Neural and Fuzzy Modelling of Hydrological Data

Neruda, Roman
2012

Institute of Computer Science
Academy of Sciences of the Czech Republic

Neural and Fuzzy Modelling of Hydrological Data

David Coufal, Roman Neruda

Technical report No. 1172

2012

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: +420 266 051 111, fax: +420 286 585 789,
e-mail:ics@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Neural and Fuzzy Modelling of Hydrological Data[1]

David Coufal, Roman Neruda

Technical report No. 1172

2012

Abstract:

The main goal of this work is to model flood waves based on runoff and precipitation data. We utilize data from the Smeda rivera catchment provided by the CHMI in order to build several models of flood episodes. Multilayer perceptron networks and Fuzzy system models are used and their performance is compared to traditional hydrological approaches.

Keywords:
Environmental modelling, Fuzzy systems, Neural networks, Meta-learning

[1]This research has been supported by the Ministry of Education of the Czech Republic under project no. OC10047.

Contents

Chapter 1

Introduction

Discovering the patterns in data usually requires deeper understanding of both the data and the data mining methods, in order to be able to use them with satisfactory results. It should therefore be the next goal of artificial intelligence research to develop meta learning systems, which would learn from their previous experience, and which would be able to give advice on what methods to use in particular situations.

The main goal of this work is to improve real time flood warning system operated by the Czech Hydrometeorological Institute (CHMI) in a very sensitive part of Northern Bohemia, the Smeda River basin. This area has been subject to several flash floods during last decade, and thus it is important to model and predict the dynamics of the flood wave. Following the operational reality we reformulated the traditional time series prediction problem as either a runoff regression problem, or a classification of water level values into predefined decisive water level thresholds. Moreover, in contrast to our previous work, the modelled system utilizes data from three subsequent runoff gauges, namely Bily Potok, Frydlant and Visnova. The distance between them is 15 km and 12 km, respectively. The watershed area is 180 km^2. Together with flood wave time series we utilized relevant precipitation totals from Hejnice rain gauge.

While it is difficult to forecast the time of occurrence and the extent of floods, it is possible to predict fairly accurately the movement of the flood wave along a river. Several methods are available for the flood wave propagation forecasting in general. Two simple hydrometric methods based on the extrapolation of the discharge difference and discharge-travel time are in use in CHMI. On a similar base, the neural model is created whose inputs are historical runoff values in the first two gauges, and an output is a classification of predicted water level (or runoff prediction) in the third gauge. Number of previous runoff values depends on the shape of a flood wave.

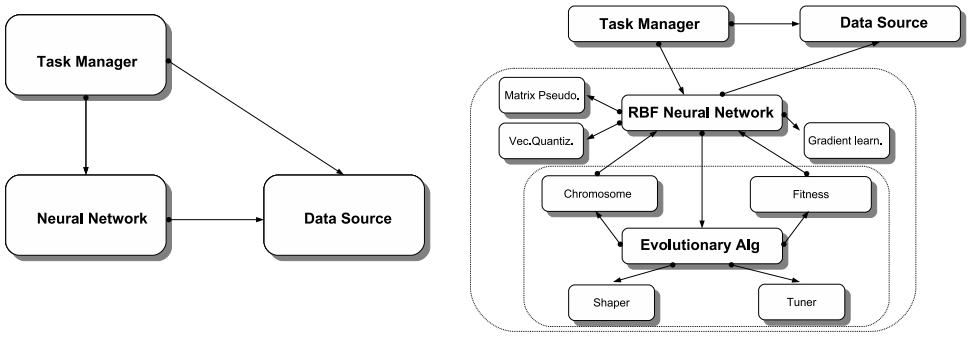

Figure 1.1: Two examples of computational MAS — the simplest one (left), and the more complicated one (right) containing a neural network trained by an evolutionary algorithm.

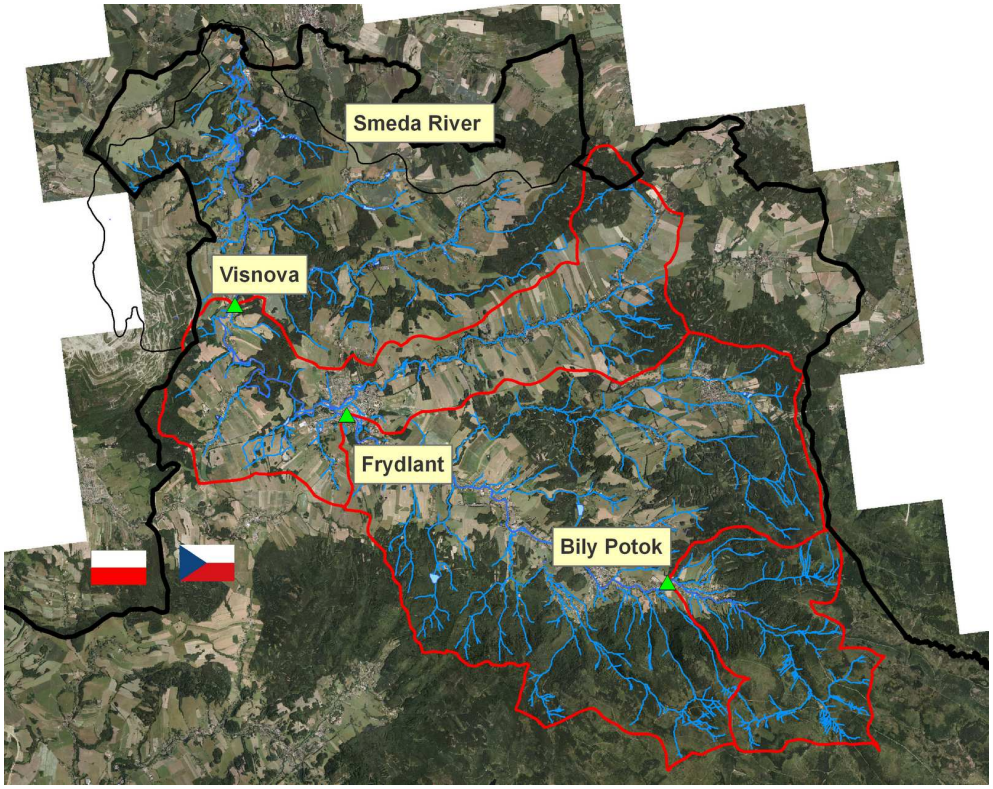

Figure 1.2: Map of the Smeda river catchment (Czech Hydrological Institute).

It has been shown that during the training phase of neural models that setting of proper configuration of the model is important for successful prediction, while the common practice is to set these parameters (type of network, number of hidden units or layers, learning rate, kernel function type) empirically. Moreover, the problem is data-dependent, thus it makes sense to use some meta-learning search heuristics to set the parameters with respect to particular data set at hand. In our research we utilized the evolutionary algorithms and local hill climbing techniques to efficiently search the parameter space in order to improve the quality of the model. While the procedure itself is computationally exacting, it provides improvement in terms of prediction quality of flood wave predictors.

An approach taking fuzzy systems into account to create a human-understandable models is also described in this report. In both neural and fuzzy models we model some relation of interest. Here it is the mapping of selected predictors on the predicted water level. However, in a neural network the function of interest is coded mainly in weights of the network an any insight on the importance and composition of the predictors can hardly be done. That is why neural networks are commonly seen as black box models.

Contrary to neural networks, in fuzzy systems, the relation of interest is created establishing the relation between fuzzy sets composing IF-THEN rules of the system. The lingustic description of fuzzy sets is generally more understable than representation of the relation in terms of weights. In the report we present a fuzzy model for water level prediction. The model is built as a radial implicative fuzzy system with S-shaped fuzzy sets allowed to be used in the model. This type of systems was studied recently, and brings an enhancement with respect to the purely radial systems. The model is presented both theoretically and practically in the form of a MATLAB code.

The neural and fuzzy modelling approaches are compared with the model which is currently in the use at CHMI (it is called Kinfill). Using real data it is shown that both approaches are capable to improve the current model. The experimental part of the report presents the details.

Chapter 2

Neuro-evolutionary systems

In this chapter we present the Pikater multi-agent system that embeds different data mining methods. The system not only provides an user-friendly environment for experimenting with different data mining methods (and is capable of figuring out their best settings), but it also learns with every task it solves, so that it is eventually capable of giving advice on choosing the appropriate data mining method.

The multi-agent-based approach brings in many advantages to the complex task of meta learning when compared to non-agent solutions (such as WekaMetal extension of Weka data mining tool [9]). The main contribution of the multi-agent-based approach lies in its distributed nature (the system can spread over the Internet and be accessed by many users that — only by using the system and running their experiments — provide the data needed for meta learning algorithms) and the easy extensibility of multi-agent system (MAS), which enables the researchers to add their own components (such as data mining or parameter space searching methods) to the system. The extensibility of our system is assured by the use of the structured ontology language and following the international standards of agents' communication.

The paper has the following structure. First we introduce the meta-learning scenario with parts devoted to parameter search, data description and recommendations, and we show a sound formal model of roles to deal with description, composition and communication of agents. Then, a system architecture is presented in the implementation section, and an example problem solution is described.

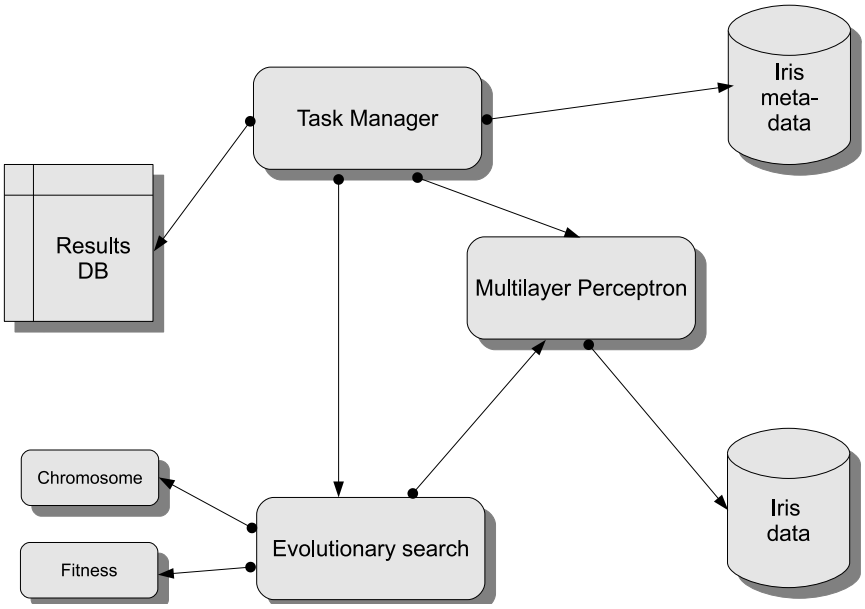

Figure 2.1: Scenario of meta learing in computational MAS

2.1 Meta learning

2.1.1 Overall scheme

One of the unique features of our system is its ability to meta learn over the results of the previous tasks and finding the best possible method even for the new datasets based on the ontological compatibility of the datasets. The system thus can in a way replace a human expert.

For a general view on the meta learning scenario in computational MAS, where the suitable computational method's parameters are chosen according to the metadata, see Figure 2.1. The metadata contain the information about the data-set (e.g. number of instances and attributes, data types, default task, or whether it contains missing values). These descriptions of computational methods correspond to physical implementation of computational agents employing the JADE agent platform and Weka data mining library [14]. The task manager has specified training and testing data and partially determined computational method with its options. It employs the search algorithm which optimizes the option set according its evaluation, i.e. error rate over the data. These evaluations are computed by means of

the computational agent (implementing multi-layer perceptron) and the data source agent. The optimal solution together with the corresponding metadata is stored in the persistent database. These results are later exploited to find the suitable configurations for unknown data according to metadata metric [11]. The MAS-based solution of meta learning allows a flexibility in choice of the parameter space search algorithms, e.g. general tabulation, random search, hill-climbing, simulated annealing or parallel methods, such as evolutionary algorithms. It is also capable of computational agents execution in parallel with various parameters during the execution of the search process.

Figure 2.1.1 describes the above mentioned procedures in a pseudo-code, thus providing a funtional poin of view on the meta learning problem. The `train` procedure realizes a basic standard task - creating a model by applying a computational intelligence method embodied in agent A on a data set D (with a set of options O). A simple `meta-train` procedure employs a search algorithm embodied in agent S in order to find an optimal options set O by iterating through train procedure. Note, that the meta-train can represent different search strategies, from simple tabelataion to evolutionary search. The `get-experience` shows a pro-active agent behavior to learn its performance on different data sets. The strategy of this background task can again be fairly complex. Finally, the `recommend` procedure sketches the task to recommend the best possible method for presented data based on the previous experience.

2.1.2 Roles description

In this section we employ the concept of role to model the organizational aspects of our multi-agent system. Roles allow flexible composition of concrete MAS of more simple blocks — group structures. We employ the AGR (Agent, Group and Role) model [7, 15]. In this model, the MAS consists of groups, instances of group structures. The agent can enter a group only by playing a role which is defined in the corresponding group structure. Two agents interact by the communication protocol defined between two roles they have assumed, where both agents have to be in the same group.

The scenario of meta learning consists of the following subproblems. The first is *administration* of the computation, i.e. decomposition of user's experiments to tasks of concrete method with possibly unfilled slots in parameters of methods, and then to pairs of method and its options. The second subproblem is *computation*, execution of computational task on train and test data from a data source with concrete computing method. The *parameter-space search* of method's options is the third suproblem of meta learning. The

```
train(A,O,D) {
  initialize agent A with options O;
  train A on data D with results E;
  return E
}

meta-train(A,D,S) {
// either error treshold or
// iteration count is met
  i=0;
  repeat
    apply search algorithm S for option O;
    E = train(A,O,D);
    i++;
  until (E < eps_E) or (i=max_I)
  return E;
}

get-experience (A) {
// D_1 ... D_n all data sets in the
// storage (or benchmark data sets)
  for i=1,..n {
    E = meta-train(A,D_i);
    store (A,D_i,E) to persitent storage;
  }
}

recommend (D) {
  find D' such that:
    m(D,D') = min { d(D,D_i); i=1..n };
  find A' such that:
    E' = test(A',D') = min { test(A,D')
      A is any comp.agent in the system }
 return A';
}
```

Figure 2.2: Pseudocode depiction of the important tasks performed by the data-mining MAS.

previous experiments' results will be stored and possibly exploited for *recommendation* of configurations, the fourth subproblem, based on metadata the new experiments. The last subproblem, the *data management*, provides data of user in the local computing system.

