, both the number of possible states and the number of the possible actions could be an extremely high value.

1.1 Reinforcement Learning in Continuous Environment

To implement reinforcement learning in continuous environment (continuous valued states and actions), function approximation methods are widely used. Many of these methods are applying tailing or partitioning strategies to handle the continuous state and action spaces in the similar manner as it was done in the discrete case [17]. One of the difficulties of building an appropriate partition structure (the way of partitioning the continuous universe) is the anonymity of the action-value-function structure. Applying fine resolution in the partition leads to high number of states, while coarse partitions could yield imprecise or unadaptable system. Handling high number of states also leads to high computational costs, which could be also unacceptable in many real time applications

There are many methods in the literature for applying fuzzy techniques in reinforcement learning (e.g. for "Fuzzy Q-Learning" [1], [8], [9], [11], [12]). One of the main reasons of their application beyond the simplicity of expressing priory knowledge in the form of fuzzy rules is the universal approximation property [10], [22] of the fuzzy inference. It means that any kind of function can be approximated in an acceptable level, even if the analytic structure of the function is unknown. Despite of this useful property, the use of fuzzy inference could be strictly limited in time-consuming reinforcement learning by its complexity problems [13], because of the exponential complexity problem of fuzzy rule bases [5], [20]. Fuzzy logic inference systems are suffering from exponentially growing computational complexity in respect to their approximation property. This difficulty comes from two inevitable facts. The first is that the most adopted fuzzy inference techniques do not hold the universal approximation property, if the numbers of antecedent sets are limited, as stated by Tikk in [18]. Furthermore, their explicit functions are sparse in the approximation function space. This fact inspires to increase the density, the number of antecedents in pursuit of gaining a good approximation, which, however, may soon lead to a conflict with the computational capacity available for the implementation, since the increasing number of antecedents explodes the computational requirement. The latter is the second fact and stated by *Kóczy* et al. in [13]. The effect of this contradiction is gained by the lack of a mathematical framework capable of estimating the necessary minimal number of antecedent sets. Therefore a heuristic setting of the number of antecedent sets is applied, which usually overestimates, in order to be on the safe side, the necessary number of antecedents resulting in an unnecessarily high computational cost. E.g. the structurally different Fuzzy Q-Learning method implementations introduced [8], [9], [11] and [12] are sharing the same concept of fixed, predefined fuzzy antecedent partitions, for state representation. One possible solution for this problem is suggested in [1]. By introducing "Adaptive State Partitions", an incremental fuzzy clustering of the observed state transitions. This method can lead to a better partition than the simple heuristic, by finding the best fitting one in respect to the minimal squared error, but still has the problem of limited approximation property inherited from the limited number of antecedent fuzzy sets.

Another promising solution, as a new topic in fuzzy theory, is the application of fuzzy rule base complexity reduction techniques.

1.2 Fuzzy rule base complexity reduction

The main goal of introducing fuzzy rule base complexity reduction techniques in reinforcement learning is enhancing the universal approximation property of the fuzzy inference by extending the number of antecedent sets while the computational complexity is kept relatively low. SVD based fuzzy approximation technique was initialized in 1997 by Yam [19], which directly finds a minimal rule-base from sampled values. Shortly after, this concept was introduced as SVD fuzzy rule base reduction and structure decomposition in [2], [20]. Its key idea is conducting SVD of the consequents and generating proper linear combinations of the original membership functions to form new ones for the reduced set. An extension of [21] to multi-dimensional cases may also be conducted in a similar fashion as the Higher Order SVD (HOSVD) reduction technique proposed in [5], [19], [20]. Further developments of SVD based fuzzy reduction are proposed in [3], [5] and its extension to the generalized inference forms are proposed in [14], [15], [16].

The key idea of using SVD in complexity reduction is that the singular values can be applied to decompose a given system and indicate the degree of significance of the decomposed parts. Reduction is conceptually obtained by the truncation of those parts, which have weak or no contribution at all to the output, according to the assigned singular values. This advantageous feature of SVD is used in this paper for enhancing the universal approximation property of the fuzzy inference by extending the number of antecedent sets while the computational complexity is kept relatively low. The complexity and its reduction is discussed in regard of the number of rules, which result simplicity in operating with the rules, in reinforcement learning methods.

On the other hand, as one of the natural problems of any complexity reduction technique, the adaptivity property of the reduced approximation algorithm becomes highly restricted. Since the crucial concept of the Fuzzy Q-learning is based on the adaptivity of the action-value function this paper is aimed propose to adopt an algorithm [6] capable of embedding new approximation points into the reduced approximation while the calculation cost is kept (where the calculation cost could be defined in the terms of the number of product operations done during the calculation).

2. Fuzzy Q-Learning

For introducing a possible way of application of SVD complexity reduction techniques in Fuzzy Reinforcement Learning, a simple direct (model free) reinforcement learning method, the Fuzzy Q-Learning, was chosen.

The goal of the Q-learning is to find the fixed-point solution Q of the Bellman Equation [7] through iteration. In discrete environment *Q-Learning* [23], the action-value-function is approximated by the following iteration:

$$Q_{i,u} \approx \widetilde{Q}_{i,u}^{k+1} = \widetilde{Q}_{i,u}^{k} + \Delta\widetilde{Q}_{i,u}^{k+1} = \widetilde{Q}_{i,u}^{k} + \alpha_{i,u}^{k} \cdot \left(g_{i,u,j} + \gamma \cdot \max_{v \in U} \widetilde{Q}_{j,v}^{k+1} - \widetilde{Q}_{i,u}^{k}\right), \quad \forall i \in I, \forall u \in U \qquad (2)$$

where $\widetilde{Q}_{i,u}^{k+1}$ is the $k+1$ iteration of the action-value taking the u^{th} action A_u in the i^{th} state S_i, S_j is the new (j^{th}) observed state, $g_{i,u,j}$ is the observed reward completing the $S_i \to S_j$ state-transition, γ is the discount factor and $\alpha_{i,u}^{k} \in [0,1]$ is the step size parameter (which can change during the iteration steps), I is the set of the discrete possible states and U is the set of the discrete possible actions.

