Algorithms and error-bounds for the usage of soft computing tools in anytime systems

PhD Thesis summary

Orsolya Takács
Supervisor: Annamária R. Várkonyi-Kóczy
2004
1. Introduction

Nowadays, the industrial applications can be characterized by two mean features: their time-critical property and the increasing complexity of the problems to be solved. The different fuzzy and fuzzy-neural tools can be well used for solving complex modeling and controlling problems, where the exact physical-mathematical description of the system is not known, or is too complicated, and thus the classical tools can not or can only with difficulties be used [1-5]. These soft computing tools can also be advantageously used in case of newer, “intelligent” application areas, like autonomous robot-navigation or document-filtering.

On the other hand, in fuzzy and fuzzy-neural tools, higher complexity – higher number of antecedent fuzzy sets or neurons – usually results more accurate approximation [6,7]. Because these techniques are usually used in case of systems, where the mathematical description is not known, it is usually not possible to determine the needed number of antecedent fuzzy sets/neurons in advance, and there is no universal algorithm for the approximation of the needed complexity or for the optimal choosing of antecedent fuzzy sets. To achieve a good approximation, one is tempted to overestimate the size of the fuzzy system/neural network, and the result is a large and redundant system, with too high complexity and computational need, which can make difficult its usage in time-critical systems.

One solution for the complexity-problem can be the using of different complexity-reduction algorithms, which can be categorized into two groups. The algorithms, belonging to the first group, are alternative inference methods, with reduced computational need, while the techniques of the other group can be used to reduce the complexity of existing rule-bases. Algorithms, belonging to this second group, can be used, if the rule-base is not modified during the operation of the system.

In case of time-critical systems, the amount of available time and resources are often not only limited, but also can change during the operation of the system. These temporary shortages of resources can cause critical breakdowns in the performance, or even can make impossible the operation of the system.
To avoid critical situations, the so-called anytime techniques can be used [8-10], which make the system to react flexibly to the changes of working environment and to tolerate temporary shortages of time, resources or input data. The missing input data can be substituted with the help of some prediction technique, while in case of shortage of time or resources the complexity of the computing algorithms should be temporarily reduced and the available resources should be optimally redistributed. Naturally, the temporal complexity-reduction results the temporal reduction of the accuracy of the computations, thus approximation of this temporal error is also an important task.

While there are known anytime algorithms for several problems and a framework for building anytime-systems also exists [8], the insertion of fuzzy and fuzzy-neural elements into anytime-systems is still unsolved. On the other hand, because of the increasing complexity of industrial applications and the growing popularity of fuzzy and fuzzy-neural techniques, it would be advantageous to combine these soft computing tools with anytime techniques.

The main problem is that the complexity of these systems increases exponentially with the number of inputs, and there is no universal method for the approximation of the needed complexity in advance. For solving these problems, some flexible complexity-reduction method is needed for the reduction of fuzzy and fuzzy-neural systems.

The author has chosen for this purpose the so-called SVD-based (Singular Value Decomposition) complexity-reduction. The SVD-based complexity reduction was first proposed by Yam as a method for creating fuzzy inference systems for the approximation of functions, based on grid-point sampling [11]. Later the method was improved by Baranyi et al. and proposed as a method for the rule-base reduction for certain types of fuzzy inference systems - Product-Sum-Gravity with Singleton consequences (PSGS) [12], Product-Sum-Gravity with Nonsingleton consequences (PSGN) [13], Takagi-Sugeno [14] – and for generalized fuzzy-neural systems [15-16].

The advantage of the SVD-based reduction algorithms is, that they can be used for a wide range of fuzzy systems without change of the inference algorithm. They don’t need any special properties of the rule-base/fuzzy-neural system, nor human intervention or heuristics.

In case of exact complexity-reduction, they offer a way for the automatic elimination of the
redundancy of the rule-base or neural network, and in case of further, non-exact reduction they make possible flexible complexity-reduction: if the tolerable error can be higher, then the reduced system will be smaller.

