Complexity Reduction in Reinforcement Learning

Szilveszter Kovács[1], Péter Baranyi[2] and Masaharu Sugiyama[3]

Intelligent Integrated Systems Japanese Hungarian Laboratory,
Budapest University of Technology and Economics, Hungary
Gifu Prefectural Research Institute of Manufacturing Information Technology, Japan
e-mail[1]: szkszilv@gold.uni-miskolc.hu
e-mail[2]: baranyi@ttt-202.ttt.bme.hu
e-mail[3]: sugi@gifu-irtc.go.jp

Abstract

Reinforcement learning 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 [2]. 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.

1. Introduction

Reinforcement learning methods are trial-and-error stile 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 [1]. One common goal of the reinforcement learning strategies is to find an optimal policy by building the state-value- or action-value-function [2]. 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 behavior, 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 take the optimal (the greatest estimated value) action in every states, can be constructed as [2]:

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

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. RL in Continuous Environment

To implement reinforcement learning (RL) 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 [2]. One of the difficulties of building an appropriate partition structure 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.

[1] On leave from the Department of Information Technology, University ofMiskolc, Hungary.

1.2. Fuzzy Techniques in Continuos RL

There are many methods in the literature for applying fuzzy techniques in reinforcement learning (e.g. for 'Fuzzy Q-Learning" [10, 11, 12, 13, 6]). One of the main reason of their application beyond the simplicity of expressing priory knowledge in the form of fuzzy rules, is the universal approximator property [8,9] 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 [17], because of the exponential complexity problem of fuzzy rule bases [14,3,4]. 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 [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 in [17]. 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 [10], [11], [12] and [13] are sharing the same concept of fixed, predefined fuzzy antecedent partitions, for state representation. One possible solution for this problem is suggested in [6]. 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.

1.3. Fuzzy rule base complexity reduction

The main idea of application fuzzy rule base complexity reduction techniques for reinforcement learning is enhancing the universal approximator property of the fuzzy inference by extending the number of antecedent sets while the computational complexity is kept relatively low.

Some reduction techniques are classified regarding their concept in [15] and [4]. A fuzzy rule importance based technique is proposed by Song et al. in [21]. Another recent method proposed by Sudkamp et al. [23] combines rule learning with a region merging strategy.

Recently, several publications have applied orthogonal transformation methods for selecting important rules from a given rule base, for instance, in 1999 Yen and Wang investigated various techniques in [15] for possible fuzzy rule base simplification techniques such as orthogonal least-squares, eigenvalue decomposition, SVD-QR with column pivoting method, total least square method and direct SVD method. [22] also proposes an SVD based technique with examples.

SVD based fuzzy approximation technique was initialized in 1997 [16], 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 [3,25,26]. 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. [3,16] characterizes fuzzy functions by the conditions of sum-normalization (SN), nonnegativenes (NN) and normality (NO), and extends SVD reduction with further tools to preserve SN and NN conditions of the new membership functions. It may have significant role if the purpose is not only saving computational cost, but maintaining the fuzzy concept and having a theoretical study of the reduced rule's features.

An extension of [15] to multi-dimensional cases may also be conducted in a similar fashion as the higher order SVD reduction technique proposed in [14,3,16]. Further developments of SVD based fuzzy reduction [3,16] are proposed in [14,19,20,24].

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 Q-Learning [5], was chosen.

The goal of the Q-learning is to find the fixed-point solution Q of the Bellman Equation [1] through iteration. In discrete environment *Q-Learning* [5], the action-value-function is approximated by the following iteration:

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

where $\widetilde{Q}_{i,u}^{k+1}$ is the $k+1$ iteration of the action-value 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 $\alpha_{i,u}^{k} \in [0,1]$ is the step size parameter (which can change during the iteration steps).

For applying this iteration to continuous environment by adopting fuzzy inference (Fuzzy Q-Learning), there are many solutions exist in the literature [10, 11, 12, 13, 5].