For applying this iteration to continuous environment by adopting fuzzy inference (Fuzzy Q-Learning), there are many solutions exist in the literature [1], [8], [9], [11], [12]. Having only demonstrational purposes, in this paper one of the simplest one, the order-0 Takagi-Sugeno Fuzzy Inference based Fuzzy Q-Learning is studied (a slightly modified, simplified version of the Fuzzy Q-Learning introduced in [1] and [12]). This case, for characterising the value function $Q(s,a)$ in continuous state-action space, the order-0 Takagi-Sugeno Fuzzy Inference System approximation $\widetilde{Q}(s,a)$ is adapted in the following manner:

$$\textbf{If } s \textbf{ is } S_i \textbf{ And } a \textbf{ is } A_u \textbf{ Then } \widetilde{Q}(s,a) = Q_{i,u},\ i \in I, u \in U, \qquad (3)$$

where S_i is the label of the i^{th} membership function of the n dimensional state space, A_u is the label of the u^{th} membership function of the one dimensional action space, $Q_{i,u}$ is the singleton conclusion and $\widetilde{Q}(s,a)$ is the approximated continuous state-action-value function. Having the approximated state-action-value function $\widetilde{Q}(s,a)$, the optimal policy can be constructed by function (1).

Setting up the antecedent fuzzy partitions to be *Ruspini partitions*, the order-0 Takagi-Sugeno Fuzzy Inference forms the following approximation function:

$$\widetilde{Q}(s,a) = \sum_{i_1,i_2,\ldots,i_N,u}^{I_1,I_2,\ldots,I_N,U} \prod_{n=1}^{N} \mu_{i_n,n}(s_n) \cdot \mu_u(a) \cdot q_{i_1 i_2 \ldots i_N u} \qquad (4)$$

where $\widetilde{Q}(s,a)$ is the approximated state-action-value function $\mu_{i_n,n}(s_n)$ is the membership value of the i_n $^{\text{th}}$ state antecedent fuzzy set at the n $^{\text{th}}$ dimension of the N dimensional state antecedent universe at the state observation s_n, $\mu_u(a)$ is the membership value of the u $^{\text{th}}$ action antecedent fuzzy set of the one dimensional action antecedent universe at the action selection a and $q_{i_1 i_2 \ldots i_N u}$ is the value of the singleton conclusion of the $i_1,i_2,\ldots,i_N,u$ $^{\text{-th}}$ fuzzy rule. (A fuzzy partition is a *Ruspini partition* if the sum of the membership values of the member sets of the partition is equal to one for the entire universe of discourse: $\sum_{i}^{I}\mu_i(x)=1$ for $\forall x \in X$, where $\mu_i(x)$ is the membership function of the i^{th} fuzzy set of the I element fuzzy partition on the universe of discourse X – see e.g. on Fig.1.a)

Applying the approximation formula of the Q-learning (2) for adjusting the singleton conclusions in (4), leads to the following function:

$$q^{k+1}_{i_1 i_2 \ldots i_N u} = q^{k}_{i_1 i_2 \ldots i_N u} + \prod_{n=1}^{N} \mu_{i_n,n}(s_n)\cdot\mu_u(a)\cdot\Delta\widetilde{Q}^{k+1}_{i,u} \tag{5}$$

$$q^{k+1}_{i_1 i_2 \ldots i_N u} = q^{k}_{i_1 i_2 \ldots i_N u} + \prod_{n=1}^{N} \mu_{i_n,n}(s_n)\cdot\mu_u(a)\,\alpha^{k}_{i,u}\cdot\left(g_{i,u,j} + \gamma\cdot\max_{v\in U}\widetilde{Q}^{k+1}_{j,v} - \widetilde{Q}^{k}_{i,u}\right)$$

where $q^{k+1}_{i_1 i_2 \ldots i_N u}$ is the $k+1$ iteration of the singleton conclusion of the $i_1 i_2 \ldots i_N u$ th fuzzy rule taking action A_u in state S_i, S_j is the new observed state, $g_{i,u,j}$ is the observed reward completing the $S_i \to S_j$ state-transition, γ is the discount factor and $\alpha^{k}_{i,u} \in [0,1]$ is the step size parameter. The $\max_{v\in U}\widetilde{Q}^{k+1}_{j,v}$ and $\widetilde{Q}^{k}_{i,u}$ action-values can be approximated by equation (4).

3. Dynamic Partition Allocation

The next problematic question of the Fuzzy Reinforcement Learning, as it was introduced in Section 1, is the proper way of building the fuzzy partitions. The methods sharing the concept of fixed, predefined fuzzy partitions, like [8], [9], [11] and [12] are facing the following question: More detailed partitions are yielding exponentially growing state spaces (rule base sizes), elongating the adaptation time, and dramatically increasing the computational resource demand, while less detailed partitions (containing only a few member fuzzy sets) could cause high approximation error, or unadaptable situation. One possible solution for this problem is suggested in [1]. By introducing “Adaptive State Partitions”, an incremental fuzzy clustering of the observed state transitions. This method can lead to a better partition than the simple heuristic, by finding the best fitting one in respect to the minimal squared error, but still has the problem of limited approximation property inherited from the limited number of antecedent fuzzy sets.

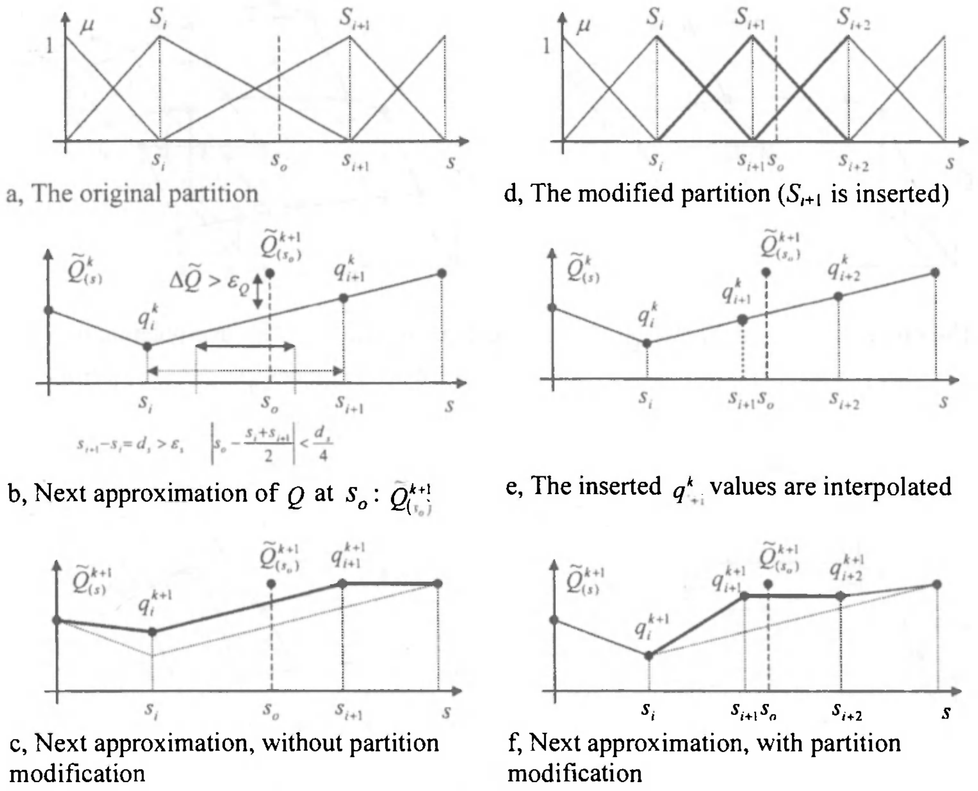
In this paper another dynamic partition allocation method is suggested, which is instead of adjusting the sets of the fuzzy partition, simply increase the number of the fuzzy sets by inserting new sets in the required positions. The main idea is very simple (see Fig.1. for an example). Initially a minimal sized (e.g. 2-3 sets only) Ruspini partition built up triangular shaped fuzzy sets on all the antecedent universes (see Fig.1.a). In the case when the action-value function update (5) is high (e.g. greater than a preset limit ε_Q: $\Delta\widetilde{Q} > \varepsilon_Q$), and the partition is not too dense already at that point (e.g. the distance of the cores of the surrounding fuzzy sets (d_s) is greater than a preset limit ε_s: $s_{i+1}-s_i=d_s>\varepsilon_s$), and the actual state-action point (s_o) is far from the existing partition members (e.g. the actual state-action point is closer to the middle than one of the surrounding fuzzy sets cores:

$\left|s_o - \frac{s_i+s_{i+1}}{2}\right| < \frac{d_s}{4}$) – see e.g. on Fig.1.b, then a new fuzzy state is inserted among the existing partition to increase the resolution (e.g. $s_{k+1}=s_k$, $\forall k > i$, $s_{i+1}=\frac{s_i+s_{i+2}}{2}$) – see e.g. on Fig.1.d.

If the update value is relatively low ($\Delta\tilde{Q} \le \varepsilon_Q$, see e.g. Fig.2.), or the actual state-action point is close to the existing partition members ($\left|s_o - \frac{s_i+s_{i+1}}{2}\right| \ge \frac{d_s}{4}$, see e.g. Fig.3.), than the partition is staying unchanged. The state insertion is done in every state dimensions separately (in multidimensional case it means an insertion of a hyperplane), by interpolating the inserted values from the neighbouring ones (see Fig.1.e and Fig.4. as a two dimensional example). Having the new state plane inserted in every required dimension, the value update is done regarding to the Fuzzy Q-Learning method as it was introduced in Section 2, by the equation (5). (See e.g. on Fig.1.c, Fig.1.d, or Fig.4.d.)

The proposed dynamic partition allocation method has the property of local step-by-step refinement in a manner very similar to the binary search. It can locate the radical positions of the value action function with the precision of $d_s^{i+k} = \frac{d_s^i}{2^k}$ in k steps (where d_s^i is the starting precision).

The main problem of the proposed simple dynamic partition allocation method is the non-decreasing adaptation manner of the antecedent fuzzy partitions. In some situation, it could mean rapidly increasing partition sizes in the sense of the number of the component fuzzy sets. Moreover, these cases also lead rapidly growing, or at least non-decreasing computational resource demand.



a, The original partition

d, The modified partition (S_{i+1} is inserted)

b, Next approximation of Q at s_o: $\tilde{Q}^{k+1}_{(s_o)}$

e, The inserted q^k_{i+1} values are interpolated

c, Next approximation, without partition modification

f, Next approximation, with partition modification

Fig. 1. The proposed dynamic partition allocation method.

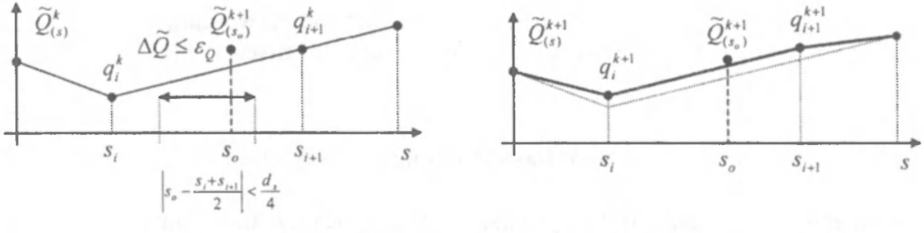

$\tilde{Q}^k_{(s)}$
$\tilde{Q}^{k+1}_{(s_o)}$
$\Delta\tilde{Q} \le \varepsilon_Q$
q^k_i
q^k_{i+1}
s_i
s_o
s_{i+1}
s
$\left|s_o-\frac{s_i+s_{i+1}}{2}\right|<\frac{d_s}{4}$
$\tilde{Q}^{k+1}_{(s)}$
q^{k+1}_i
q^{k+1}_{i+1}

Fig. 2. The action-value function update is relatively low.