These subproblems correspond to group structures of the organizational structure. Thus, we can design the following group structure, which implements the subproblems:

- The *administrative group structure* contains the roles of user interface, manager and options manager.
- In the *computational group structure*, there is a task manager determining the computational task, a data source providing the data and a computing agent (e.g. RBF network) performing the task.
- The general *search group structure* consists of optimized agent, which determines the pattern of search space and returns the evaluation of concrete solutions, and of search agent implementing the search algorithm.
- The *recommendation group structure* serves as a representation of previous experience of computation. In the recommendation group, roles of experiment (...nebo tak neco...) and recommendation agent are entering.
- The *data management group structure* contains roles of a data manager and data provider. The data provider sends data to the data manager, which stores to the local machine for data sources.

The concrete organization of MAS is composed of instances of these group structures and agents which play the specific roles. In the Figure 2.1.2, there is depicted such a concrete organization of Pikater. Some of the agents play more roles in the same time. The manager in administrative group is also experiment in the recommendation group and data provider in data management group. The options manager is an optimized agent in the search group and a task manager in computational group.

2.1.3 Data description

Along with the data themselves, the system stores *metadata* (described by the ontology proposed in [16]), i.e. extra information about the training datasets, in the database. Some of those information can be obtained directly from the data, some of them need to be set by the user along with the data when specifying the task. *Metadata* contain the following items:

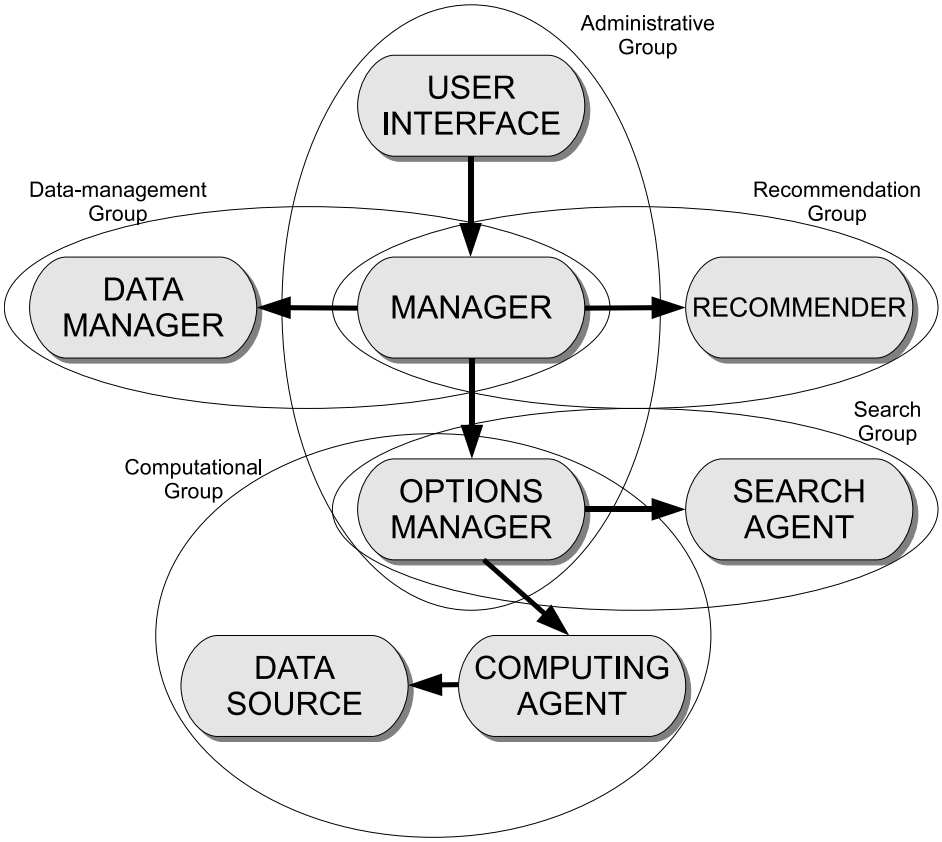

Figure 2.3: The role and group based organization of the data-mining MAS.

- *number of attributes* that specify data characteristics,
- *number of instances* — number of records in a dataset,
- *data type* — refers to all values of all attributes in the dataset. Possible values are *integer*, *real*, *categorical*, or *multivariate*,
- *default task* — type of a task that is connected with the data, currently the system can solve classification and regression types of tasks,
- *missing values*.

2.1.4 Recommendations

We determine the most similar dataset to the dataset provided by user (we use *metadata metric* proposed in [16] to determine the distance between the datasets). Afterwards we search the database for the method that showed the lowest error rate on the selected (closest) dataset.

The data stored in the database — both *results* of computations and *metadata* about the datasets together with the proposed *metric* provide fundamentals for further experimenting with different learning techniques, such as evolutionary algorithms, or application of classification methods for finding similar datasets. Since both the *metadata* and *results* are defined by ontologies, it is easy to add items (i.e. duration of the computation) and experiment with different metrics. We also intend to further improve the ontological description of tasks and data mining methods in order to provide better specification of the compatible *task-method* couples, so that we would be able to benefit from the strength of OWL-DL language and use automated reasoning for choosing the valid couples.

2.2 System Architecture

Lets have a look at the typical task that the system should be able to solve — a researcher usually wants to gain some knowledge from a dataset. They either have decided to use a particular data mining method, or they want the system to give advice on choosing the method for them. The whole solution to the data mining task thus falls apart into four separate layers (see Figure 1) — *user interface layer*, *computational layer*, *data layer*, and *administrative layer*.

At the *user interface layer*, the system handles all the communication with user, i.e. all the inputs (the definition of the task system is to solve)

and outputs of the system (the results of computations). The user defines the task in a human understandable language via one of the *user interface agents* which subsequently translate the problem into ontologies [16] and communicate it to the system.

The *computational layer* contains the specific data mining methods, such as Multilayer perceptron or RBF network (each of them embedded in an *computing agent*).

The *data layer* contains *reader agents* capable of reading data from files and *data manager service* that provides access to data stored in a database.

The most sophisticated part of the system is the *administrative layer* which carries out several functions:

- It is in charge of the whole problem solving process. It connects the *user interface layer* and the *computational layer* (*agent manager*).
- It is capable of meta learning over the methods (see 'Meta learning' section). (*agent manager*)
- It can search the parameter space of the data mining methods and choose the best configuration (*agent option manager*).
- It gathers the task results and computes some statistic information (*agent manager*).

The system is designed to easily add researchers' own *option managers*. So far the parameters values can be either chosen randomly, or the required number of the values uniformly distributed over the specified interval is selected.

To ensure the easy extensibility and re-use of the system, the general communication issues are ruled by FIPA specifications, while the inner language used in messages is specified by an OWL-DL ontology. Most of the *computing agents* in our system embed Weka [9] data mining methods. We use XML format to communicate the tasks to the system as well as for the output, i.e. the results of the tasks. The Pikater multi-agent system is being developed using JADE framework.

Chapter 3

Fuzzy systems

This chapter introduces the class of *S-shaped radial implicative fuzzy systems* along with related software implementations. The chapter consists of two parts. In the first part, the classes of the radial and S-shaped radial fuzzy systems are introduced theoretically. The presentation is mainly based on the recent paper [4]. In the second part, there are presented MATLAB implementations of the introduced systems and related routines, e.g., the script for the coherence checking algorithm.

3.1 Radial implicative fuzzy systems

Radial implicative fuzzy systems are created by the combination of the concepts of *radial fuzzy systems* and *implicative fuzzy systems*. The radial fuzzy systems incorporate *radial functions* for representation of membership functions of involved fuzzy sets. In implicative fuzzy systems, the IF and THEN parts of IF-THEN rules are combined by *residuated fuzzy implications*.

A computational model of implicative fuzzy systems is generally complicated due to the employment of (typically discontinuous) residuated implications, however, when the *radial property*, exhibited by the radial systems, is taken into account, then the computational model starts to be tractable. That is why the fusion of these two concepts is reasonable and advantageous. Moreover, the restriction on the class of radial implicative fuzzy systems is not so limiting because these systems exhibit the universal approximation property; and the representation ability of radial systems is wide enough to they can be used in real applications.

Finally, due to the tractability of the computational model the question of *coherence*, that is the question of the non-contradictoriness of stored knowledge, can be resolved effectively.

3.1.1 Implicative fuzzy systems

In what follows, we will consider implicative fuzzy systems as the systems in MISO (multiple-input, single-output) configuration of rules. This configuration means that an individual IF-THEN rule represents a fuzzy relation on an $n \times 1$-dimensional input-output space. The input-output space is treated as a subset of $\mathcal{R}^{n+1}$, $n \in \mathcal{N}$ space. Let a rule base of the fuzzy system under consideration consist of $m \in \mathcal{N}$ rules, for the j-th IF-THEN rule the relation formally writes

$$A_{j1}(x_1) \star \cdots \star A_{jn}(x_n) \to B_j(y). \tag{3.1}$$

In the formula, A_{ji}, $j = 1, \ldots, m$ represent one-dimensional fuzzy sets specified on individual subspaces $X_i \subseteq \mathcal{R}$, $i = 1, \ldots, n$ of input space $X_1 \times \cdots \times X_n \subseteq \mathcal{R}^n$. One-dimensional fuzzy sets are combined by a t-norm $\star$ to produce the IF part of the rule which corresponds to a fuzzy relation $A_j(\boldsymbol{x})$ on $\mathcal{R}^n$ space, i.e., $A_j(\boldsymbol{x}) = A_{j1}(x_1) \star \cdots \star A_{jn}(x_n)$, $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n$. The THEN part of the j-th rule is represented by a fuzzy set B_j specified on an output space $Y \subseteq \mathcal{R}$.

Implicative fuzzy systems (I-FSs in short) employ residuated fuzzy implications [13, 8] for interconnection of IF and THEN parts of rules. In formula (3.1), the implicative connection is indicated by the $\to$ symbol. Considering the more compact form for (3.1), we write the formula $A_j(\boldsymbol{x}) \to B_j(y)$ for the j-th IF-THEN rule. It mathematically represents a fuzzy relation on $\mathcal{R}^{n+1}$ space which we denote by $R_j(\boldsymbol{x}, y)$. Hence

$$R_j(\boldsymbol{x}, y) = A_j(\boldsymbol{x}) \to B_j(y). \tag{3.2}$$

As presented, we assume that rule base of the system consists of $m \in \mathcal{N}$ rules. The individual rules combine into the rule base by a fuzzy intersection. Mathematically, the fuzzy intersection operation corresponds to some t-norm, which is the operation generalizing the Boolean conjunction [5, 13]. The most common choice of the t-norm for combination of IF-THEN rules is the minimum t-norm. The final representation of the rule base has then form

$$RB(\boldsymbol{x}, y) = \bigwedge_j R_j(\boldsymbol{x}, y) = \min_j \{A_j(\boldsymbol{x}) \to B_j(y)\}. \tag{3.3}$$

The computation of the overall system is driven by the compositional rule of inference accompanied by the singleton fuzzification and the mean of maxima defuzzification method [5, 13, 17]. More details are presented below when we discuss the computational model of radial implicative fuzzy systems.

3.1.2 Radial fuzzy systems

Radial fuzzy systems employ radial functions to represent membership functions of incorporated fuzzy sets. Radial functions are well known from the theory of radial basis functions neural networks (RBF networks) [10]. A radial function is determined by its central point $\boldsymbol{a} \in \mathcal{R}^n$ and a shape modification function *act* (so-called activation function in the theory of neural networks) which is applied on a norm of distance from the central point. In our context, the shape function $act : [0, \infty) \to [0, 1]$ is non-increasing, $act(0) = 1$ and $\lim_{z\to\infty} act(z) = 0$.

Formally, a radial function has the form

$$act(|x - a|/b) \quad \text{or} \quad act(||\boldsymbol{x} - \boldsymbol{a}||_{\boldsymbol{b}}) \tag{3.4}$$

for one-dimensional or multi-dimensional case, respectively.

In the above specifications, $b > 0$ is the scaling parameter in the one-dimensional case. In the multi-dimensional case, instead of the absolute value we work with scaled norms on $\mathcal{R}^n$ space. Especially, we are aimed at the *scaled* ℓ_p *norms* defined for $p \in [1, \infty)$ and $\boldsymbol{u} \in \mathcal{R}^n$ as

$$||\boldsymbol{u}||_{\boldsymbol{b}} = \left(\sum_i \frac{|u_i|^p}{b_i^p} \right)^{1/p}. \tag{3.5}$$

The limit case of $p = \infty$ yields the scaled cubic norm specified as $||\boldsymbol{u}||_{C_{\boldsymbol{b}}} = \max_i\{|u_i|/b_i\}$. The scaling vector $\boldsymbol{b}$ reads as $\boldsymbol{b} = (b_1, \ldots, b_n)$, with $b_i > 0$.

In the context of radial fuzzy systems, the most prominent examples of radial fuzzy sets are (for the j-th rule, say) the one-dimensional Gaussian

$$A_j(\boldsymbol{x}) = \exp\left[-\frac{(x - a_{ji})^2}{b_{ji}^2} \right] \tag{3.6}$$

and triangular fuzzy sets

$$A_j(\boldsymbol{x}) = \max\left\{ 0, 1 - \frac{|x - a_{ji}|}{b_{ji}} \right\}, \tag{3.7}$$

with $a_{ji} \in \mathcal{R}$, $b_{ji} > 0$ being the central point and the scaling parameter in the i-th dimension, respectively.

Concerning consequent fuzzy sets, we introduce trapezoidal versions of radial fuzzy sets which allows kernels of output fuzzy sets to be controlled. Formally, the membership functions of trapezoidal Gaussian and triangular

fuzzy sets write as

$$B_j(y) = \exp\left[-\frac{\max\{0, |y - c_j| - s_j\}^2}{d_j^2}\right], \tag{3.8}$$

$$B_j(y) = \max\left\{0, 1 - \frac{\max\{0, |y - c_j| - s_j\}}{d_j}\right\}, \tag{3.9}$$

with $c_j \in \mathcal{R}$ being the central point and $d_j > 0$ the scaling parameter.