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 [10] and [6]). 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:

If s **is** S_i **And** a **is** A_u **Then** $\widetilde{Q}(s,a) = Q_{i,u}$,

$$i \in I, u \in U, \quad (3)$$

where S_i is the label of the ith membership function of the n dimensional state space, A_u is the label of the uth 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):

$$\widetilde{\pi}(i) = \arg\max_{u \in U} Q(i,u), \quad (4)$$

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

$$f = \sum_{j_1, j_2, \dots, j_N}^{J_1, J_2, \dots, J_N} \prod_{n=1}^{N} \mu_{j_n,n}(x_n) b_{j_1 j_2 \dots j_N}. \quad (5)$$

where $\mu_{j_n,n}(x_n)$ is the membership value of the j_n th antecedent fuzzy set at the n th dimension of the N dimensional antecedent universe at the state-action observation x_n and $b_{j_1 j_2 \dots j_N}$ is the value of the singleton conclusion of the $j_1 j_2 \dots j_N$ th fuzzy rule.

Applying the notation introduced in (3), equation (5) turns to the following:

$$\widetilde{Q}(s,a) = \sum_{i_1, i_2, \dots, i_N, u}^{I_1, I_2, \dots, I_N, U} \prod_{n=1}^{N} \mu_{i_n,n}(s_n) \cdot \mu_u(a) \cdot q_{i_1 i_2 \dots i_N u} \quad (6)$$

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 th state antecedent fuzzy set at the n 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 th action antecedent fuzzy set of the one dimensional action antecedent universe at the action selection a and $q_{i_1 i_2 \dots i_N u}$ is the value of the singleton conclusion of the $i_1 i_2 \dots i_N u$ th fuzzy rule.

Applying the approximation formula of the Q-learning (2) for adjusting the singleton conclusions in (5), leads to the following function:

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

$$q_{i_1 i_2 \dots i_N u}^{k+1} = q_{i_1 i_2 \dots i_N u}^{k} + \prod_{n=1}^{N} \mu_{i_n,n}(s_n) \cdot \mu_u(a) \, \alpha_{i,u}^{k} \cdot \left(g_{i,u,j} + \gamma \cdot \max_{v \in U} \widetilde{Q}_{j,v}^{k+1} - \widetilde{Q}_{i,u}^{k} \right) \quad (7)$$

where $q_{i_1 i_2 \dots i_N u}^{k+1}$ is the $k+1$ iteration of the singleton conclusion of the $i_1 i_2 \dots i_N u$ 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 $\alpha_{i,u}^{k} \in [0,1]$ is the step size parameter. The $\max_{v \in U} \widetilde{Q}_{j,v}^{k+1}$ and $\widetilde{Q}_{i,u}^{k}$ action-values can be approximated by equation (6).

3. SVD reduction in RL

One of the natural problems of any complexity reduction technique is that 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 an algorithm capable of embedding new approximation points into the reduced approximation while the calculation cost is kept.

To facilitate the distinction between the types of given quantities, the notation will be reflected by their representation: scalar values are denoted by lower-case letters $\{a, b, \dots; \alpha, \beta, \dots\}$; column vectors and matrices are given by bold-face letters as $\{\mathbf{a}, \mathbf{b}, \dots\}$ and $\{\mathbf{A}, \mathbf{B}, \dots\}$ respectively, matrix $\mathbf{0}$ contains zero values only; tensors correspond to capital letters as $\{A, B, ...\}$. The transpose of matrix $\mathbf{A}$ is denoted as $\mathbf{A}^{\mathrm{T}}$. Subscript is consistently used for a lower order of a given structure. E.g. and element of matrix $\mathbf{A}$ is defined by row-column number i, j symbolized as $(\mathbf{A})_{i,j} = a_{i,j}$. Systematically, the i-th column vector of $\mathbf{A}$ is denoted as $\mathbf{a}_i$, i.e. $\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \cdots]$. To enhance the overall readability characters $i, j, ...$ in the meaning of indices (counters), $I, J, ...$ are reserved to denote the index upper bounds, unless stated otherwise. $\Re^{I_1 \times I_2 \times ... \times I_N}$ is the vector space of real valued $(I_1 \times I_2 \times ... \times I_N)$-tensors. Letter N serves to denote the number of variables. Letter k has special role and it is: $k = 1...N$, $k \neq n$.

See detailed discussion and notation of matrix SVD and Higher Order SVD (HOSVD) in [14].