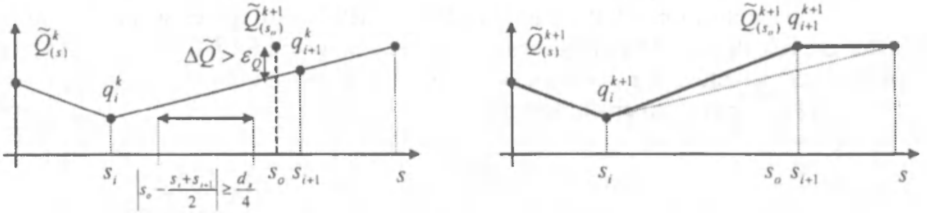
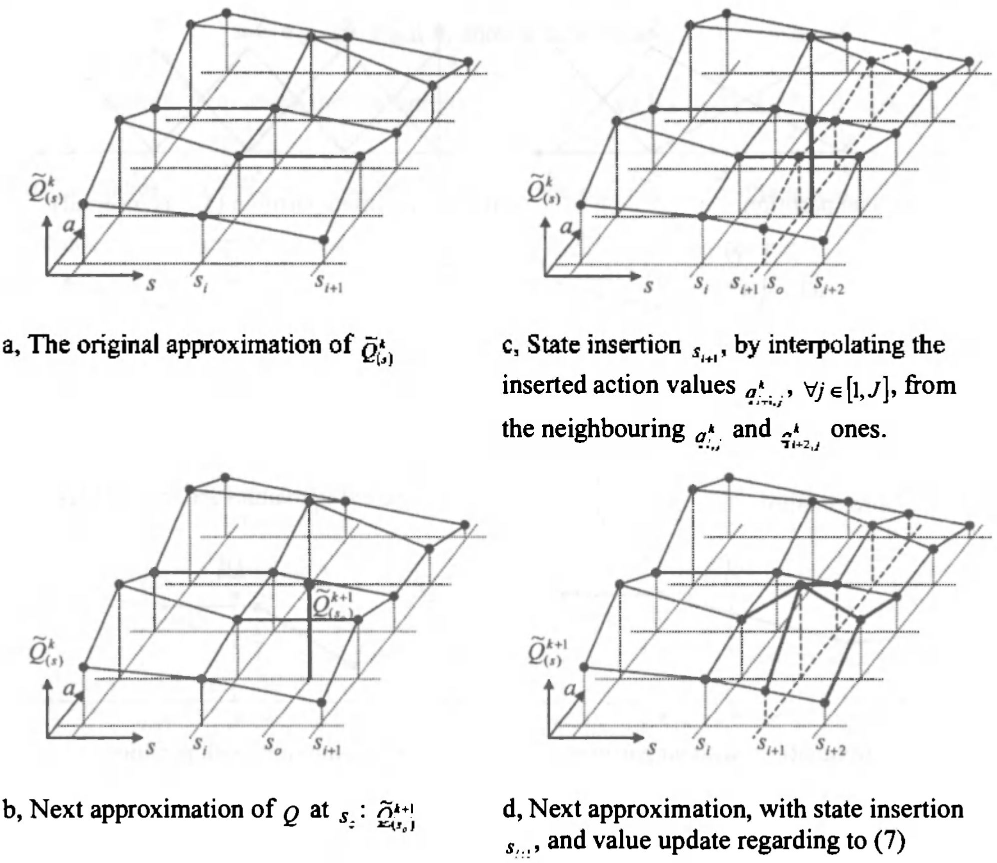

$\tilde{Q}^k_{(s)}$
$\tilde{Q}^{k+1}_{(s_o)}$
$\Delta\tilde{Q} > \varepsilon_Q$
q^k_i
q^k_{i+1}
s_i
$\left|s_o-\frac{s_i+s_{i+1}}{2}\right|\ge\frac{d_s}{4}$
s_o
s_{i+1}
s
$\tilde{Q}^{k+1}_{(s)}$
q^{k+1}_i
q^{k+1}_{i+1}

Fig. 3. The actual state-action point is close to the existing partition members.



a, The original approximation of $\tilde{Q}^k_{(s)}$

c, State insertion s_{i+1}, by interpolating the inserted action values $a^k_{i+1,j}$, $\forall j \in [1,J]$, from the neighbouring $a^k_{i,j}$ and $a^k_{i+2,j}$ ones.

b, Next approximation of Q at s_o: $\tilde{Q}^{k+1}_{(s_o)}$

d, Next approximation, with state insertion s_{i+1}, and value update regarding to (7)

Fig. 4. The proposed dynamic partition allocation in two-dimensional (single state and action) antecedent case.

4. SVD based Complexity Reduction

For retaining the benefits of the dynamic partition allocation and maintaining the overall computational resource demand low, in this paper, the adoption of Higher Order SVD [5] based fuzzy rule base complexity reduction techniques and its fast adaptation method is suggested. The application of the fast adaptation method [6] gives a simple way for increasing the rule density of a rule base stored in a compressed form directly. Providing an economic sized structure for handling continuously increasing and varying rule bases, which is so typical in reinforcement learning.

4.1. SVD Based Fuzzy rule base complexity reduction

The essential idea of using SVD in complexity reduction is that the singular values can be applied to decompose a given system and indicate the degree of significance of the decomposed parts. Reduction is conceptually obtained by the truncation of those parts, which have weak or no contribution at all to the output, according to the assigned singular values. This advantageous feature of SVD is used in this paper for enhancing the universal approximation property of the fuzzy inference by extending the number of antecedent sets while the computational complexity is kept relatively low. The complexity and its reduction is discussed in regard of the number of rules, which result simplicity in operating with the rules, in reinforcement learning methods.

Definitions:

N-mode matrix of a given tensor ***A*:** Assume an N-th order tensor $A\in\Re^{I_1\times I_2\times\dots}$ The n-mode matrix $A_{(n)}\in\Re^{I_n\times J}$, $J=\prod_k I_l$ contains all the vectors in the n-th dimension of the tensor A. The ordering of the vectors is arbitrary, this ordering shall, however, be consistently used later on. $(\mathbf{A}_{(n)})_j$ is called an j-th n-mode vector. Note that any matrix of which the columns are given by n-mode vectors $(\mathbf{A}_{(n)})_j$ can evidently be restored to be the tensor A. (See a three dimensional example on Fig.5.)

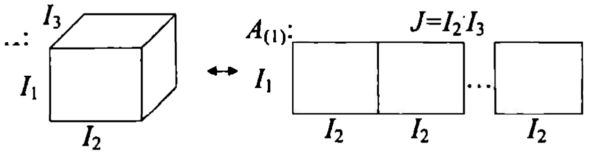


Fig. 5. N-mode matrix of a tensor (three dimensional example).

N-mode matrix-tensor product: The n-mode product of a tensor $A\in\Re^{I_1\times I_2\times\ldots\times I_N}$ by a matrix $\mathbf{U}\in\Re^{J\times I_n}$, denoted by $A\times_n\mathbf{U}$ is an $(I_1\times I_2\times\ldots\times I_{n-1}\times J\times I_{n+1}\times\ldots\times I_N)$-tensor of which the entries are given by $A\times_n\mathbf{U}=B$, where $B_{(n)}=\mathbf{U}\cdot A_{(n)}$. Let $A\bigotimes_{n=1}^{N}\mathbf{U}_n$ stand for $A\times_1\mathbf{U}_1\times_2\mathbf{U}_2\ldots\times_N\mathbf{U}_N$.