The meaning of parameters is best demonstrated on graphs of membership functions presented in Figs. 3.1 and 3.2.

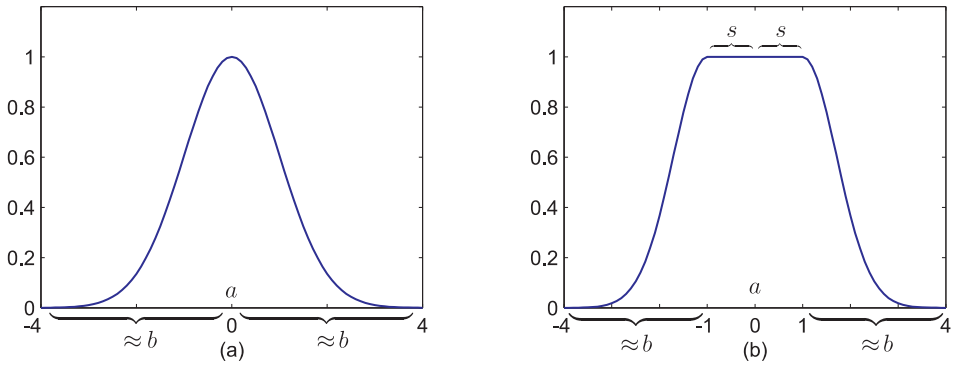

Figure 3.1: An example of Gaussian radial fuzzy sets; (a) an antecedent fuzzy set; (b) a consequent trapezoidal fuzzy set.

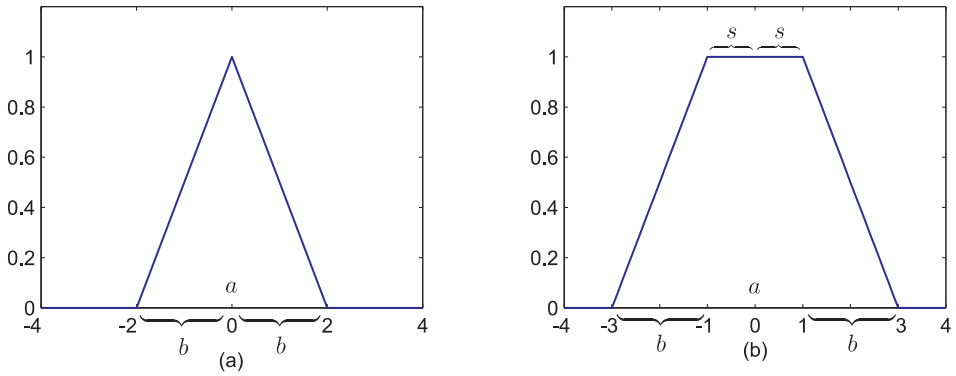

Figure 3.2: An example of triangular radial fuzzy sets; (a) an antecedent fuzzy set; (b) a consequent trapezoidal fuzzy set.

3.1.3 Radial implicative fuzzy systems

Radial implicative fuzzy systems are created by the fusion of introduced concepts of implicative and radial fuzzy systems. The combination is not totally free, but it has a certain structure which gives the birth to the effectiveness of the related computational model.

Let us start with the well known example of combination of Gaussian fuzzy sets by the product t-norm. As it is well known, we again get a Gaussian fuzzy set, but defined on a multi-dimensional space. In other words, by the specific t-norm combination we move from one-dimensional spaces to a multi-dimensional one, but the shape of the set is preserved. This fact substantially simplifies a subsequent mathematical analysis.

The idea of the *shape-preservation* lies behind the study of conditions under which the combination of one-dimensional fuzzy sets by a given t-norm is invariant with respect to a general *act* function. Formally, for a given t-norm $\star$, what the shape *act* of a radial fuzzy set is allowed in order to the following functional equation hold (for the case of the j-th rule):

$$act\left(\frac{|x_i - a_{ji}|}{b_{j1}}\right) \star \cdots \star act\left(\frac{|x_n - a_{jn}|}{b_{jn}}\right) = act(||\boldsymbol{x} - \boldsymbol{a}_j||_{\boldsymbol{b}_j}), \tag{3.10}$$

where $||\cdot||_{\boldsymbol{b}_j}$ is an scaled ℓ_p norm defined by formula (3.5)?

The answer to this question is given by the following theorem proved in [2].

Theorem 1. *Let the t-norm $\star$, which is used to represent* IF-THEN *rules of a fuzzy system, be continuous Archimedean. Then the radial property* (3.10) *holds if and only if the act function has form $act(z) = t^{(-1)}(qz^p)$, where $t^{(-1)}$ is the pseudo-inverse of the additive generator of t-norm $\star$ and $q > 0$, $p \geq 1$ are parameters.*

The theorem tell us how to combine the shape of a radial fuzzy set (in fact how to choose the *act* function) and the corresponding t-norm. We easily find that Gaussian radial functions corresponds to the product t-norm, with the pseudo-inverse [12] of the generator written as $t^{(-1)} = \exp(-z)$, and choosing $p = 2$, $q = 1$. For triangular fuzzy sets the Łukasiewicz t-norm fits, with the pseudo-inverse written as $t^{(-1)} = \max\{0, 1 - z\}$ and choosing $p = q = 1$.

In applications, the employment of the *minimum t-norm* is quite common. The minimum t-norm is not Archimedean so the above theorem does not apply in this case, but we have the radial property still retained as it is shown in the following theorem proved in [1].

Theorem 2. *Let an* I-FS *consisting of m rules be specified as follows*:
1) *The employed t-norm is the minimum t-norm.*
2) *The act function is any non-increasing function from* $[0, +\infty)$ *to* $[0, 1]$.
3) *Fuzzy sets forming antecedents are radial, i.e., they are constituted as* $A_{ji}(x_i) = act(|x_i - a_{ji}|/b_{ji})$.
Then the I-FS *is radial.*

The proof of the theorem is based on the fact that as the *act* function is non-increasing, we have for any $u_i \in [0, +\infty)$,

$$\min\{act(u_1), \ldots, act(u_n)\} = act(\max\{u_1, \ldots, u_n\}). \tag{3.11}$$

Therefore, if we employ the minimum t-norm for an I-FS specification, then we obtain its antecedents in form

$$A_j(\boldsymbol{x}) = \min_i \left\{ act\left(\frac{|x_i - a_{ji}|}{b_{ji}}\right)\right\} = act\left(\max_i \left\{\frac{|x_i - a_{ji}|}{b_{ji}}\right\}\right). \tag{3.12}$$

Since $b_{ji} > 0$ for every j and i, employing the scaled cubic norm $||\boldsymbol{u}||_{C_{\boldsymbol{b}}} = \max_i\{|u_i/b_i|\}$, $\boldsymbol{u} \in \mathcal{R}^n$, the above can be written as $A_j(\boldsymbol{x}) = act(\,||\boldsymbol{x} - \boldsymbol{a}_j||_{C_{\boldsymbol{b}_j}})$, which validates the radial property.

On the basis of this theorem we can see that by employing the minimum t-norm for an I-FS specification, we can assure the validity of the radial property by a suitable, but a very non-restrictive choice of an *act* function.

3.1.4 Computational model

The representation of individual rules and composition of rules in implicative and radial fuzzy systems have been presented in Sections 3.1.1 and 3.1.2. Here we are going to present the computational model of radial I-FSs in details. In order to do this we will review residuated fuzzy implications and their properties.

The residuated fuzzy implications are generalization of the Boolean implication with evaluation extended to the unit interval [8, 13]. Formally, they are operations on the unit square, i.e., $x \to y : [0, 1]^2 \to [0, 1]$, that are derived from corresponding t-norms by the operation of residuation $\{(x \to y) = \sup\{z \,|\, z \star x \leq y\}$. In this form, the residuated implications together with the corresponding t-norms (generalizations of the Boolean conjunction) form the basis of the fuzzy logic in the narrow sense [8].

The most prominent examples of residuated implications are those derived from the product and the minimum t-norms. They are called the Goguen $\to_{\mathrm{P}}$

and Gödel $\rightarrow_{\mathrm{M}}$ fuzzy implications, respectively, and they are specified by the following explicit formulas

$$\begin{aligned}(x \rightarrow_{\mathrm{P}} y) &= 1 \quad \text{for } x \leq y;\ (x \rightarrow_{\mathrm{P}} y) = y/x \quad \text{for } x > y,\\ (x \rightarrow_{\mathrm{M}} y) &= 1 \quad \text{for } x \leq y;\ (x \rightarrow_{\mathrm{M}} y) = \qquad \text{for } x > y.\end{aligned}$$

An important fact here is that if $x \leq y$, then both implication are evaluated by 1, i.e., they are true in the degree one. This is the property of all residuated implications as it can be verified by the inspection of the residuation formula taking into account the boundary condition $1 \star z = z$ and monotonicity of t-norms.

Let $\boldsymbol{x}^* \in \mathcal{R}^n$ be an input to the radial I-FS. The singleton fuzzification together with the application of compositional rule of inference [5, 13] produces the system's output fuzzy set in the form of

$$B(y) = RB(\boldsymbol{x}^*, y) = \max_{j=1}^{m}\{A_j(\boldsymbol{x}^*) \rightarrow B_j(y)\}. \tag{3.13}$$

This output fuzzy sets is composed from m modified consequents fuzzy sets $B_j(y)$, namely from the sets $B_j^*(y) = A_j(\boldsymbol{x}^*) \rightarrow B_j(y)$. If we are searching for the kernel of $B(y)$, i.e., for those ys for which $B(y) = 1$, then, due to the combination by maximum, we find that these are the ys which lie in the intersection of kernels of $B_j^*(y)$. For the elements (outputs) in this intersection we have - under the knowledge stored in the rule base of the system - the perfect, in degree 1, compatibility with the presented input $\boldsymbol{x}^*$.

Let us now search for compatible ys for the j-th rule. From the properties of residuated implications we have the evaluation 1 iff $A_j(\boldsymbol{x}^*) \leq B_j(y)$. In a radial I-FS, due to the radial property, the shape of the antecedent $A_j(\boldsymbol{x})$ is the same as the shape of the consequent B_j and due to the monotonic character of *act* function we obtain the explicit solution of $A_j(\boldsymbol{x}^*) \leq B_j(y)$ inequality in the form of interval

$$I_j(\boldsymbol{x}^*) = [c_j - d_j||\boldsymbol{x}^* - \boldsymbol{a}_j||_{\boldsymbol{b}_j} - s_j,\ c_j + d_j||\boldsymbol{x}^* - \boldsymbol{a}_j||_{\boldsymbol{b}_j} + s_j]. \tag{3.14}$$

That is, for $y \in I_j(\boldsymbol{x}^*)$ we have $A_j(\boldsymbol{x}^*) \leq B_j(y)$ and consequently $B_j^*(y) = 1$. The derivation of this result is presented in [1].

Denoting the left and the right limit point of $I_j(\boldsymbol{x}^*)$ by $L(I_j(\boldsymbol{x}^*))$ and $R(I_j(\boldsymbol{x}^*))$, respectively, we obtain the intersection of $I_j(\boldsymbol{x}^*)$ intervals explicitly expressed as interval

$$I(\boldsymbol{x}^*) = \bigcap_{j=1}^{m} I_j(\boldsymbol{x}^*) = [\max_j\{L(I_j(\boldsymbol{x}^*))\}, \min_j\{R(I_j(\boldsymbol{x}^*))\}], \tag{3.15}$$

under the condition $\max_j\{L(I_j(\boldsymbol{x}^*))\} \leq \min_j\{R(I_j(\boldsymbol{x}^*))\}$, otherwise we have $I(\boldsymbol{x}^*) = \emptyset$.

If $I(\boldsymbol{x}^*) \neq \emptyset$, we have for $y \in I(\boldsymbol{x}^*)$ the degree-1 compatibility with the input $\boldsymbol{x}^* \in \mathcal{R}^n$ and all those ys are good candidates for a single-point output of the system. The most prominent choice for this output is the middle point of $I(\boldsymbol{x}^*)$ interval, which corresponds to the application of the mean-of-maxima (MOM) defuzzification method [13, 17]. Under the MOM method we identify the single-point output $y^* \in I(\boldsymbol{x}^*)$ as

$$y^* = \frac{L(I(\boldsymbol{x}^*)) + R(I(\boldsymbol{x}^*))}{2}. \tag{3.16}$$

The question for when the interval $I(\boldsymbol{x}^*)$ (actually the kernel of $B(y)$ set in (3.13)) is non-empty for all possible inputs $\boldsymbol{x}^* \in \mathcal{R}^n$ is the question for the *coherence of the system* [3, 6]. That is, we would like to be assured that for all possible inputs to the system there is at least one perfectly compatible output. This turns out to the requirement that for any possible input the rules composing the rule base of the system are non-contradictory.

3.1.5 Coherence

As stated, the question of coherence is the question for the non-contradictoriness of rules in the rule base. More specifically, we say that a fuzzy system is *incoherent* if there exists an input $\boldsymbol{x}^* \subseteq \mathcal{R}^n$ to the system such that the output fuzzy set is *not normal*, i.e., for any y in the output space the membership degree to $B^*(y)$ is strictly less than 1. An example of incoherent computation is presented in Fig. 3.3.