3.1. SVD Based Reduction

Since the state action value function is approximated by PSG method this section is intended to provide a brief survey of the fundamentals in SVD based PSG fuzzy rule base reduction techniques, which are proposed in [16,17,4,14]. The calculation complexity of (5) explodes with values $J_1, J_2, ... J_N$, in this regards, for comprehensive analysis and exact theorems, see [17]. Decreasing the upper bound of the indices in the sum operator of (5), namely the number of antecedent sets, leads to the initial idea of calculation reduction. Formula (5) can be equivalently written in tensor product form as: $f(x_1, x_2, ..., x_N) = B \overset{N}{\underset{n=1}{\otimes}} \mathbf{m}_n$, where tensor $B \in \Re^{J_1 \times J_2 ... \times J_N}$ and vector $\mathbf{m}_n$ respectively contain elements $b_{j_1 j_2 ... j_N}$ and $\mu_{j_n, n}(x_n)$. This reduction can be conceptually obtained by reducing the size of tensor B via Higher Order SVD (HOSVD). According to the special terms in this topic the following naming has emerged [14,3,4]:

DEFINITION 1. **(Exact / non-exact reduction)** Assume an N-th order tensor $A \in \Re^{I_1 \times I_2 \times ... \times I_N}$. **Exact** reduced form $A = A^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n$, where "r" denotes "reduced", is defined by tensor $A^r \in \Re^{I_1^r \times I_2^r \times ... \times I_N^r}$ and basis matrices $\mathbf{U}_n \in \Re^{I_n \times I_n^r}$, $\forall n : I_n^r \leq I_n$ 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.

The above properties directly lead to the following fundamental concept of.

METHOD 1. **(exact SVD based fuzzy rule base reduction)** The SVD based fuzzy rule base reduction introduced in transforms equ. (5) to the form of:

$$f(x_1, x_2, ..., x_N) = \sum_{j_1, j_2, ..., j_N}^{J_1^r, J_2^r, ..., J_3^r} \prod_{n=1}^{N} \mu_{j_n, n}^r(x_n) b_{j_1 j_2 ... j_N}^r, \quad (8)$$

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

The reduced form is obtained via Method 1 capable of decomposing B into $B = B^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{U}_n$. Having $B^r \in \Re^{J_1^r \times J_2^r ... \times J_N^r}$ and its singular vectors the reduced form is determined as: $f(x_1, x_2, ..., x_N) = B^r \overset{N}{\underset{n=1}{\otimes}} \mathbf{m}_n^r$, where $\mathbf{m}_n^r = \mathbf{U}_n^{\mathrm{T}} \mathbf{m}_n$. Equation (5) is an equivalent of (8) that is the starting point for theoretical developments of this topic.

REMARK-1: Note that, the obtained functions may not be interpretable as antecedent fuzzy sets. In order to obtain functions which can be represented as antecedent fuzzy stets, further to have *Ruspini* partition, sum-normalization (SN), nonnegativenes (NN) and normality (NO) transformation techniques are developed to HOSVD algorithm in [14,3,4].

REMARK-2: The error bound advantage of the reduction technique is conceptually obtained by the truncation of non-zero singular values. The error bound of $\hat{f}(x_1, x_2, ..., x_N)$ can be estimated during the execution of the SVD reduction algorithm. As a matter of fact the final error of $\hat{f}(x_1, x_2, ..., x_N)$ depends on the type of the antecedent functions applied. Typical practical cases are analyzed in [4].

3.2. SVD Adaption

The essence of this section is to improve the rule density of B^r with A, which is the result of reinforcement at a certain state, when the increase of the density of the approximation points is required. Only those sub-tensors of A will be embedded into B^r, which are linearly dependent from B^r. Since the number of singular vectors is fixed, no SVD is needed during embedding, which offers a chance to develop a fast algorithm to adapt HOSVD. A subsequent aim is here to develop an error controllable property as well, while the low calculation cost and simplicity are maintained. A localized error threshold is in focus here for projecting the new points into the compressed form. Here an elementary step of the idea is discussed when the rule base is being increased n an arbitrary dimension n.

METHOD 2 (n **mode fast adaption)** Assume a reduced rule base defined by tensor $B^r \in \Re^{J_1^r \times J_2^r \times .. \times J_N^r}$ and its corresponding matrices $\mathbf{Z}_n \in \Re^{J_n \times J_n^r}$ resulted from B by Method 1. Furthermore, let $A \in \Re^{J_1 \times J_2 \times ... J_{n-1} \times I \times J_{n+1} \times .. \times J_N}$ be given, that should be embedded in the original rule base and, hence, has the same size as B except in the n-th dimension where I may differ from J_n. The localized error threshold of the adaption is defined by ∇.