N-th Order SVD or Higher Order SVD (HOSVD):

Every tensor $A\in\Re^{I_1\times I_2\times\ldots\times I_N}$ can be written as the product $A=S\bigotimes_{n=1}^{N}\mathbf{U}_n$, in which $\mathbf{U}_n=[\mathbf{u}_{1,n}\quad \mathbf{u}_{2,n}\qquad \mathbf{u}_{I_N,n}]$ is a unitary $(I_N\times I_N)$-matrix called n-mode singular matrix. Tensor $S\in\Re^{I_1\times I_2\times\ldots\times I_N}$ of which the subtensors $S_{i_n=\alpha}$ have the properties of all-orthogonality (two subtensors $S_{i_n=\alpha}$ and $S_{i_n=\beta}$ are orthogonal (their scalar product equals 0) for all possible values of n,α and β: $\langle S_{i_n=\alpha}, S_{i_n=\beta}\rangle = 0$ when $\alpha \neq \beta$ (where $\langle A,B\rangle = \sum_{i_1}\sum_{i_2}\cdots\sum_{i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots}$ is the scalar product of two tensors $A,B \in \Re^{I_1\times I_2\times\ldots\times I_N}$)) and ordering: $\|S_{i_n=1}\| \geq \|S_{i_n=2}\| \geq \ldots \geq \|S_{i_n=I_n}\| \geq 0$ for all possible values of *n* (where $\|A\| = \sqrt{\langle A,A\rangle}$ is the Frobenius-norm of a tensor *A*). (See a three dimensional example on Fig.6.) See detailed discussion and notation of matrix SVD and Higher Order SVD (HOSVD) in [5].

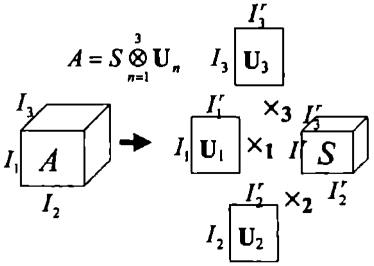


Fig. 6. N-th Order SVD or Higher Order SVD (three dimensional example).

Exact / non-exact reduction

Assume an *N*-th order tensor $A \in \Re^{I_1\times I_2\times\ldots\times I_N}$. **Exact** reduced form $A = A^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n$, where "*r*" denotes "reduced", is defined by the tensor $A^r \in \Re^{I_1^r\times I_2^r\times\ldots}$ and basis matrices $\mathbf{U}_n \in \Re^{I_n\times I_n^r}$, $I_n^r \leq I_n$, $n = 1,2,\ldots,N$ which are the result of HOSVD, where only zero singular values and the corresponding singular vectors are discarded. **Non-exact** reduced form $\hat{A} = A^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n$, is obtained if not only zero singular values and the corresponding singular vectors are discarded.

SVD Based Fuzzy Rule Base Complexity Reduction

The explicit formula of the order-0 Takagi-Sugeno Fuzzy Inference method: (e.g. (4)) Assume an N-variable fuzzy rule base given by: antecedent fuzzy sets $\mu_{i_n,n}(x_n)$ defined on input universe X_n and all combination of the antecedents corresponds to one consequent fuzzy set defined on output universe Y These relations are expressed by rules in the form of

$$\textbf{If } \mu_{i_1,1}(x_1) \textbf{ And } \mu_{i_2,2}(x_2) \textbf{ And } \ldots \textbf{ And } \mu_{i_N,N}(x_N) \textbf{ Then } y = \beta_{i_1 i_2 \ldots} \quad . \tag{6}$$

Singleton consequent fuzzy sets $\beta_{i_1 i_2 \ldots}$ are defined by their location $b_{i_1 i_2 \ldots i_N}$ Setting up the antecedent fuzzy partitions to be *Ruspini partitions*, the explicit formula of the inference technique is (for more detailed explanation see [20]):

$$f(x_1, x_2, \ldots, x_N) = \sum_{i_1, i_2, \cdots, i_N}^{I_1, I_2, \ldots, I_N} \prod_{n=1}^{N} \mu_{i_n, n}(x_n) b_{i_1 i_2 \ldots} \tag{7}$$

SVD Based Fuzzy Rule Base Complexity Reduction: The formula of the order-0 Takagi-Sugeno Fuzzy Inference method (7) can be equivalently written in tensor product form as: $f(x_1, x_2, \ldots, x_N) = B \overset{N}{\underset{n=1}{\otimes}} \mathbf{m}_n$, where the tensor $B \in \Re^{I_1 \times I_2 \ldots}$ and the vector $\mathbf{m}_n$ respectively contain elements $b_{i_1 i_2 \ldots i_N}$ and $\mu_{i_n, n}(x_n)$. This reduction can be conceptually obtained by reducing the size of the tensor B via Higher Order SVD (HOSVD). See more detailed description in [5], [19], [20]. The SVD based fuzzy rule base reduction transforms the structure of equation (7) to the form of:

$$f(x_1, x_2, \ldots, x_N) = \sum_{i_1, i_2, \ldots, i_N}^{I_1^r, J_2^r, \ldots, I_3^r} \prod_{n=1}^{N} \mu_{i_n, n}^r(x_n) b_{i_1}^r \tag{8}$$

where $\forall n : J_n^r \leq J_n$ is obtained as the main essence of the reduction.

The reduced form (8) of (7) is obtained via HOSVD capable of decomposing B into $B = B^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n$ Having $B^r \in \Re^{I_1^r \times I_2^r \ldots}$ and its singular vectors the reduced form is determined as: $f(x_1, x_2, \ldots, x_N) = B^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{m}_n^r$, where $\mathbf{m}_n^r = \mathbf{m}_n \mathbf{U}_n$.

4.2. Adaptation of SVD based Approximation

According to the previous sections the crucial concept of the reinforcement learning is based on the adaptivity of the action-value function. It was also concluded in the previous sections that the approximation accuracy of the action-value function is highly restricted by its computational complexity. For instance, the increase of the density of the approximation grid on Fig. 4 improves the approximation accuracy. Each learning step may insert a new gridline into dimension *S*. However, this may lead to a high complexity soon, since adding a grid-line exponentially increases the number of the approximation grid. Therefore, it is highly desired to reduce the complexity of the action-value function. However, one should note that a natural problem of typical complexity reduction is that it decreases the adaptivity property with the complexity in general. This disadvantage is also true for SVD based reduction technique discussed in the previous section. This implies that executing the SVD based reduction on the action-value function would destroy the effectiveness of the whole learning concept. Therefore, this paper proposes to utilize a “fast adaptation” technique, introduced in [6], capable of keeping the action-value function in SVD based complexity reduced form, but also capable of adapting the function without considerable computational effort. This method let us directly adapt the complexity compressed action-function over any specified point of the learning space and add new approximation grid-lines, see Fig. 4. The key idea is that the fast adaptation technique transforms the given new grid-lines and corresponding values into the complexity reduced space of the action-function where the adaptation can immediately be done. The ability of embedding new approximation points provides the practical applicability of the proposed dynamic partition allocation method discussed in the previous section. Therefore, the application of the fast adaptation method in the proposed reinforcement learning structure is twofold. On one side, it helps the dynamic partition allocation by increasing the grid density. On the other side, by the replacement of the previously fetched and modified values serves the adaptation of the approximated action-value function.