Testing for coherence is generally a difficult problem because we actually have to test the character of the output fuzzy set for all possible inputs $\boldsymbol{x} \in \mathcal{R}^n$ to the system. However, in the case of radial I-FSs, due to their computational model, it can be shown that 1) the testing can be reduced to tests on pairs of rules and 2) a sufficient condition can be stated on the basis of checking the intersection of output intervals $I_j(\boldsymbol{x}^*)$. The relevant theorem presented in [3] reads as follows:

Theorem 3. *Let a radial* I-FS *consist of m rules. Let $\alpha_j = 1/\max_i\{b_{ji}\}$, $\alpha_k = 1/\max_i\{b_{ki}\}$ for $= 1, \ldots, m$. If for any pair of rules $j, k \in \{1, \ldots, m\}$ the following holds:*

$$|c_j - c_k| - (s_j + s_k) \;\leq\; \min\{d_j\alpha_j, d_k\alpha_k\} \cdot ||\boldsymbol{a}_j - \boldsymbol{a}_k||, \tag{3.17}$$

then the radial I-FS *is coherent.*

In words, the theorem says that a sufficient condition to a radial I-FS be coherent is that the difference between the central points of consequent

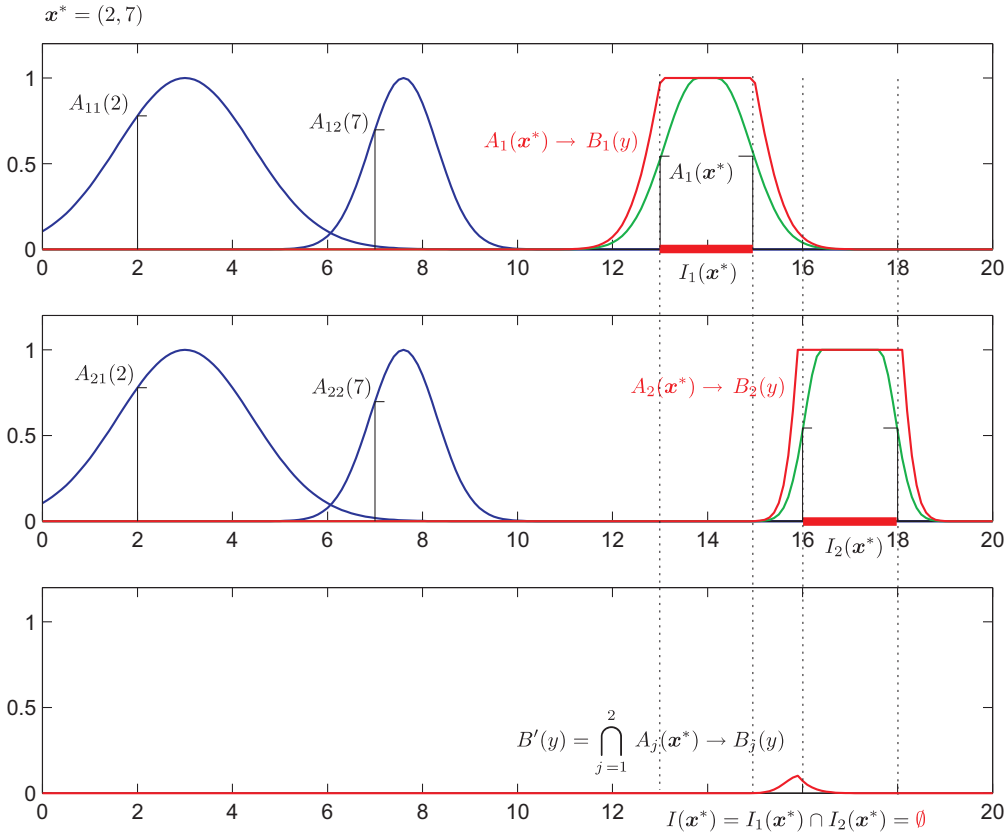

Figure 3.3: An example of incoherent rules in a radial implicative fuzzy system. In the picture, there are presented kernels $I_1(\boldsymbol{x}^*)$ and $I_2(\boldsymbol{x}^*)$ of output fuzzy sets for input $\boldsymbol{x}^* = (2, 7)$.

fuzzy sets is less than the difference between the central points of antecedent fuzzy sets. The antecedent fuzzy sets' width parameters are incorporated into the alpha constants, and this must hold for each pair of rules in the rule base. An intuitive meaning of the incoherence is that rules having similar preconditions state contrary conclusions [5], and this is impossible to happen under the validity of (3.17) inequalities.

3.1.6 Examples of radial I-FSs

Here we introduce two main important examples of radial implicative fuzzy systems. These are the *Gaussian and Mamdani radial I-FSs*. We introduce these systems only briefly here, for more details see [1].

Gaussian radial I-FS

The Gaussian radial I-FS is determined by the product t-norm and the corresponding Goguen residuated implication (if $x \leq y$, then $x \to_{\mathrm{P}} y = 1$; if $x > y$, then $x \to_{\mathrm{P}} y = y/x$). The *act* function is derived from the pseudo-inverse of the additive generator of the product t-norm according to Theorem 1 using parameters $p = 2$ and $q = 1$. That is, $act(z) = \exp(-z^2)$. In order to the system be radial, the employed fuzzy sets are selected to be Gaussians. The membership functions are specified by (3.6) for antecedents and by (3.8) for consequents fuzzy sets. The scaled ℓ_p norm occuring on the right hand side of the radial property definition formula (3.10) is the scaled Euclidean norm, i.e., $||\boldsymbol{u}||_{\boldsymbol{b}} = (\sum_i (u_i/b_i)^2)^{1/2}$.

The computational model is given by formulas (3.14), (3.15) and (3.16), determining the output intervals $I_j(\boldsymbol{x}^*)$ from individual rules, their intersection $I(\boldsymbol{x}^*)$ and the final output y^* from the system, under the assumption that the system is coherent, i.e., that $I(\boldsymbol{x}^*) \neq \emptyset$.

Mamdani radial I-FS

The Mamdani radial I-FS is determined by the minimum t-norm, the Gödel residuated implication (if $x \leq y$, then $x \to_{\mathrm{M}} y = 1$, $x \to_{\mathrm{M}} y = y$ otherwise) and the *act* function given as $act(z) = \max\{0, 1 - z\}$. The employed fuzzy sets are these of (3.7) and (3.9). That is, the triangular ones for antecedents and their trapezoidal versions for consequents.

Such an implicative fuzzy system is radial. The norm on the right hand side of (3.10) is the cubic norm $||\boldsymbol{u}||_{C_{\boldsymbol{b}}} = \max_i\{|u_i|/b_i\}$. This norm is the limit case of the scaled ℓ_p norms for $p \to \infty$. The radial property (3.10) then writes as $\min\{A_{j1}(x_1), \ldots, A_{jn}(x_n)\} = \max\{0, 1 - ||\boldsymbol{x} - \boldsymbol{a}_j||_{C_{\boldsymbol{b}_j}}\}$.

Because the support of triangular fuzzy sets is not the whole real line, it may happen that for a given input $\boldsymbol{x}^* \in \mathcal{R}^n$ the membership degree to an antecedent of a rule is zero, i.e., $A_j(\boldsymbol{x}^*) = 0$. In this case, the output interval of (3.14) is $I_j(\boldsymbol{x}) = (-\infty, +\infty)$ due to the fact that $0 \to_G B_j(y) = 1$ for any $y \in \mathcal{R}$. In real implementations we set $I_j(\boldsymbol{x}) = [-\mathrm{BigNumber}, +\mathrm{BigNumber}]$. Where BigNumber > 0 is a selected constant, e.g., we use BigNumber=10^6 in our MATLAB implementation, see Section 3.3.

If the intersection of individual rules is non-empty, i.e., if the system is coherent, then the output from the Mamdani system is determined by (3.16). Theoretically, it again may happen that $I(\boldsymbol{x}) = (-\infty, +\infty)$ (all rules are non-informative). In this case we set by definition that the output is zero. Practically, using the BigNumber approximation, we obtain the same result, i.e., that $y^* = 0$.

3.2 S-shaped radial implicative fuzzy systems

In this section we introduce an enhancement of the class of radial implicative fuzzy systems. The enhancement is made by allowing not only purely radial fuzzy sets to be presented in the rules of the system, but also their S-shaped versions. The reason for the enhancement is that S-shaped fuzzy sets are commonly used in applications.

3.2.1 S-shaped radial fuzzy sets

S-shaped radial fuzzy sets can be described in words as half-split radial fuzzy sets composed of a radial part and the constant degree-1 part. In our approach these sets are derived from an original radial function taking into account the sign of the difference from the central point. In what follows we will consider the one-dimensional case. If the difference is positive, then an S-shaped fuzzy set looks like the original radial one. If the difference is negative, then the distance is cut to zero, which turns out into the overall S-shape of the set. In fact, this gives what we call the right S-shaped fuzzy set (the degree-1 constant part is on the right, radial part is on the left and the graph of the membership function looks like the proper (right oriented) letter S).

If we reduce the positive difference to zero and apply an *act* function on the negative distance turned to the positive one, we obtain the left S-shaped fuzzy set (the degree-1 constant part is on the left, the radial part on the right, and the membership function looks like the Z letter (the left-mirrored S letter)).

The counterpart of one-dimensional formula (3.4) reads as

$$act\left(\frac{\max\{0, x-a\}}{b}\right) \quad \text{and} \quad act\left(\frac{\max\{0, a-x\}}{b}\right), \tag{3.18}$$

where the first formula corresponds to the left S-shaped set and the other to the right S-shaped set.

Turning out to our examples of Gaussian and triangular radial fuzzy sets, we get the following formulas (first in the left version and the other in the right version)

$$A_{ji}(x_i) = \exp\left[-\frac{\max\{0, x_i - a_{ji}\}^2}{b_{ji}^2}\right], \tag{3.19}$$

$$A_{ji}(x_i) = \max\left\{0, 1 - \frac{\max\{0, a_{ji} - x_i\}}{b_{ji}}\right\}. \tag{3.20}$$

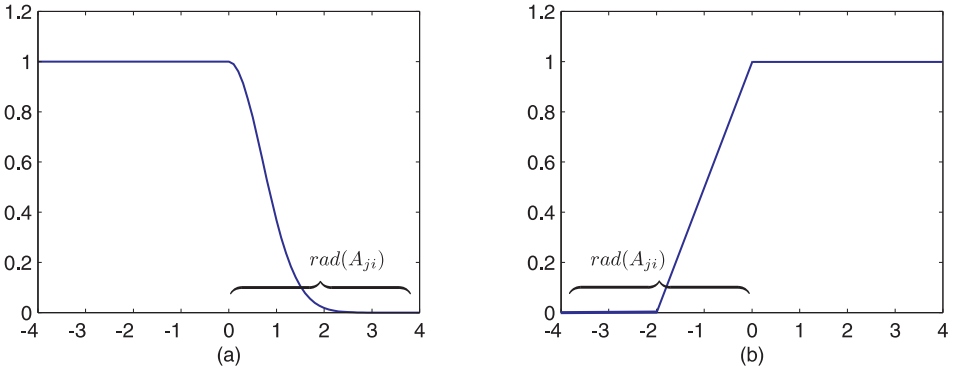

Figure 3.4: An example of S-shaped radial fuzzy sets; (a) a Gaussian left S-shaped fuzzy set (the Z letter); (b) a triangular right S-shaped fuzzy set (the S letter).

Here we do not consider the trapezoidal versions because they are actually incorporated. Figures again clear the concept. In Fig. 3.4. there are delimited radial parts $rad(A_{ji})$ of S-shaped fuzzy sets.

3.2.2 Radial property and S-shaped fuzzy sets

The notion of a radial fuzzy set relies on the notion of the scaled ℓ_p norm. The radial property then represents the fact of the shape preservation in the representation of antecedents of IF-THEN rules. In order to set up the radial property for S-radial fuzzy sets we extend the scaled ℓ_p norms to the *scaled* ℓ_p *S-norms*.

Let $\boldsymbol{b} = (b_1, \ldots, b_n)$, $b_i > 0$ be a vector of scaling parameters. Let the symbol $u^{(+-)}$ denote either positive $u^+ = \max\{0, u\}$ or negative $u^- = \max\{0, -u\}$ part of $u \in \mathcal{R}$, or its absolute value $|u|$. By the scaled ℓ_p S-norm of $\boldsymbol{u} \in \mathcal{R}^n$ we denote every expression of form

$$||\boldsymbol{u}||_{s\boldsymbol{b}} = \left(\sum_i \frac{|u_i^{(+-)}|^p}{b_i^p} \right)^{1/p}, \tag{3.21}$$

with parameter $p \geq 1$. In (3.21), the meaning of $u_i^{(+-)}$ is fixed in each dimension, i.e., it is fixed whether $u_i^{(+-)}$ means the positive or the negative part, or the absolute value for each $i = 1, \ldots, n$.

The scaled ℓ_p S-norm $||\boldsymbol{u}||_{s\boldsymbol{b}}$ satisfies the following properties:

1) $||\boldsymbol{u}||_{s\boldsymbol{b}} \geq 0$, $||\boldsymbol{u}||_{s\boldsymbol{b}} = 0$ not only for $\boldsymbol{x} = \boldsymbol{0}$,
2) $||c\boldsymbol{u}||_{s\boldsymbol{b}} = c \cdot ||\boldsymbol{u}||_{s\boldsymbol{b}}$ for $c \geq 0$,
3) $||\boldsymbol{u}||_{s\boldsymbol{b}} + ||\boldsymbol{v}||_{s\boldsymbol{b}} \geq ||\boldsymbol{u} + \boldsymbol{v}||_{s\boldsymbol{b}}$.

We show only the third property, i.e., the triangle inequality. Let $||\boldsymbol{u}||_{\boldsymbol{b}}$ be the scaled norm defined by (3.5). Let $(\boldsymbol{u})_s$ be $\boldsymbol{u}$ modified by the application of corresponding functions $u_i^{(+-)}$ of (3.21) on individual coordinates u_i of $\boldsymbol{u}$. We have $||\boldsymbol{u}||_{s\boldsymbol{b}} = ||(\boldsymbol{u})_s||_{\boldsymbol{b}}$. That is, the scaled ℓ_p S-norm $||\cdot||_{s\boldsymbol{b}}$ is represented using $||\cdot||_{\boldsymbol{b}}$ norm. The triangular inequality for $||\cdot||_{\boldsymbol{b}}$ gives $||\boldsymbol{u}||_{s\boldsymbol{b}} + ||\boldsymbol{v}||_{s\boldsymbol{b}} = ||(\boldsymbol{u})_s||_{\boldsymbol{b}} + ||(\boldsymbol{v})_s||_{\boldsymbol{b}} \geq ||(\boldsymbol{u})_s + (\boldsymbol{v})_s||_{\boldsymbol{b}} \geq ||(\boldsymbol{u} + \boldsymbol{v})_s||_{\boldsymbol{b}} = ||\boldsymbol{u} + \boldsymbol{v}||_{s\boldsymbol{b}}$. The last inequality is due to the fact that for all $u, v \in \mathcal{R}$ we have $u^+ + v^+ \geq (u+v)^+$. Also $u^- + v^- \geq (u+v)^-$.