The aim of the Thesis is twofold. First, it attempts to extend the usability of the SVD-based complexity-reduction methods. Second, it examines the possibilities of the usage of fuzzy and fuzzy-neural tools in anytime systems with the help of the SVD-based complexity-reduction algorithms.

In the first area, I’ve given new mathematical proof for the error-bound of the non-exact reduction of PSGS fuzzy systems, which doesn’t use the linearity of the antecedent fuzzy sets, thus it makes possible the usage of SVD-based complexity-reduction in case of non-linear antecedent fuzzy sets, as well. Because the reduction and the error-estimation of reduction of PSGN and Takagi-Sugeno fuzzy systems is similar, the new proof for the error-bound is valid for nonlinear-antecedent fuzzy sets in case of these systems, as well.

The new SVD-based reduction algorithm for “near PSGS” fuzzy systems, where the antecedent fuzzy sets are not in Ruspini-partition, also belongs to the first area. This can be especially useful in case of rule-bases, based on expert knowledge, because in these rule-bases the Ruspini-partition often can be ensured only with difficulties. I’ve also given the error-bound of the non-exact reduction, together with the mathematical proof.

To make easier the application of complexity-reduction techniques, I’ve proposed a new, more accurate error-estimation method for the matrix-reduction. The matrix-reduction is the base of the SVD-based complexity-reduction algorithms, thus the more accurate error estimation affects the error-bounds of these algorithms, as well. It not only can be used to make higher reduction with a given tolerable error, then the original error-estimation method, but it also makes possible the estimation of the error for sub-domains, which in some applications – e.g. at the reduction of near PSGS fuzzy systems - can give important extra information.

The SVD-based complexity-reduction algorithms make also possible the reduction of generalized fuzzy-neural networks [17,18]. In generalized fuzzy-neural networks, the non-linear transfer function is not in the nodes, but in the links, and can also be trained (Fig. 1.).
To reduce the computational complexity, the unknown weight-functions are approximated by simple – PSGS or PSGN – fuzzy systems, and the parameters of these fuzzy systems are trained. This structure was already effectively used in applications for modeling complex functions and correspondences [19,20].

For practical application of the non-exact complexity reduction of generalized fuzzy-neural networks, the estimation of the error of the reduction is also needed. Thus I’ve given error-estimations with mathematical proof for the non-exact complexity reduction of singleton-based and non-singleton-based fuzzy-neural networks. Thus the SVD-based complexity reduction algorithms can also be used for the non-exact reduction of generalized fuzzy-neural networks, and the error can always be estimated.

By pursuing the second aim, there are basically two possible ways to insert soft computing tools to anytime systems: to transform fuzzy of fuzzy-neural systems into a form, which can be evaluated iterative-type, or to insert these systems into an anytime framework.

---

Figure 1.: Generalized fuzzy-neural network
This second solution gives a possibility for the usage of wide range of fuzzy and fuzzy-neural tools in anytime systems. I’ve given an algorithm, to generate contract-type anytime algorithms from fuzzy and fuzzy-neural systems with the help of the SVD-based complexity-reduction algorithms, and I’ve also given a method to generate the performance profile of these. With this method, the above-mentioned soft computing tools can be inserted into the anytime framework of Zilberstein [8]. On the other hand, in case of basically numeric algorithms, it is more suitable to use a description based on accuracy/error-bound, instead of the more general quality, thus I’ve also given an alternative description method.

I’ve also proposed a transformation method for transforming PSGS-fuzzy systems into an iteratively computable form, which makes possible the usage of these systems in anytime applications without an anytime framework, but the algorithm, resulted by this transformation can also be inserted in the framework of Zilberstein, as an interruptible anytime algorithm. I’ve also given the error-bound data of the transformation, needed for the computation of the performance-profile.