Let the goal of the adaptation technique be specified in the followings: The goal is to insert a set of new rules included in A into the existing rule base B. Assume that the rule base B is already reduced into B^r The new rules contained in A should directly be inserted into B^r Directly means that without decompressing B^r to B. Assume that the size of B^r is fixed, it must not be increased with the adaptation. As a matter of fact, there may be such rules in A which require the increase of the size of B^r The fast adaptation technique discards these rules and inserts only those ones collected in A' which do not increase the size of B. In order to insert as much rules as possible the fast adaptation technique has a further option. Subject to a given threshold ∇, it is capable of modifying the discarded rules in order to insert them to B^r If the rule bases are represented by tensors as discussed in the previous section then the adaptation can be defined as: only those sub-tensors A' of A are embedded into B^r, which are linearly dependent from B^r [6]. An important advantage of the fast adaptation is that no SVD is needed during inserting the new rules.

***N* mode fast adaptation** *[6]:*

"*N* mode" means in the present case that the rules, to be inserting, have new antecedents on dimension *N*. Namely, this means that the number the approximation grid-lines under the function, see Fig. 4, is increased in dimension *N*.

Assume a reduced rule base defined by tensor $B^r \in \Re^{J_1^r \times J_2^r \times \dots}$ and its corresponding matrices $\mathbf{Z}_n \in \Re^{J_n \times J_n^r}$ resulted from B by HOSVD as:

$$B = B^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{Z}_n \tag{9}$$

Furthermore, let $A \in \Re^{J_1 \times J_2 \times \dots}$ be given, that has the same size as B except in the n-th dimension where I may differ from J_n. Let us have a brief digression here and explain A and B on Fig. 4. Tensor B, which is a matrix in the case of Fig. 4, consists of the values of the function over the grid-points defined by the orthogonal grid-lines located at values s. Tensor A, which is also a matrix in the present case, contains the values over the grid-points and the new grid-lines located on dimension N, that is S on Fig. 4. We can observe that the size of A is equivalent to the size of B except on that dimensions where the new gridlines are defined. If B is compressed to B^r then we do not have this matrix point-wise equivalency to the rectangular grid. In this case the inserting of the new grid-lines and their corresponding new approximation points is not trivial.

The localized error interval of the adaptation is defined by ∇ Localised means that ∇ is a tensor whose elements are intervals and assigned to the grid-points like in the case of A and B. It defines the acceptable varying of the function over the grid-points. The goal is to determine the reduced form E' of extended rule base E, defined by tensor $E' = \begin{bmatrix} B & A' \end{bmatrix}_n$. In the case of Fig. 4 E is a matrix and contains the values of the function over all the new and the original grid-lines. E' contains the selected n mode sub-tensors of E according to the given interval ∇ In the case of Fig. 4 E' contains the values over all the original grid-points and over those grid-lines, which are accepted to be inserted. Only those grid-lines are accepted which do not increase the size of B^r, or whose modified values are still in the intervals of ∇ Thus

$$\hat{E}' = \left(B^r \overset{N}{\underset{k=1}{\otimes}} \mathbf{Z}_k \right) \times_n \mathbf{U}, \tag{10}$$

and $A \in \Re^{J_1 \times J_2 \times \ldots J_{n-1} \times I' \times J_{n+1} \times \ldots \times J_N}$ contains the selected n mode sub-tensors of A and let the corresponding sub-tensors $T'_{\text{min/max}}$ be selected from the corresponding $T_{\text{min/max}}$ which define the maximal and minimal values of the elements of ∇ For brevity let $\nabla' = [T'_{\text{min}} \quad T'_{\text{max}}]$. $\mathbf{U} = \begin{bmatrix} \mathbf{Z}_n & \mathbf{V} \end{bmatrix} \in \Re^{(J_n + I') \times J_n^r}$, $I' \leq I$, where $\mathbf{V}$ is determined to satisfy (10) subject to $\hat{E}' - E' \in_I \nabla'$. $\in_I$ means that the elements of tensor $\hat{E}' - E'$ is in the interval defined by the corresponding elements of ∇' (the bound of the intervals are defined by the corresponding elements of T_{min} and T_{max}).

Inserting new gridlines on all dimensions is done in the same way. This means that the desnity of the hyper rectangular approximation grid can be incerased by the above algorithm even in case when the values assigned to grid are compressed into a reduced form where there is no structure which can be localised according to the grid-points. The more detailed description of the fast adaptation algorithm is given in [4] and [6].

5. Practical use of the Proposed Reinforcement Technique

For introducing the proposed application way of SVD based fuzzy rule based approximation techniques in reinforcement learning, a simple application example, where the state-transition function characterised by the following equation, was chosen:

$$s^{k+1} = 2 \cdot \left(s^k + a^k \right), \tag{11}$$

where $s \in S = [-1,1]$ is the one dimensional state and $a \in A = [-0.2, 0.2]$ is the action. The reward is calculated in the following manner: $r = 1$ *iff* $s \in [-0.1, 0.1]$ *else* $r = 0$