On the basis of the above properties we see that $||\cdot||_{s\boldsymbol{b}}$ is not a norm nor a semi-norm (the second property holds only for non-negative scalars), but it retains the triangular inequality. Hence it has very similar properties to the standard norm and we use it for the specification of the radial property for S-shaped radial fuzzy sets.

In fact, we only extend the radial property specification formula (3.10) in such a way that it admits also S-shaped radial fuzzy sets to be present. The counterpart of (3.10) writes

$$act\left(\frac{(x_i - a_{ji})^{(+-)}}{b_{j1}}\right) \star \cdots \star act\left(\frac{(x_n - a_{jn})^{(+-)}}{b_{jn}}\right) = act(||\boldsymbol{x} - \boldsymbol{a}_j||_{s\boldsymbol{b}_j}). \quad (3.22)$$

If (3.22) holds, then we say that the j-th rule satisfies the radial property. If it holds for all rules $j = 1, \ldots, m$ in a rule base, then we say that the fuzzy system is radial.

To present an example of an S-shaped radial rule, let us consider the rule with one left S-shaped, one purely radial and one right S-shaped set in the antecedent. The radial property then writes as

$$act\left(\frac{\max\{0, x_1 - a_1\}}{b_1}\right) \star act\left(\frac{|x_2 - a_2|}{b_2}\right) \star \star\ act\left(\frac{\max\{0, a_3 - x_3\}}{b_3}\right) = act(||\boldsymbol{x} - \boldsymbol{a}||_{s\boldsymbol{b}}), \quad (3.23)$$

where for some $p \geq 1$,

$$||\boldsymbol{x} - \boldsymbol{a}||_{s\boldsymbol{b}}^p = \frac{[(x_1 - a_1)^+]^p}{b_1^p} + \frac{|x_2 - a_2|^p}{b_2^p} + \frac{[(x_3 - a_3)^-]^p}{b_3^p}. \quad (3.24)$$

Inspecting the proof of the triangle inequality for $||\cdot||_{s\boldsymbol{b}}$, namely the $||\cdot||_{s\boldsymbol{b}} = ||(\ \cdot\ )_s||_{\boldsymbol{b}}$ identity, we can see that if the standard radial property (3.10) holds for a scaled ℓ_p norm $||\cdot||_{\boldsymbol{b}}$, then (3.22) holds for $||\cdot||_{s\boldsymbol{b}}$, assuming that parameters p and $\boldsymbol{b}$ coincide.

Conversely, let (3.22) hold for a given scaled ℓ_p S-norm $||\cdot||_{s\boldsymbol{b}}$, then also (3.10) holds for the coinciding ℓ_p norm $||\cdot||_{\boldsymbol{b}}$. Indeed, assume that there exists an $\boldsymbol{x}^*$ such that (3.10) does not hold for $||\cdot||_{\boldsymbol{b}}$, then there exists the vector $\boldsymbol{u}^* = (u_1^*, \ldots, u_n^*)$, $u_i^* \geq 0$, such that $act(u_1^*/b_1) \star \cdots \star act(u_n^*/b_n) \neq act(||\boldsymbol{u}^*||_{\boldsymbol{b}})$. Namely, $u_i^* = |x_{ji}^* - a_{ji}|$, $i = 1, \ldots, n$. Now, whatever the value of $\boldsymbol{u}^*$ is, we can always choose $\boldsymbol{x}^0 = (x_1^0, \ldots, x_n^0)$ such that $u_i^* = u_i^*(x_i^0) = (x_i^0 - a_i)^{(+-)}$, where the meaning of $^{(+-)}$ is determined by $||\cdot||_{s\boldsymbol{b}}$ in each dimension. But, for this $\boldsymbol{x}^0$, we get that $act(u_1^*(x_i^0)/b_1) \star \cdots \star act(u_n^*(x_n)/b_n) \neq act(||\boldsymbol{u}^*||_{s\boldsymbol{b}})$, which contradicts the assumption on the validity of (3.22) for all $\boldsymbol{x} \in \mathcal{R}^n$.

3.2.3 Multi-dimensional S-shaped radial fuzzy sets

The previous section presents the description of one-dimensional S-shaped radial fuzzy sets. The section also show us the direction how to describe multi-dimensional versions of these sets. In fact, they correspond to the application of some *act* function on a scaled ℓ_p S-norm presented above.

Let us consider examples in $\mathcal{R}^2$. If we have two purely radial fuzzy sets, then their combination under the radial property brings a two-dimensional radial fuzzy set. For S-shaped radial fuzzy sets, let us present the following two examples of Gaussian fuzzy sets.

In the first example, we have the fuzzy set determined by two S-shaped sets given as $A_{j1}(x_1) = \exp\left[-\frac{\max\{0,2-x_1\}^2}{15}\right]$ and $A_{j2}(x_2) = \exp\left[-\frac{\max\{0,1-x_2\}^2}{5}\right]$. The product combination gives

$$\begin{aligned} A_{j1} \cdot A_{j2} &= \exp\left[-\left(\frac{[(x_1-2)^-]^2}{15} + \frac{[(x_2-1)^-]^2}{5}\right)\right] \\ &= \exp[\ -||(x_1-2)^-, (x_2-1)^-||^2_{(\sqrt{15},\sqrt{5})}\ ]. \end{aligned} \tag{3.25}$$

If only one of the sets is S-shaped and the other is purely radial, e.g., $A_{j1}(x_1) = \exp\left[-\frac{|x_1-2|^2}{15}\right]$ (purely radial) and $A_{j2}(x_2) = \exp\left[-\frac{\max\{0,2-x_2\}^2}{5}\right]$ (S-shaped), we obtain by product

$$\begin{aligned} A_{j1} \cdot A_{j2} &= \exp\left[-\left(\frac{|x_1-2|^2}{15} + \frac{[(x_2-2)^-]^2}{5}\right)\right] \\ &= \exp[\ -||\ |x_1-2|, (x_2-2)^-\ ||^2_{(\sqrt{15},\sqrt{5})}\ ]. \end{aligned} \tag{3.26}$$

Graphically, both sets are presented in Fig. 3.5.

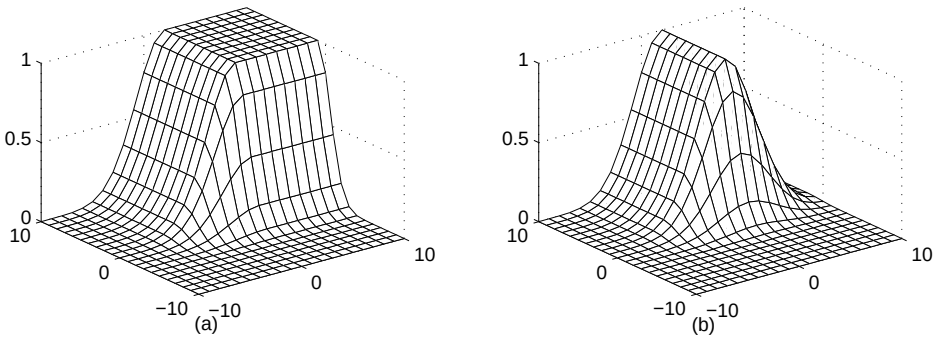

Figure 3.5: An example of two-dimensional S-shaped radial fuzzy sets; (a) a combination of two S-shaped radial fuzzy sets; (b) a combination of one purely radial and one S-shaped radial fuzzy set.

3.2.4 S-shaped radial implicative fuzzy systems

The S-shaped radial implicative fuzzy systems form an extension of purely radial I-FSs which admits an employment of S-shaped radial fuzzy sets in the antecedents of IF-THEN rules.

As we have seen, concerning the radial property when combining one-dimensional fuzzy sets, the only difference with respect to the purely radial systems is the change from the scaled ℓ_p norms to the scaled ℓ_p S-norms.

The other aspects of composition of rules in S-shaped radial I-FSs remain the same. That is, we have in antecedents purely radial or S-shaped radial fuzzy sets (in the left or the right version). These are combined by the appropriate t-norm with respect to the shape function *act* so that the shape-preservation radial property (3.22) holds.

Consequents of rules are set up by trapezoidal versions of purely radial fuzzy sets using the given *act* function.

The combination of IF and THEN parts is performed by an appropriate residuated fuzzy implication to obtain $R_j(\boldsymbol{x}, y)$ relation. The rule base relation $RB(\boldsymbol{x}, y)$ is determined by formula (3.3) and we assume that the radial property (3.22) holds for all rules $j = 1, \ldots, m$, $m \in \mathcal{N}$ forming the rule base.

3.2.5 Computational model

Concerning the computational model of S-shaped radial implicative fuzzy systems, the situation is very similar to the purely radial systems introduced in Section 3.1.4. Again, the only difference is that the scaled ℓ_p norms are replaced by the scaled ℓ_p S-norms. The formula for the j-th output interval

has then form

$$I_j(\boldsymbol{x}^*) = [c_j - d_j||\boldsymbol{x}^* - \boldsymbol{a}_j||_{s\boldsymbol{b}_j} - s_j,\ c_j + d_j||\boldsymbol{x}^* - \boldsymbol{a}_j||_{s\boldsymbol{b}_j} + s_j]. \tag{3.27}$$

The corresponding intersection over all rules writes as

$$I_s(\boldsymbol{x}^*) = \bigcap_j I_j(\boldsymbol{x}^*) = [\max_j\{L(I_j(\boldsymbol{x}^*))\}, \min_j\{R(I_j(\boldsymbol{x}^*))\}]. \tag{3.28}$$

If $I_s(\boldsymbol{x}^*) \neq \emptyset$, then the MOM defuzzifiaction gives the output in the same way as for a purely radial system, i.e.,

$$y_s^* = \frac{L(I_s(\boldsymbol{x}^*)) + R(I_s(\boldsymbol{x}^*))}{2}. \tag{3.29}$$

However, conditions on coherence of S-shaped radial implicative fuzzy systems differ from those for the purely radial ones.

3.2.6 Coherence

The statement of the coherence question for S-shaped radial systems is the same as for the radial ones. We say that an S-shaped radial system is coherent if for any input $\boldsymbol{x}^* \in \mathcal{R}^n$ the output interval $I_s(\boldsymbol{x}^*)$ of (3.28) is non-empty, i.e., $I_s(\boldsymbol{x}^*) \neq \emptyset$. The sufficient condition for coherence of S-radial I-FSs was presented in [4]. It reads as follows:

Theorem 4. *Let an S-shaped radial I-FS consists of m rules. Let for a pair of rules j, k, $AD_{jk} \subseteq \{1, \ldots, n\}$ be the set of those dimensions i for which $A_{ji}(a_{ki}) < 1$ and $A_{ki}(a_{ji}) < 1$ hold simultaneously - the set of admissible dimensions. Let $||\boldsymbol{a}_j - \boldsymbol{a}_k||_{AD_{jk}}$ be the related unscaled ℓ_p norm taken over admissible dimensions. If $AD_{jk} = \emptyset$, then we set $||\boldsymbol{a}_j - \boldsymbol{a}_k||_\emptyset = 0$. If for each pair of rules $j, k \in \{1, \ldots, m\}$ the following holds*

$$|c_j - c_k| - (s_j + s_k) \leq \min\{d_j\alpha_j, d_k\alpha_k\} \cdot ||\boldsymbol{a}_j - \boldsymbol{a}_k||_{AD_{jk}}, \tag{3.30}$$

where $\alpha_j = 1/\max_{i \in AD_{jk}}\{b_{ji}\}$, $\alpha_k = 1/\max_{i \in AD_{jk}}\{b_{ki}\}$, then the S-shaped radial I-FS is coherent.

Let us comment on the theorem, especially on its difference from Theorem 3. First of all, since AD_{jk} is a subset of $\{1, \ldots, n\}$, we have for any unscaled ℓ_p norm $||\boldsymbol{a}_j - \boldsymbol{a}_k||_{AD_{jk}} \leq ||\boldsymbol{a}_j - \boldsymbol{a}_k||$. That is, if an S-shaped radial I-FS is coherent then turning out its S-fuzzy sets into their proper radial versions we get the coherent (purely) radial I-FS. That is, the condition (3.30) is stronger than the condition (3.17), i.e., it is harder to be satisfied.

This result is understandable, in view of the fact that S-shaped radial fuzzy systems constitute the broader class of implicative systems than is the class of the purely radial systems.

3.2.7 Examples of S-shaped radial I-FSs

The description of the S-shaped counterparts of the Gaussian and the Mamdani radial implicative fuzzy systems is straightforward. The S-shaped versions differ only in the specification of antecedents fuzzy sets which can be now also S-shaped. Namely, the fuzzy sets with membership functions of (3.19) in the Gaussian case or (3.20) in the Mamdani case (or their respective right and left versions) are now allowed to be involved in the specification of antecedents of rules of the system.

The computational models are identical to the models for radial systems including the note on using the BigNumber approximation in the case of the Mamdani system.

3.3 MATLAB implementations

In this section we present the software implementations of the introduced S-shaped radial implicative fuzzy systems. Namely, we present four scripts. These implement the computation of the Gaussian and Mamdani systems, the coherence check and computation of the scaled ℓ_p norm. The scripts were implemented in the language of the MATLAB computational environment. For details about this environment see `http://www.mathworks.com`.

3.3.1 `lpbnorm.m`

`lpbnorm.m` computes the scaled ℓ_p norms of a group of column vectors. The script is called

`[nU]=lpbnorm(U,B,p)`.

Inputs:

- `U` - is the (n, N) matrix consisting of column vectors for which the scaled ℓ_p norm are computed, e.g., for $\boldsymbol{x}_k \in \mathcal{R}^n$, $k = 1, \ldots, N$ we have $\mathtt{U} = [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T]$.
- `B` - is the matrix of the same type as `U` containing scaling vectors. They are stored as column vectors, e.g., for $\boldsymbol{b}_k \in \mathcal{R}^n$, $k = 1, \ldots, N$ we have $\mathtt{B} = [\boldsymbol{b}_1^T, \ldots, \boldsymbol{b}_N^T]$.
- `p` - is the scalar which determines the value of parameter p of used ℓ_p norm, i.e., $\mathtt{p} \in [1, +\infty]$. The cubic norm is computed for $p = \infty$.