The goal is to determine the reduced form E^r of extended rule base E defined by tensor $E' = \begin{bmatrix} B & A' \end{bmatrix}_n$, where E' contains the selected n mode sub-tensors of E according to the given error threshold ∇ as

$$\hat{E}' = \left(B^r \bigotimes_{k=1}^{N} \mathbf{Z}_k \right) \times_n \mathbf{U}. \qquad (9)$$

$A' \in \Re^{J_1 \times J_2 \times .J_{n-1} \times I' \times J_{n+1} \times .\times J_N}$ contains the selected n mode sub-tensors of A and lets the corresponding sub-tensors $T'_{\text{min/max}}$ selected from the corresponding $T_{\text{min/max}}$. 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 fulfill (9) subject to $\hat{E}' - E' \in_t \nabla'$.

The detailed description of the fast adaption algorithm is given in [27]

3.3. Example Application

For demonstrating the efficiency of the proposed reinforcement learning method in overcoming complex tasks, as one of the most popular application example, robot navigation was chosen. Our simulated robot has idealistic dynamic and omnidirectional kinematics. The structure of its simulated sensor system is built up based on a real robot equipped with three PAL omnidirectional lenses [28]. In the original sensor configuration, the three omnidirectional image of the three lenses, is served for shaping an omnidirectional 3D image (see Fig.1 for the original sensor configuration). In our simulated example as an observation of the robot, a set of direction-distance information is applied only (these direction distance vectors can be based on the original omnidirectional 3D image) see on Fig.2.

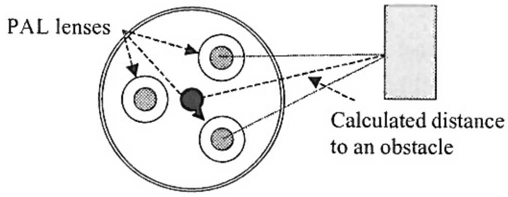

Fig.1. Sensor configuration of the simulated robot

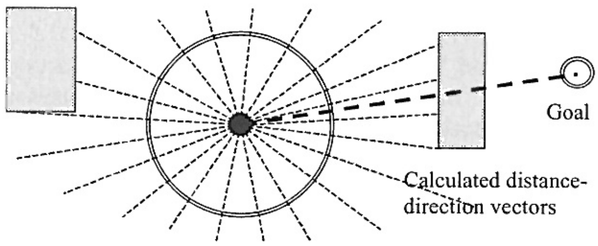

Fig.2. Virtual sensor configuration (the distance-direction vectors are fetched from the omnidirectional 3D image)

The tasks we are planning to implement for the robot, is to navigate among obstacles, and reach the goal point. The goal point is defined in the same simulated direction-distance manner as it was introduced in the case of the virtual sensors. During the training we plan to reinforce the collision to an obstacle by a negative, while reaching the goal point by a positive signal. The state value utilised for the reinforcement learning can be formed from the normalised vector of the distances sensed by the virtual distance sensors.

4. 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 [2]. 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 solution of these problems, in this paper in a short introductory way, we suggest the application of HOSVD based complexity reduction techniques to tackle both high storage and computational costs. The way of fast adaption of the compressed form is also drafted.

Acknowledgment

This research was partly supported by the Hungarian National Scientific Research Fund grant no: F 029904.

References

[1] R. E. Bellman: "Dynamic Programming", Princeton University Press, Princeton, NJ, 1957.

[2] R. S. Sutton, A. G. Barto: "Reinforcement Learning: An Introduction", MIT Press, Cambridge, 1998.

[3] Y. Yam, P. Baranyi, C. T. Yang: "Reduction of Fuzzy Rule Base Via Singular Value Decomposition", IEEE Transaction on Fuzzy Systems. Vol.: 7, No. 2, 1999, pp. 120-131.

[4] P. Baranyi, Y. Yam: "Fuzzy rule base reduction", Chapter 7 of Fuzzy IF-THEN Rules in Computational Intelligence: Theory and Applications Eds., D. Ruan and E.E. Kerre, Kluwer, 2000, pp 135-160.

[5] C. J. C. H. Watkins: "Learning from Delayed Rewards", Ph.D. thesis, Cambridge University, Cambridge, England, 1989.

[6] M. Appl: "Model-based Reinforcement Learning in Continuous Environments", Ph.D. thesis, Technical University of München, München, Germany, dissertation.de, Verlag im Internet, 2000.

[7] R. Bellman: "Dynamic Programming", Princeton University Press, 1957.