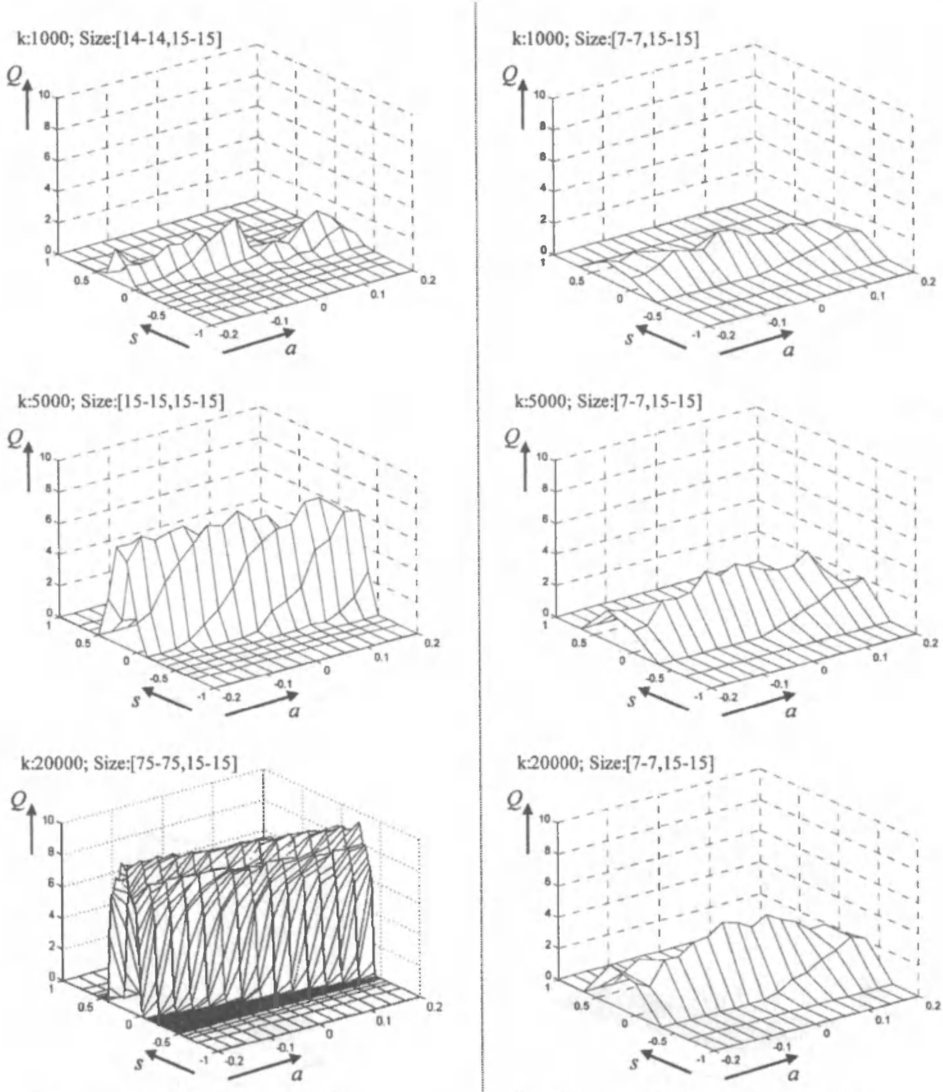
The first experiment is related to the efficiency of the proposed dynamic partition allocation method (see results on Fig.7). Fig.7.b is introducing the two basic problems of fixed partition: The lack of universal approximation property in case of rough partition and the difficulties of the adaptation.

The second experiment is related to the efficiency of the proposed SVD based complexity reduction and approximation adaptation (fast adaptation method). Fig.8. introduces three stages of a 20000 step iteration. On Fig.8.a the iteration process turns the action-value rule base to reduced form at the iteration step 1000, by applying the SVD Based Fuzzy Rule Base Complexity Reduction (introduced in Section 4.1.) From this step the iteration is continuing up to 20000 iterations using the fast adaptation method (introduced in Section

4.2.). Fig.8.b is the same experiment as Fig.8.a, except the turning the reduction is done earlier at the step 5000.

6. Conclusions

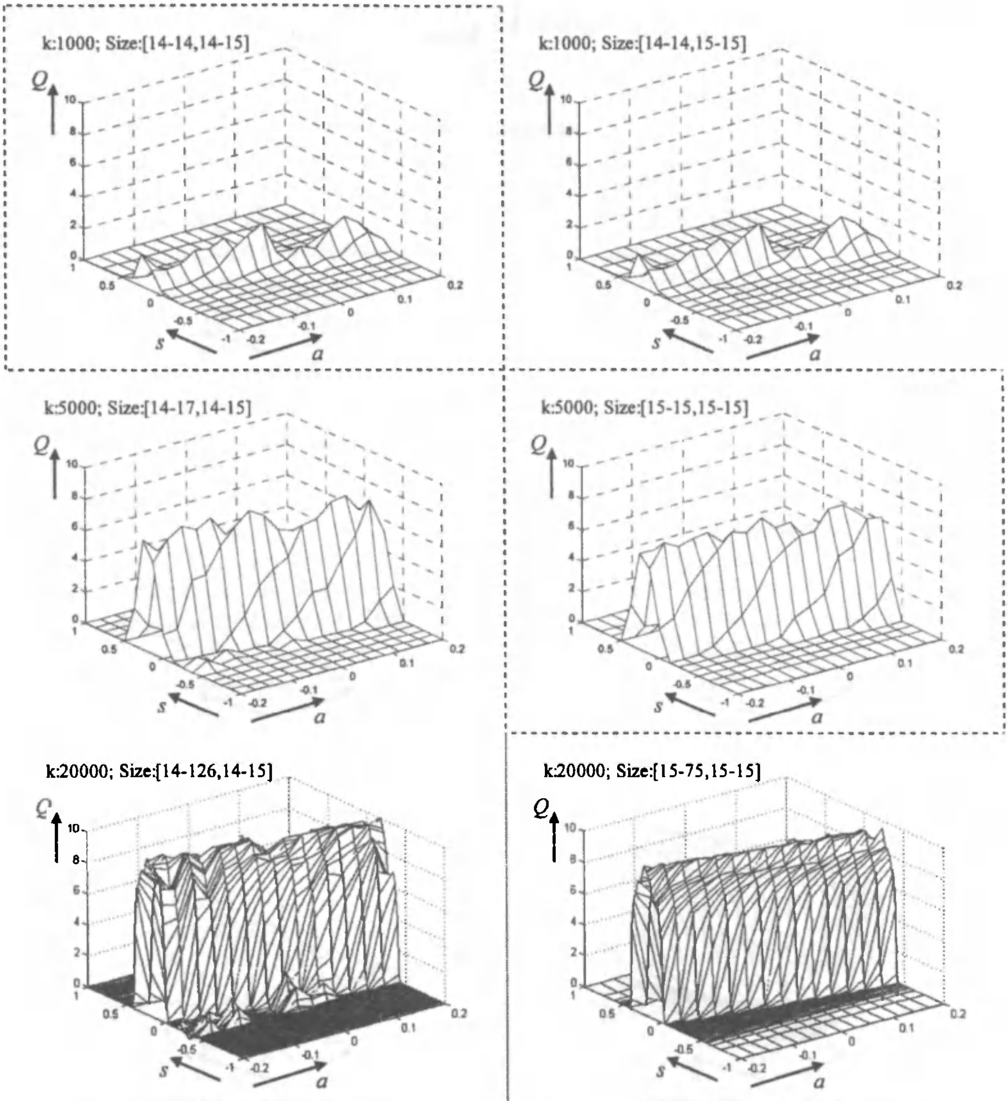
One of the possible difficulties of the reinforcement learning applications in complex situations is the huge size of the state-value- or action-value-function representation [17]. The case of continuous environment reinforcement learning could be even complicated, in case of applying dense partitions to describe the continuous universes, to achieve precise approximation of the basically unknown state-value- or action-value-function. The fine resolution of the partitions leads to high number of states, and handling high number of states usually leads to high computational costs, which could be unacceptable not only in many real time applications, but in case of any real (limited) computational resource. As a simple solution of these problems, in this paper the adoption of Higher Order SVD [5] based fuzzy rule base complexity reduction techniques and its fast adaptation method [6] is suggested. The application of the fast adaptation method [6] gives a simple way for increasing the rule density of a rule base stored in a compressed form directly. To fully exploit this feature, a dynamic partition allocation method is also suggested. Based on the application examples, the main conclusion of this paper is the reducibility of action-value function. It seems that in many cases the representation of the action-value function is considerably reducible. Moreover due to the fast adaptation method this reduction can be performed in an early stage of the adaptation and the iteration steps can be continued on an economic sized action-value function representation.