Output:

- `nU` - is the $(1, N)$ row vector consisting of scaled ℓ_p norms of vectors of `X`, i.e., `X` $= [\,||\boldsymbol{x}_1||_{\boldsymbol{b}_1}, \ldots, ||\boldsymbol{x}_N||_{\boldsymbol{b}_N}]$ for $k = 1, \ldots, N$.

Source code: Section A.1.

3.3.2 `sgauss.m`

The `sgauss.m` script implements the computation of the Gaussian S-shaped radial I-FS. The script checks the coherence using the `scohcheck.m` script which is presented below. If the system is not assured to be coherent, then an error message is released and the script is terminated, otherwise the computation is performed following the introduced computational model. The interface of the script is

`[L,R,Y]=sgauss(A,s,B,C,D,S,X)`.

Inputs:

- `A,B,C,D,S` - are matrices of parameters of the Gaussian S-shaped radial I-FS. Having the system consisting of m rules and its input space corresponding to $\mathcal{R}^n$, the composition of the matrices is the following: `A` is the (n, m) matrix containing in columns the transposed vectors $\boldsymbol{a}_j$, i.e., `A` $= [\boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_m^T]$. Similarly, `B` is the (n, m) matrix of form `B` $= [\boldsymbol{b}_1^T, \ldots, \boldsymbol{b}_m^T]$. The parameters c_j, d_j, s_j are stored in the row vectors `C,D,S`, `C` $= [c_1, \ldots, c_m]$, `D` $= [d_1, \ldots, d_m]$, `S` $= [s_1, \ldots, s_m]$.

- `s` - is the (n, m) matrix of elements from the set $\{-1, 0, 1\}$. Each column corresponds to an individual rule. In the j-th column and the i-th row, the number indicates the representation of $u_i^{(+-)}$ symbol in the antecedent of the j-th rule: -1 represents the negative part, 0 the absolute value and 1 the positive part.

- `X` - is the (n, N) matrix consisting of transposed inputs to the Gaussian S-shaped radial I-FS, i.e., `X` $= [\boldsymbol{x}_1^T, \ldots, \boldsymbol{x}_N^T]$.

Outputs:

- `L,R` - are both $(1, N)$ vectors of the left and right limit points of output interval $I_s(\boldsymbol{x}_k^*)$ which is determined according to intersection (3.28) for $k = 1, \ldots, N$.

- `Y` - is the $(1, N)$ vector specified as `Y=0.5*(L+R)`. It actually corresponds to the values of $y^*(\boldsymbol{x}_k)$ for $k = 1, \ldots, N$.

Source code: Section A.2.

3.3.3 `smamd.m`

The script is the analogy of the `sgauss.m` script for the Mamdani S-shaped radial I-FS. Again, the script checks the system for coherence employing `scohcheck.m`. If the system is not assured to be coherent, then an error message is released and the script is terminated, otherwise it performs computation according to the related computational model. The interface of the script is

`[L,R,Y]=smamd(A,s,B,C,D,S,X)`.

Inputs:

- `A,s,B,C,D,S,X` - are the same as above for `sgauss.m`.

Outputs:

- `L,R,Y` - are the same as above but determined by the computational model of the Mamdani S-shaped radial I-FS.

Source code: Section A.3

3.3.4 `scohcheck.m`

This script performs the coherence checking algorithm for the Gaussian and Mamdani S-shaped radial I-FSs. Inputs into the script are parameters of the systems and the output is the Boolean parameter specifying if the system is coherent or possibly incoherent. The script is called

`[STATUS]=scohcheck(A,s,B,C,D,S,p)`.

Inputs:

- `A,s,B,C,D,S` - are the parameters of respective S-shaped radial I-FS.
- `p` - is the parameter determining the scaled ℓ_p norm. Two values are allowed: `p=2` delimiting the Gaussian system or `p=Inf` delimiting the Mamdani S-shaped radial I-FS.

Outputs:

- `STATUS` - is a Boolean scalar. If `STATUS=1`, then the respective S-shaped radial I-FS is coherent. If `STATUS=0`, then the system is possibly incoherent (the sufficient condition).

Source code: Section A.4.

Chapter 4

Experiments

4.1 Traditional models

The modelling was based on determining several critical episodes in the runoff historical data by experts. The same data sets were used in all of our experiments. First we present results obtained by traditional Kinfill model showing rather good performance on the episodes (cf. figures 4.1 and 4.2). It should be noted that for this model a rather detailed additional information about the physical reality of the catchment is needed, while our approaches deal with the time series of runoff and precipitation values.

4.2 Neuro-evolutionary models

Neuro-evolutionary modelling utilized one-layer perceptron networks and local-unit networks trained by the meta-learning procedure described earlier. The results show that it is relatively easy to obtain better results on individual episodes in comparison to the Kinfill model (cf. fig 4.2).

The main problem seems to be able to predict when the model of individual episodes starts to over-train. This will not only decrease the modelling time but also should improve the generalization capabilities. This behavior is demonstrated on the third graph of the fig 4.2, where a model trained on episode 5 is applied to episode 1 data with much worse results. The ballance between generalization capability and over-training seems to be the important factor for proper training in this case.

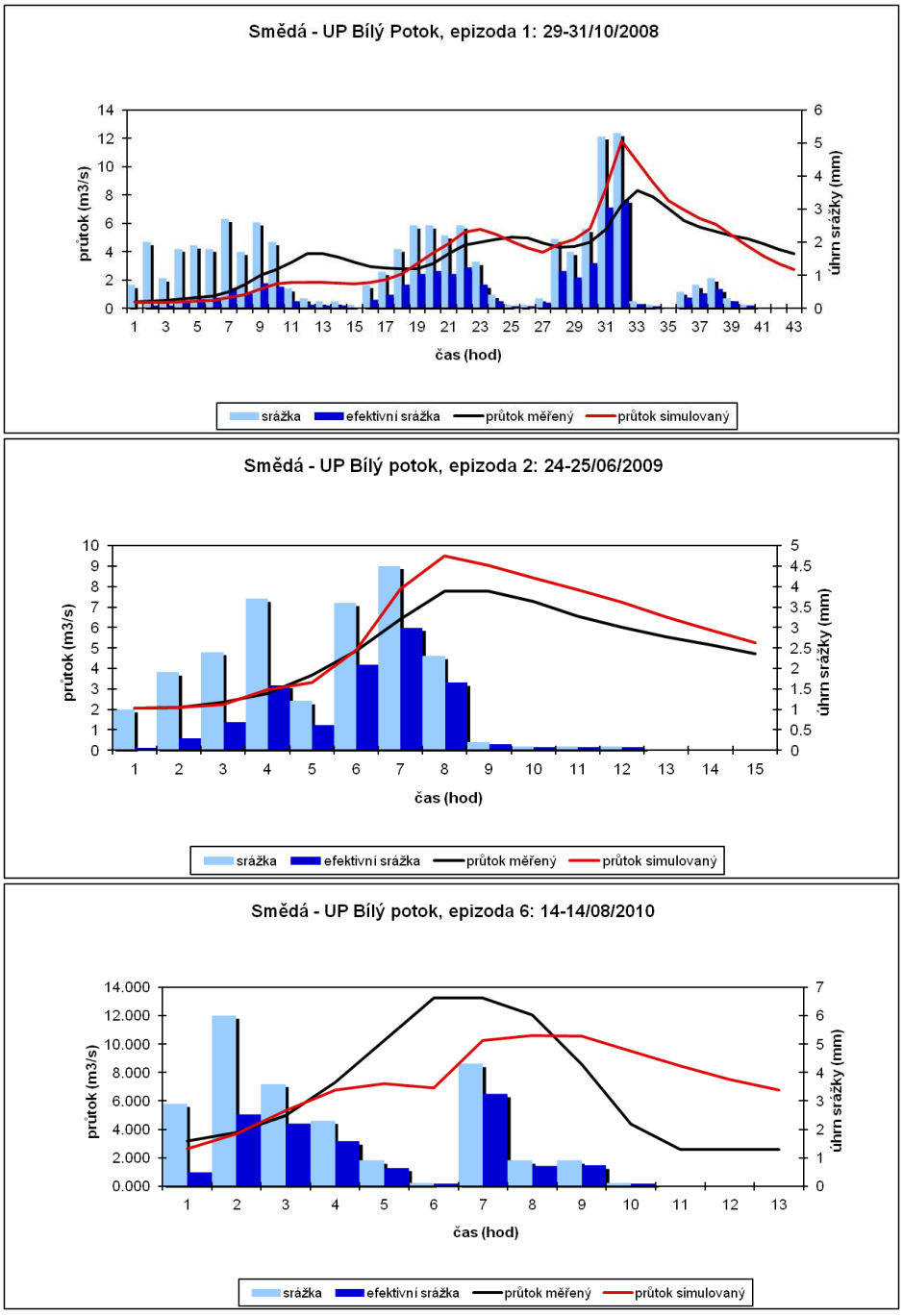

Figure 4.1: Results by the Kinfill model.

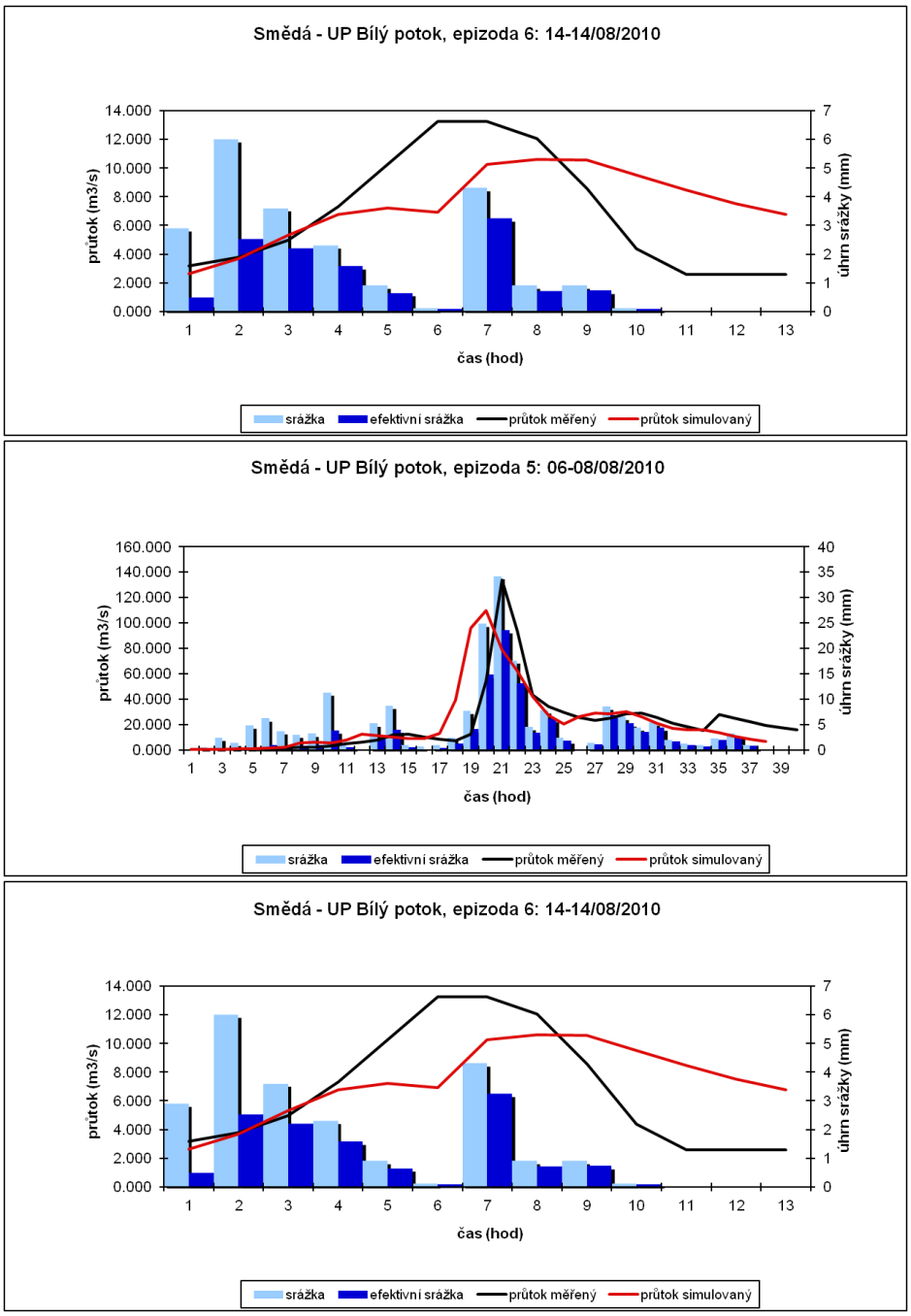

Figure 4.2: Results by the Kinfill model.