2. Methodology

The basic method of the researches was the exploration of the existing literature and – by using mathematical proof-methods – the improvement and extension of current results. I’ve made a survey of the complexity-reduction methods of fuzzy-systems, especially the SVD-based complexity-reduction method. I’ve also studied the theory of anytime systems to find the possibilities to insert fuzzy and fuzzy-neural tools into anytime applications.

Section 2. of the Thesis summarizes the basic concepts and algorithms of SVD-based complexity reduction.

Section 3. presents new results concerning the SVD-based complexity-reduction of fuzzy systems. I give a new mathematical proof for the error-bound of the SVD-based complexity reduction of PSGS fuzzy systems, which doesn’t use the linearity of antecedent fuzzy sets. I give an algorithm for the SVD-based complexity-reduction of “near PSGS” fuzzy systems, with a mathematical proven error-bound. I also propose a new, more accurate estimation method for the error-bound of the SVD-based matrix-reduction algorithm.
In the beginning of Section 4., the description of the generalized fuzzy-neural networks and their SVD-based complexity-reduction can be found. After it, I give error-bounds with mathematical proofs to the non-exact reduction of different kind of generalized fuzzy-neural networks.

In Section 5., I give a short survey of the basics of the theory of anytime systems, and propose two different methods for the application of fuzzy systems and fuzzy-neural networks in anytime environment. The first method creates contract-type anytime algorithms from fuzzy and fuzzy-neural systems with the help of the SVD-based reduction algorithms. Then these contract-type anytime algorithms can be inserted into an anytime framework – I also give the needed data (performance profile) for it. The other method is a transformation algorithm, which transform PSGS fuzzy systems into a new form, which can be evaluated iterative-type, thus these can be used as interruptible anytime algorithms.

Section 6. contains some numeric examples to illustrate the proposed algorithms, while Section 7. shows the possible application areas of the results.

Section 8. gives a summary of the Thesis and enumerates the new results. I also mention some questions, which are in close relation with the subject of the Thesis and needs further research.

The appendix contains the main notations used in the Thesis.

3. New results

1.1. thesis-group

New error-bound and algorithm for the SVD-based complexity-reduction of given types of PSGS and near PSGS fuzzy systems.

1.1. I showed, that the error of the SVD-based complexity reduction of PSGS fuzzy systems is not greater, then the sum of the discarded singular values, even, if the antecedent fuzzy sets are non-linear (Section 3.1) [J1,C1].

1.2. I’ve given an SVD-based complexity-reduction algorithm for near PSGS fuzzy systems, where the antecedent fuzzy sets are not in Ruspini-partition. I showed, that the reduced system will be a PSGN fuzzy system (Section 3.2) [C2].

1.3. I’ve shown, that the error-bound of the SVD-based matrix-reduction algorithm can be written as the dyadic product of the discarded columns, and thus I’ve given a
more exact error-bound for the non-exact SVD-based complexity reduction algorithms (Section 3.3.) [C3].

2. thesis-group


2.1. I’ve showed, that the error-bound of the non-exact SVD-based complexity reduction of generalized fuzzy-neural networks with singleton consequents can be estimated from the sum of discarded singular values (Section 4.1.) [C4-C6,E1].

2.2. I’ve showed, that the error-bound of the non-exact SVD-based complexity reduction of generalized fuzzy-neural networks with non-singleton consequents can be estimated from the sum of discarded singular values, when the fuzzy-properties are maintained during the reduction (Section 4.2.) [J2,C7].

2.3. I’ve showed, that the error-bound of the non-exact SVD-based complexity reduction of generalized fuzzy-neural networks with non-singleton consequents can be estimated from the sum of discarded singular values, when the fuzzy-properties are lost during the reduction (Section 4.3.) [J2,C7].

3. thesis-group

Usability of soft computing methods in anytime systems.

3.1. I’ve given an algorithm for the automatic creation of contract-type anytime algorithms from existing fuzzy and fuzzy-neural systems. I’ve given a method for the determination of the performance profile of the created contract-type anytime algorithms, and I’ve proposed an alternative, accuracy-based description (Section 5.1) [J4,C6,C8-C10].