a, Dynamic partition allocation

b, Fixed, 7 equidistant set partition

Fig. 7. The first experiment, the lack of universal approximation property in case of rough predefined fixed partition (difficulties in adaptation).



a, Reduction at iteration step 1000

b, Reduction at iteration step 5000

Fig. 8. The effect of SVD based complexity reduction and approximation adaptation, where k is the iteration number and Size is the size of the reduced (*B'* as it is stored) and the extended (*B* as its used) action-value rule base (e.g. Size:[14-126,14-15] means, that the original 126x15 sized action value rule base is stored and adapted in a 14x14 reduced format).

ACKNOWLEDGEMENTS

This research was partly supported by the Hungarian National Scientific Research Fund grant no. F 029904. Szilveszter Kovács is supported by the György Békésy Postdoctoral Scholarship. Péter Baranyi is supported by the Zoltán Magyary Postdoctoral Scholarship.

REFERENCES

1. APPL, M.: *Model-based Reinforcement Learning in Continuous Environments.* Ph.D. thesis, Technical University of München, München, Germany, dissertation.de, Verlag im Internet (2000)
2. BARANYI, P., YAM, Y.: *Singular Value-Based Approximation with Non-Singleton Fuzzy Rule Base.* 7th Int. Fuzzy Systems Association World Congress (IFSA'97) Prague (1997) pp 127-132.
3. BARANYI, P., YAM, Y. VÁRLAKI, P., MICHELBERGER, P.: *Singular Value Decomposition of Linguistically Defined Relations.* Int. Jour. Fuzzy Systems, Vol. 2, No. 2, June (2000) pp 108-116.
4. BARANYI, P., VÁRKONYI-KÓCZY, A.R.: *Adaptation of SVD Based Fuzzy Reduction via Minimal Expansion.* IEEE Trans. on Instrumentation and Measurement, Vol. 51, No. 2 (2002) pp 222-226. (ISSN 0018-9456)
5. BARANYI, P., VÁRKONYI-KÓCZY, A.R., YAM, Y., PATTON, R.J., MICHELBERGER, P., SUGIYAMA, M.: *SVD Based Reduction to TS Fuzzy Models.* IEEE Transaction on Industrial Electronics, Vol. 49, No. 2, 2002, pp 433-443.
6. BARANYI, P., VÁRKONYI-KÓCZY, A.R., YAM, Y., VÁRLAKI, P., MICHELBERGER, P.: *An Adaption Technique to SVD Reduced Rule Bases.* IFSA 2001, Vancouver (2001) pp 2488-2493.
7. BELLMAN, R. E.: *Dynamic Programming.* Princeton University Press, Princeton, NJ (1957)
8. BERENJI, H.R.: *Fuzzy Q-Learning for Generalization of Reinforcement Learning.* Proc. of the 5th IEEE International Conference on Fuzzy Systems (1996) pp 2208-2214.
9. BONARINI, A.: *Delayed Reinforcement, Fuzzy Q-Learning and Fuzzy Logic Controllers.* In Herrera, F., Verdegay, J. L. (Eds.) Genetic Algorithms and Soft Computing, (Studies in Fuzziness, 8), Physica-Verlag, Berlin, D, (1996) pp 447-466.
10. CASTRO, J.L.: *Fuzzy Logic Controllers are Universal Approximators.* IEEE Transaction on SMC, Vol.25, 4 (1995)
11. GLORENNEC, P.Y., JOUFFE, L.: *Fuzzy Q-Learning.* Proc. of the 6th IEEE International Conference on Fuzzy Systems (1997) pp 659-662.
12. HORIUCHI, T., FUJINO, A., KATAI, O., SAWARAGI, T.: *Fuzzy Interpolation-Based Q-learning with Continuous States and Actions.* Proc. of the 5th IEEE International Conference on Fuzzy Systems, Vol.1 (1996) pp 594-600.
13. KÓCZY, L.T., HIROTA, K.: *Size Reduction by Interpolation in Fuzzy Rule Bases.* IEEE Trans. SMC, vol. 27 (1997) pp 14-25.

14. RUDAS, I.J.: *Towards the generalization of t-operators: a distance-based approach.* Journal of Information and Organizational Sciences. Vol.23. No.2. (1999) pp 149-166.
15. RUDAS, I.J. KAYNAK, M.O.: *Entropy-Based Operations on Fuzzy Sets.* IEEE Transactions on Fuzzy Systems, vol.6, no. 1, (1998) pp 33-40.
16. RUDAS, I.J., KAYNAK, M.O.: *Minimum and maximum fuzziness generalized operators.* Fuzzy Sets and Systems (1998) pp 83-94.
17. SUTTON, R. S., BARTO, A. G.: *Reinforcement Learning: An Introduction*, MIT Press, Cambridge (1998)
18. TIKK, D.: *On nowhere denseness of certain fuzzy controllers containing prerestricted number of rules.* Tatra Mountains Mathematical Publications vol. 16. (1999) pp 369-377.
19. YAM, Y.: *Fuzzy approximation via grid point sampling and singular value decomposition.* IEEE Trans. SMC, Vol. 27 (1997) pp 933-951.
20. YAM, Y., BARANYI, P., YANG, C. T.: *Reduction of Fuzzy Rule Base Via Singular Value Decomposition.* IEEE Transaction on Fuzzy Systems. Vol.: 7, No. 2 (1999) pp 120-131.
21. YEN, J., WANG, L.: *Simplifying Fuzzy Rule-based Models Using Orthogonal Transformation Methods.* IEEE Trans. SMC, Vol 29: Part B, No. 1 (1999) pp 13-24.
22. WANG, L.X.: *Fuzzy Systems are Universal Approximators.* Proceedings of the First IEEE Conference on Fuzzy Systems, San Diego (1992) pp 1163-1169.
23. WATKINS, C. J. C. H.: *Learning from Delayed Rewards.* Ph.D. thesis, Cambridge University, Cambridge, England (1989).