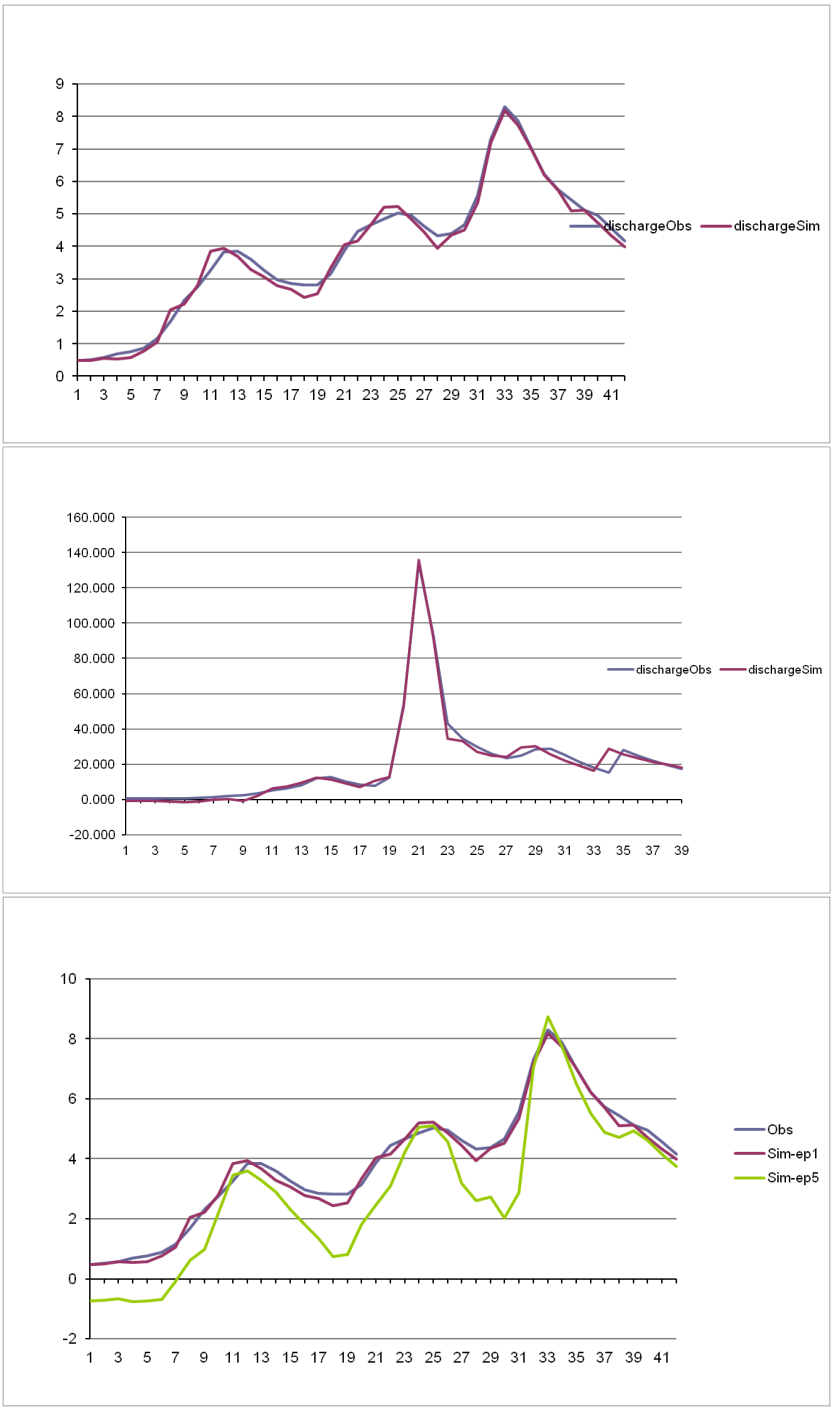

Figure 4.3: Results by the neuro-evolutionary model.

4.3 Fuzzy models

Here we present the results of fuzzy modelling of hydrological environmental data. The data consists of the 6 flood wave episodes on small river Smědá located in Jizera Mountains in Bohemia.

Each episode consists of hourly record of current rainfall called 'excessRain' (eR) and current flow denoted 'dischargeObs' (dO). The task is to predict this actuall flow on the basis of past information, that is on the past values of 'ExcessRain' and past values of 'DischargeObs'. An Example of data is presented in Tab. 4.1.

no.	eR-2	eR-1	ΔeR-12	dO-1	dO-2	ΔdO-12	dischargeObs
1.	0.01	0.08	0.07	0.481	0.511	0.030	0.572
2.	0.08	0.07	-0.01	0.511	0.572	0.061	0.691
3.	0.07	0.19	0.12	0.572	0.691	0.119	0.755
4.	0.19	0.27	0.08	0.691	0.755	0.064	0.876
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 4.1: An excerpt from flood waves data.

The columns presented in the table have the following meaning: eR-2 is the value of 'excessRain' variable prior two hours; eR-1 is the value of 'excessRain' variable prior one hour. ΔeR-12 is the difference of eR-1 and eR-2. Similarly for dO-1, dO-2 and ΔdO-12. The last column contains the values of the actual flow which is to be predicted on the basis of values in other columns.

The distribution of data over individual episodes is summarized in Table 4.2. Graphically, the episodes are presented in Fig. 4.4.

episode	1	2	3	4	5	6
#data (154)	41	13	28	23	38	11

Table 4.2: The data distribution into individual episodes.

In order to build up a predictive fuzzy inference system we have employed the introduced Gaussian and Mamdani S-shaped radial I-FSs. Practically, the developed MATLAB sripts have been used to perform the related computations. In both cases, 5 rules were identified as providing the good level of prediction ability with simplicity retained. The obtained results are presented in the following sections.

4.4 Gaussian and Mamdani fuzzy systems

In the first group of experiments, we employed the Gaussian system with 6 inputs (eR-1, eR-2, ΔeR-12, dO-1, dO-2 and ΔdO-12). For each of the flood episodes we built up the fuzzy system with a different number of rules $m = 2, 3, 4, 5, 6, 7, 8, 9$. Predicted values of the flow were compared with the observed values, and the mean squared error (MSE) was computed according to formula $\frac{1}{n}\sum(\mathrm{dO}_{predicted} - \mathrm{dO}_{observed})^2$, where n is the number of hours constituting the episode.

MSE - G/episode	1	2	3	4	5	6
2-MSE (82.59)	2.37	2.76	4.17	7.27	322.89	2.93
3-MSE (20.23)	2.47	2.33	4.98	7.13	68.78	6.17
4-MSE (52.67)	0.17	0.31	0.38	0.71	211.68	2.72
5-MSE (17.48)	0.84	0.18	0.66	5.80	64.63	4.30
6-MSE (16.95)	0.69	1.11	0.82	2.10	64.39	4.50
7-MSE (15.83)	1.07	2.61	1.70	4.38	55.90	7.92
8-MSE (19.22)	0.73	0.55	0.94	2.72	73.35	4.25
9-MSE (20.03)	2.72	2.30	6.53	9.27	65.17	6.41

Table 4.3: The MSE for the Gaussian fuzzy system.

In Table 4.3, there are presented the obtained results for the Gaussian system. Each row of the table corresponds to the particular number of rules $m = 2, \ldots, 9$. In the first column, in brackets, there is presented the cumulative MSE over all episodes. In the other columns, the MSE for each particular episode is displayed. In each column, the lowest value is underlined.

MSE - M/episode	1	2	3	4	5	6
2-MSE (212.51)	2.61	2.37	2.96	5.86	849.71	7.50
3-MSE (35.70)	1.41	1.51	1.87	4.21	136.90	6.33
4-MSE (109.86)	0.58	0.32	6.36	10.85	427.99	18.09
5-MSE (17.05)	1.13	1.34	0.92	4.24	63.07	3.82
6-MSE (24.85)	0.81	0.92	1.32	4.91	93.36	7.61
7-MSE (55.11)	0.82	0.90	1.38	3.72	217.59	4.51
8-MSE (52.23)	0.99	0.89	1.19	3.44	205.84	5.14
9-MSE (27.40)	0.88	0.87	1.56	5.74	103.37	6.15

Table 4.4: The MSE for the Mamdani fuzzy system.

The same group of experiments was performed with the Mamdani fuzzy system. The results are presented in Table 4.4.

By the inspection of two tables, we see that the Gaussian system performs better in terms of the MSE error and we can choose $m = 5$ as to be the suitable number of rules constituting the system.

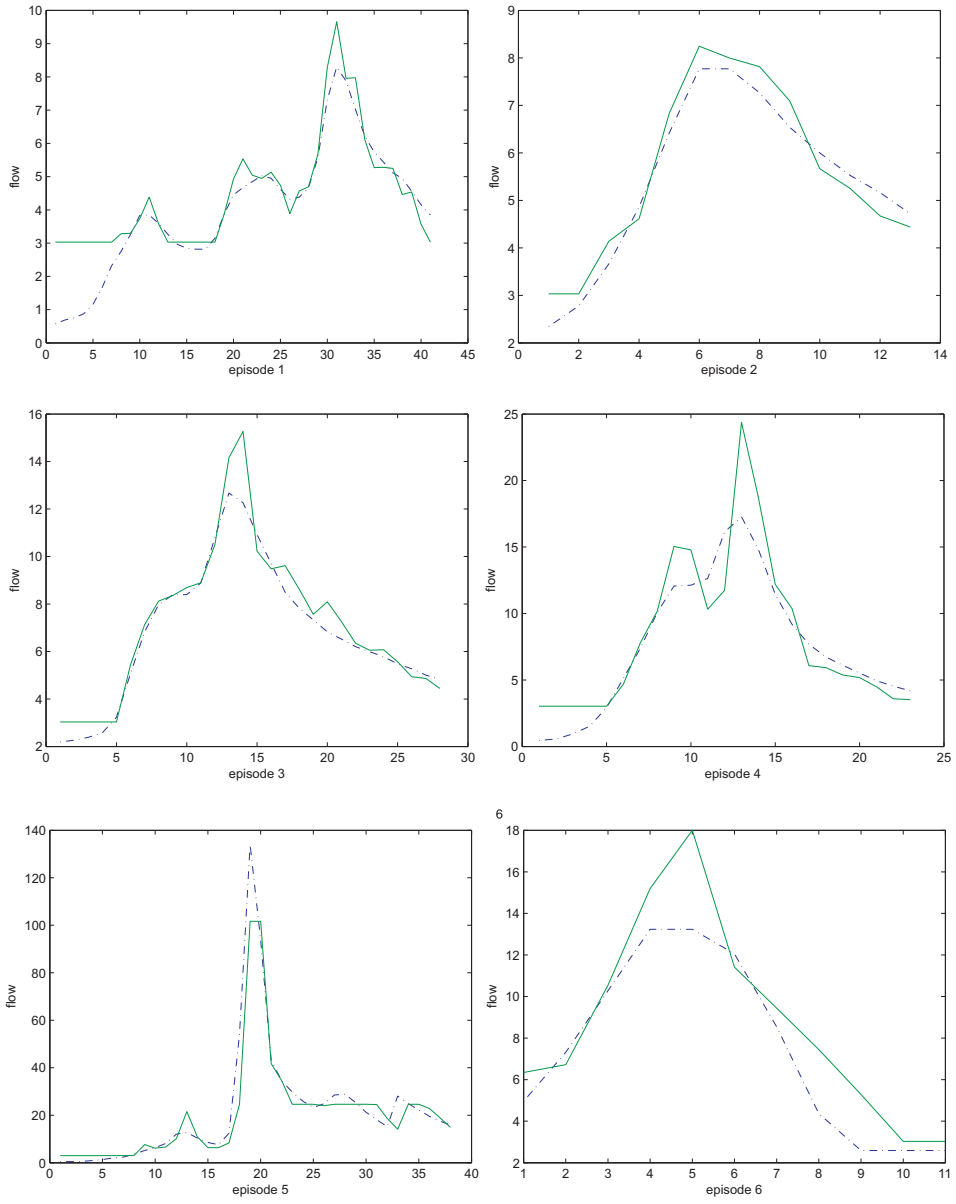

Figure 4.4: Flood waves episodes. The dashed line - real flow. The solid line - the output of the Gaussian system with 5 rules.

Fig. 4.4 shows that the fuzzy modelling has good description capabilities. Concerning the prediction capabilities, the above results bear the drawback that the flow data of each episode were used for establishing the fuzzy system. That is, in the language of machine learning, the testing data were incorporated into the training data. This is inappropriate when testing the predictive capabilities and some type of cross-validation should be further considered and explored.

4.5 Cross-validation

The idea of cross-validation is as follows. First of all we have employed only the Gaussian system because it has lower MSE than the Mamdani system. We have 6 periods. In the cross-validation procedure, we have used as training data all data from all periods except the data from the period which was used for testing. That is, for example, the data from the 1-st period were used to test the system learned on the the data from the 2, 3, 4, 5 and the 6-th period. Similarly, the data from the 2-nd period were used for testing the system learned on data from the 1, 3, 4, 5 and 6-th period. The obtained results are presented in Table 4.5. All experiments were performed for the Gaussian system with 5 rules.

Gauss-CrossVal-ep	1	2	3	4	5	6
5-MSE	1.06	0.09	0.83	2.93	770.72	9.46
Kinfill model	1.48	0.88	5.98	2.89	119.20	12.88

Table 4.5: Comparison of fuzzy and traditional models predicitions.

From the table we see that our results are better than those for the Kinfill model with the exception of the 4-th and 5-th periods. This is good result as our MSE are based on the cross-validation procedure explained above. The results from the Kinfill model are from the experiments where no split to training and unseen testing data was taken. From this point of view the obtained results are promising. Graphically, the predicted flows for individual episodes from the Gaussian S-shaped radial I-FS, which was built up on the basis of unseen data, are presented in Fig. 4.5.

The problem with the 5-th period is clearly seen from the Fig. 4.5. We can see that information acquired from remaining periods was not sufficient to detect flood wave in the 5-th period. The reason is that the acceleration of the wave is much higher in comparison to other periods so if only these are used in learning, then the system cannot acquire this pattern from training data.