3.2. I’ve given an SVD-based algorithm for transformation of PSGS systems into a new form, which can be evaluated iterative-type (Section 5.2) [B1,J3,C11-C13,E2].

4. Application of new results

The results concerning SVD-based complexity reduction can be well used in applications, where because of the complexity of the problem – not known or too complex mathematical description, expert knowledge, which can be formalized with classical tools only with
difficulties, information available only in form of training patterns, etc. – fuzzy or fuzzy-neural systems should be used, but the available time and resources are limited.

If the fuzzy systems is based basically on expert knowledge, the linearity of the antecedent fuzzy sets or the Ruspini-partition can not be always ensured, but with the help of the algorithms, proposed in the Thesis, these fuzzy systems also can be reduced. An other possible application can be to create a huge fuzzy-system by representing the information, originating from different experts, with different fuzzy rules, and generate a smaller and more transparent rule-base with the SVD-based complexity-reduction.

With the proposed improved error-bound estimation method it is possible to have more accurate error estimations in case of already existing SVD-based applications, like e.g. the control of a prototypical aeroelastic wing section [21].

Generalized fuzzy-neural networks were already effectively used in applications like document filtering [19] and autonomous robot navigation [20]. In cases, where the available time and resources are limited, non-exact complexity-reduction can be needed, for which the proposed error-bound estimations can be used.

With the methods of Section 5., it becomes possible to use fuzzy and fuzzy-neural tools in anytime applications.

One potential application area is the area of system monitoring and diagnostic systems, which are able to supervise complex industrial processes and determine the appropriate actions in case of some deviation from the optimal operation mode [22,23]. These systems usually use the model of the optimal operation mode to detect deviations. While industrial processes can be complex, and their exact physical-mathematical description is not always known, or it depends on too many factors, measured data from the system is usually available. Thus it can be a good solution to use some trainable algorithm – e.g. a fuzzy or fuzzy-neural system.

Soft computing tools – especially fuzzy systems – can also be well used for the automatic fault-diagnosis, because knowledge, originating from experts, can also be inserted into these systems.

On the other hand, the model – or, in case of a complex system, models – must be evaluated on-line, and the fault diagnosis algorithm must also give a result with a given response time.
A further problem can be, that the available time and resources can change during the operation of the system – a computing unit can be faulty, fault diagnosis can take away resources from the model-estimation, needed respond-time of fault diagnosis can depend on the seriousness of the fault, etc.

The usage of a modular anytime system can solve these problems, and with the proposed methods, fuzzy or fuzzy-neural tools – solving one sub-task, as e.g. system modeling or fault diagnosis - can also be inserted into this system. With using anytime techniques, it is possible in case of an immediate fault diagnosis task to rearrange the resources. In this period the model – or its some, not so important parts – can be evaluated with a reduced accuracy, but this reduced accuracy can still be enough to detect serious deviations, which needs instant intervention.

An other area is, where soft computing tools and anytime techniques should be used together is the research area of intelligent agents. These agents usually fulfill tasks, which need some kind of “intelligence” – information collection, navigation and movement, planning, etc. – thus it is common to use soft computing tools in this area. On the other hand, these agents work usually in a changing environment, where they have to react to the changes in time, which results, that the perception, information processing, planning, etc. should be made with a given response time. This response time can also depend on the actual circumstances and the speed of changes of the environment. Because the agents works independently and are usually mobile, typically the amount of available resources is also limited.

With an anytime framework, in which the used fuzzy and fuzzy-neural tools can be inserted, the agent will be able to fulfill its tasks in a changing environment, as well.

5. Publications on the Topic of the Thesis

Book chapter:

Papers in Periodicals (published abroad):


Conference papers:


Paper in Periodicals (published in Hungarian):


Electronic publications


References