[8] L.X. Wang: "Fuzzy Systems are Universal Approximators", Proceedings of the First IEEE Conference on Fuzzy Systems, San Diego, pp.1163-1169, 1992.

[9] J.L. Castro: "Fuzzy Logic Controllers are Universal Approximators", IEEE Transaction on SMC, Vol.25, 4, 1995.

[10] T. Horiuchi, A. Fujino, O.Katai, T. Sawaragi: "Fuzzy Interpolation-Based Q-learning with Continuous States and Actions", Proc. of the 5th IEEE International Conference on Fuzzy Systems, Vol.1, pp.594-600, 1996.

[11] P.Y. Glorennec, L. Jouffe: "Fuzzy Q-Learning", Proc. of the 6th IEEE International Conference on Fuzzy Systems, pp.659-662, 1997.

[12] H.R. Berenji: "Fuzzy Q-Learning for Generalization of Reinforcement Learning", Proc. of the 5th IEEE International Conference on Fuzzy Systems, pp.2208-2214, 1996.

[13] A. Bonarini: "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, 447-466., 1996.

[14] P.Baranyi, A.Várkonyi-Kóczy, Y.Yam, R.J.Patton, P.Michelberger and M.Sugiyama "SVD Based Reduction of TS Models" IEEE Trans. Industrial Electronics (accepted with minor revision)

[15] J.Yen, and L.Wang, "Simplifying Fuzzy Rule-based Models Using Orthogonal Transformation Methods", IEEE Trans. SMC, 1999, Vol 29: Part B, No. 1, pp. 13-24.

[16] Y.Yam "Fuzzy approximation via grid point sampling and singular value decomposition" IEEE Trans. SMC, 1997, Vol. 27, pp. 933-951.

[17] L.T.Kóczy and K.Hirota, "Size Reduction by Interpolation in Fuzzy Rule Bases," IEEE Trans. SMC, 1997, vol. 27, pp. 14-25.

[18] D.Tikk, "On nowhere denseness of certain fuzzy controllers containing prerestricted number of rules", Tatra Mountains Mathematical Publications vol. 16. pp. 369-377, 1999

[19] K.Lei, P.Baranyi, and Y.Yam, "Complexity Minimalisation of Non-singleton Based Fuzzy-Neural Network" International Journal of Advanced Computational Intelligence, vol.4, no.4, 2000. pp.1-8

[20] P.Baranyi, Y.Yam, P.Várlaki and P.Michelberger "Singular Value Decomposition of Linguistically Defined Relations", Int. Jour. Fuzzy Systems, Vol. 2, No. 2, June 2000. pp. 108-116.

[21] F.Song and S.M.Smith "A Simple Based Fuzzy Logic Controller Rule Base Reduction Mathod" IEEE Int. Conf. System Man and Cybernetics (IEEE SMC' 2000), 2000, Nashville, Tennessee, USA, pp. 3794-3798

[22] Magne Setnes and Hans Hellendoorn "Orthogonal Transforms for Ordering and Reduction of Fuzzy Rules" 9th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2000), San Antonio, Texas, 2000, pp. 700-705.

[23] T.Sudkamp, A.Knapp and J.Knapp "A Greedy Approach to Rule Reduction in Fuzzy Models" IEEE Int. Conf. System Man and Cybernetics (IEEE SMC'2000), 2000, Nashville, Tennessee, USA, pp. 3716-3721

[24] P.Baranyi, Y.Yam, C.T. Yang, P.Várlaki and P.Michelberger "Generalised SVD Fuzzy Rule Base Complexity Reduction" International Journal of Advanced Computational Intelligence (accepted, to be printed in 2001)

[25] P.Baranyi and Yeung Yam "Singular Value-Based Approximation with Non-Singleton Fuzzy Rule Base" 7th Int. Fuzzy Systems Association World Congress (IFSA'97) 1997 Prague pp. 127-132.

[26] P.Baranyi and Yeung Yam "Singular Value-Based Approximation with Takagi-Sugeno Type Fuzzy Rule Base" 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97) 1997 Barcelona, Spain, pp 265-270.

[27] P. Baranyi, A.R. Várkonyi-Kóczy, Y. Yam, P. Várlaki, P. Michelberger, "An Adaption Technique to SVD Reduced Rule Bases", IFSA 2001, Vancouver (accepted for presentation).

[28] P. Greguss, A.H. Vaughan: "Development and Optimization of Machine Vision Systems Using Panoramic Annular Lens PAL", Proc. of ICAR, 1999.