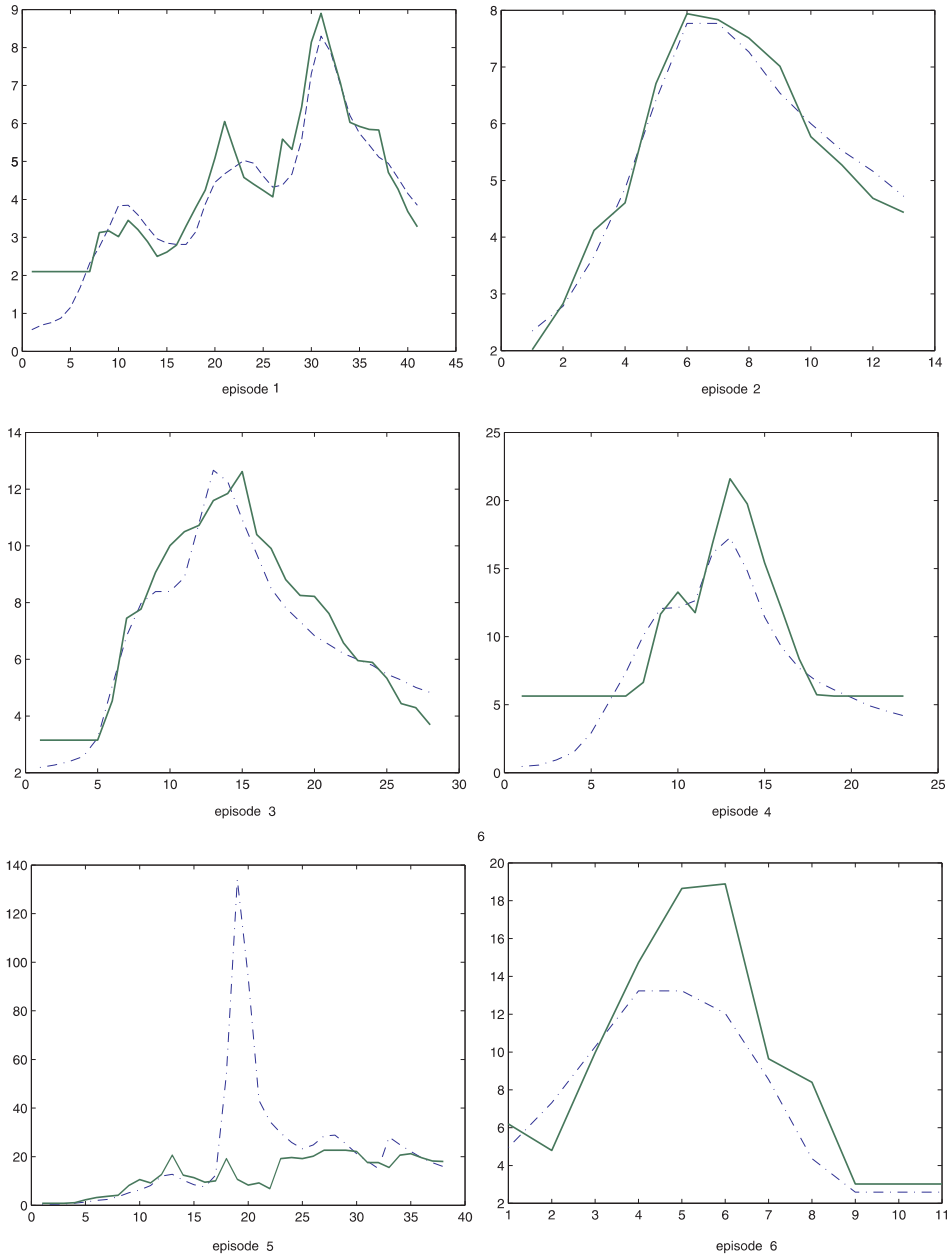

Figure 4.5: The flood waves episodes predictions using the Gaussian S-shaped radial I-FS. The dashed line represents the real flow and the solid one the prediction.

On the other hand, as we have seen when all data are used for learning, the fuzzy system is capable to accommodate such patterns and fuzzy modelling represents a suitable alternative to the Kinfill model.

Chapter 5

Conclusions

We have applied two new methodologies for data mining to a critical infrastructure data of the flood prediction problem. It has been shown that our approaches yield competitive results while needing much less input data and generally less computational time.

Both our approaches require only the time series data and can provide results within seconds to minutes of training time. Several problems for further research have been identified. Concerning the neural models, the main problem seems to be able to predict when the model starts to over-train. This will not only decrease the modelling time but also should improve the generalization capabilities.

Concerning fuzzy modelling, we have found that S-shaped radial fuzzy implicative systems are capable to accommodate the flood waves modelling. This was shown when all available data were used to built up the fuzzy system. In the case of the proper predictive design, we have demonstrated that the fuzzy system provides, on average, better results than the traditional Kinfill model. The only exception was the identification of the 5-th flood wave episode on the basis of data from other episodes. The reason for this inappropriate behavior is that the character of the flood waves in episodes 1-4 and 6 is different from the one in the 5-th episode. Namely, the 5-th wave accelerates faster with much higher peak than it is the case in the other episodes. Hence, in order to such the pattern could be predicted, the presence of this type of pattern is desired in training data.

An advantage of IF-THEN representation of knowledge in a fuzzy system is that adding new information is easily done by adding new rules into the rule base. More specifically, let $RB_1(\boldsymbol{x}, y)$ be a rule base of a fuzzy system built up on the basis of data available at time t_1. At time t_2, when new data come, let $RB_2(\boldsymbol{x}, y)$ be the rule base of the system built up on the basis of new data. Combination of these systems (two pieces of knowledge)

is done in a straightforward way according to formula (3.3). That is, the new rule base is given as $RB_1(\boldsymbol{x}, y) \wedge RB_2(\boldsymbol{x}, y) = \min\{RB_1(\boldsymbol{x}), y), RB_2(\boldsymbol{x}), y)\}$, however, the consistency of the knowledge should be assured. That is, the coherence check for rules of the combined system must be performed so that the combined knowledge is non-contradictory.

When fuzzy models are built up only on the basis of past data, i.e., when no expert knowledge is exploited, then the prediction capabilities are critically conditioned on information carried by training data. In our concrete application, the data are sparse with respect to the complexity of various flood waves profiles, and various profiles in training data are required to improve the performance of the presented system. In fuzzy models the information on new flood wave profiles can be added sequentially, as explained above, and there is the tool how to check the coherence of the combined knowledge.

Bibliography

[1] D. Coufal. Radial Implicative Fuzzy Systems. In *The IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005, Reno, Nevada, USA 2005*, pages 963–968, 2005.

[2] D. Coufal. Representation of Continuous Archimedean Radial Fuzzy Systems. In *International Fuzzy Systems Association World Congress IFSA 2005, Beijing, China 2005*, pages 1174–1179, 2005.

[3] D. Coufal. Coherence of Radial Implicative Fuzzy Systems. In *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2006, Vancouver, Canada 2006*, pages 903–970, 2006.

[4] D. Coufal. A sufficient condition on coherence of S-shaped radial implicative fuzzy systems. In *WCCI 2012 IEEE World Congress on Computational Intelligence, Brisbane, Australia 2012*, 2012.

[5] D. Driankov, H. Hellendoorn, and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer-Verlag, Berlin Heidelberg, 1993.

[6] D. Dubois, H. Prade, and L. Ughetto. Checking the Coherence and Redundancy of Fuzzy Knowledge Bases. *IEEE Transactions on Fuzzy Systems*, 5(3):398–417, 1997.

[7] J. Ferber, O. Gutknecht, and M. Fabien. From agents to organizations: An organizational view of multi-agent systems. In P. Giorgini et al., editors, *AOSE 2003*, number LNCS 3950, pages 214–230, 2004.

[8] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publishers, 1998.

[9] Mark Hall et al. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[10] S. S. Haykin. *Neural Networks: A Comprehensive Foundation, 2nd Edition*. Prentice Hall, Upper Saddle River, 1998.

[11] Ondřej Kazík, Klŕa Pešková, Martin Pilát, and Roman Neruda. Meta learning in multi-agent systems for data mining. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, 2:433–434, 2011.

[12] E. P. Klement, R. Mesiar, and A. Pap. *Triangular Norms*. Kluwer Academic Publishers, Dordrecht, 2000.

[13] G. J. Klir and B. Yuan. *Fuzzy sets and Fuzzy logic - Theory and Applications*. Prentice Hall, Upper Saddle River, 1995.

[14] Roman Neruda and Gerd Beuster. Toward dynamic generation of computational agents by means of logical descriptions. *International Transactions on Systems Science and Applications*, pages 139–144, 2008.

[15] Roman Neruda and Ondřej Kazík. Role-based design of computational intelligence multi-agent system. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, MEDES '10, pages 95–101, 2010.

[16] K. Pešková. Ontological description of intelligent agents, 2010.

[17] L. X. Wang. *A Course in Fuzzy Systems and Control*. Prentice Hall, 1997.

Appendix A

MATLAB source codes

Here we present source codes of MATLAB scripts introduced in Section 3.3.

A.1 lpbnorm.m

```
1  function [nU]=lpbnorm(U,B,p);
2
3   [n,N]=size(U);
4
5   U=abs(U./B);
6   if n==1,
7    nU=U;
8   elseif (n>1)&(p==Inf),
9    nU=max(U);
10  elseif (n>1)&(p~=Inf),
11   nU=sum(U.^p).^(1/p);
12  else
13   disp('bad lpb norm parameters');
14  end;
```

A.2 sgauss.m

```
1  function [L,R,Y]=sgauss(A,s,B,C,D,S,X);
2
3   p=2;
4
5   [n,N]=size(X);
6   [n,m]=size(A);
7
8   if m<2, error('only one rule in the Gaussian SrI-FIS'); end;
9
10
11  %--- coherence check ---
```

```
12
13    [STATUS]=scohcheck(A,s,B,C,D,S,p);
14
15    if STATUS~=1,
16     error('the Gaussian SrI-FS is possibly incoherent');
17    end;
18
19
20    %--- computation ---
21
22    nk=zeros(m,N);
23    Aj=zeros(m,N);
24    Lj=zeros(m,N);
25    Rj=zeros(m,N);
26
27    for j=1:m,
28     XA=X-A(:,j)*ones(1,N);
29     for i=1:n,
30      if s(i,j)== 1,XA(i,:)=max(0, XA(i,:)); end;
31      if s(i,j)==-1,XA(i,:)=max(0,-XA(i,:)); end;
32     end;
33     nk(j,:)=lpbnorm(XA,B(:,j)*ones(1,N),p);
34     Aj(j,:)=exp(-nk(j,:).^2);
35     Lj(j,:)=C(j)-D(j)*nk(j,:)-S(j);
36     Rj(j,:)=C(j)+D(j)*nk(j,:)+S(j);
37    end;
38
39    L=max(Lj);
40    R=min(Rj);
41    Y=0.5*(L+R);
42
43    %--- coherence verification ---
44
45    if min(R-L)<0,
46     disp('incoherent computation of the Gaussian SrI-FS');
47    end;
```

A.3 smamd.m

```
1   function [L,R,Y]=smamd(A,s,B,C,D,S,X);
2
3    p=Inf; Inf_w=10e+6;
4
5    [n,N]=size(X);
6    [n,m]=size(A);
7
8    if m<2, error('only one rule in the Mamdani SrI-FS'); end;
9
```

```
10
11   %--- coherence check ---
12
13   [STATUS]=scohcheck(A,s,B,C,D,S,p);
14
15   if STATUS~=1,
16    error('the Mamdani SrI-FS is possibly incoherent');
17   end;
18
19
20   %--- computation ---
21
22   nk=zeros(m,N);
23   Aj=zeros(m,N);
24   Lj=zeros(m,N);
25   Rj=zeros(m,N);
26
27   for j=1:m,
28    XA=X-A(:,j)*ones(1,N);
29    for i=1:n,
30     if s(i,j)== 1,XA(i,:)=max(0, XA(i,:)); end;
31     if s(i,j)==-1,XA(i,:)=max(0,-XA(i,:)); end;
32    end;
33    nk(j,:)=lpbnorm(XA,B(:,j)*ones(1,N),p);
34    Aj(j,:)=max(0,1-nk(j,:));
35    Lj(j,:)=C(j)-D(j)*nk(j,:)-S(j);
36    Rj(j,:)=C(j)+D(j)*nk(j,:)+S(j);
37    i0=find(Aj(j,:)==0);
38    if any(i0),
39     Lj(j,i0)=-Inf_w*ones(1,length(i0));
40     Rj(j,i0)=+Inf_w*ones(1,length(i0));
41    end;
42   end;
43
44   L=max(Lj);
45   R=min(Rj);
46   Y=0.5*(L+R);
47
48
49   %--- coherence verification ---
50
51   if min(R-L)<0,
52    disp('incoherent computation of the Mamdani SrI-FS');
53   end;
```

A.4 scohcheck.m

```
1   function [STATUS]=scohcheck(A,s,B,C,D,S,p);
```

```
2
3   % if STATUS=0, then the S-shaped radial I-FS is posibly incoherent
4   % if STATUS=1, then the S-shaped radial I-FS is coherent
5
6   %--- domains check ---
7
8   if min(min(B))<0, error('some element of B is negative');
9   elseif  min(S)<0, error('some element of S is negative');
10  end;
11
12  [n,m]=size(A);
13
14  %--- admissible dimensions ---
15
16  AD=zeros(m,m,n);
17  for j=1:m-1, for k=j+1:m,
18   for i=1:n, if A(i,j)~=A(i,k),
19    if (s(i,j)==0)&&(s(i,k)==0),
20     AD(j,k,i)=1;
21    end;
22    if (s(i,j)==0)&&(s(i,k)==+1),
23     if A(i,j)>A(i,k), AD(j,k,i)=1; end;
24    end;
25    if (s(i,j)==0)&&(s(i,k)==-1),
26     if A(i,j)<A(i,k), AD(j,k,i)=1; end;
27    end;
28    if (s(i,j)==+1)&&(s(i,k)==0),
29     if A(i,k)>A(i,j), AD(j,k,i)=1; end;
30    end;
31    if (s(i,j)==-1)&&(s(i,k)==0),
32     if A(i,k)<A(i,j), AD(j,k,i)=1; end;
33    end;
34    if (s(i,j)==+1)&&(s(i,k)==-1),
35     if A(i,j)<A(i,k), AD(j,k,i)=1; end;
36    end;
37    if (s(i,j)==-1)&&(s(i,k)==+1),
38     if A(i,j)>A(i,k), AD(j,k,i)=1; end;
39    end;
40   end; end;
41  end; end;
42
43
44  %--- S-coherence check ---
45
46  STATUS=1;
47  if m>1,
48   for j=1:m-1, for k=j+1:m,
49     adjk=zeros(n,1);
50     for l=1:n, adjk(l,1)=AD(j,k,l); end;
```

```
51        if max(adjk)==0, RHS=0; end;
52        if max(adjk)>0,
53         njk=lpbnorm(adjk.*(A(:,j)-A(:,k)),ones(n,1),p);
54         alphaj=1/max(adjk.*B(:,j));
55         alphak=1/max(adjk.*B(:,k));
56         RHS=min(alphaj*D(j),alphak*D(k))*njk;
57        end;
58        LHS=abs(C(j)-C(k))-(S(j)+S(k));
59        if LHS>RHS, STATUS=0; end;
60       end; end;
61     end;
```