

Visualization of Evolving Fuzzy Rule-Based Systems

Abstract

Evolving fuzzy systems are data-driven fuzzy (rule-based) systems supporting an incremental mode of model adaptation in dynamically changing environments; typically, such models are learned on a continuous stream of data in an online manner. This paper advocates the use of visualization techniques in order to help a user gain insight into the process of model evolution. More specifically, *dynamic parallel coordinates*, *vertical parallel coordinates* and *rule chains* are introduced as novel visualization techniques for the inspection of evolving Takagi-Sugeno-Kang (TSK) fuzzy systems. These techniques are realized in the software tool FISVis (Fuzzy Inference System Visualizer), the architecture and functionality of which are presented in this work. To show the usefulness of the proposed techniques, we illustrate their application in the context of learning from data streams with temporal concept drift.

1 Introduction

Visualization has become an important tool in data-driven research fields, such as machine learning and data mining. Apart from the visualization of data objects, data relationships, and aggregated information content [13], the visualization of models learned from data has recently attracted increasing attention in the field of computational intelligence [1, 7, 17, 18]. Going beyond the presentation of static models, this paper suggests the use of visualization techniques for tracking *evolving* models, that is, models that are learned and adapted in an online manner on a continuous stream of data [2, 4, 8].

More specifically, we developed an interactive visualization tool called FISVis, which is short for Fuzzy Inference System Visualizer, that allows for monitoring evolving Takagi-Sugeno-Kang (TSK) fuzzy inference systems in real time [3]. *Temporal dynamic parallel coordinates* and *vertical parallel coordinates* arranged in a time line are proposed for visualizing the evolution of selected rules and the interaction between such rules (in the form of merging and splitting processes). Moreover, so-called *rule chains* are proposed for visualizing changes of a complete rule system between two consecutive time points. Experimentally, we show that characteristic patterns emerge in a rule chain visualization when applied to evolving rule systems learned from data streams exhibiting concept drift.

The paper is structured as follows. Prior to introducing our main visualization techniques, we revisit the underlying TSK fuzzy rule-based systems in Section 2 and discuss some key

challenges arising in the visualization of such systems in Section 3. By way of background, some important notations for measuring the similarity of fuzzy rules and rule systems are recalled in Section 4. A general outline of the architecture and functionality of FISVis is then given in Section 5. In addition to the visualization techniques already mentioned, FISVis provides a kind of monitoring system which is introduced in Section 6. The visualization techniques themselves are then introduced, respectively, in Sections 7, 8 and 9. In Section 10, an experimental study with synthetic data is presented, showing the usefulness of our visualization techniques for analyzing the evolution of fuzzy models learned on data streams with concept drift. The paper ends with a summary and concluding remarks in Section 11.

2 TSK fuzzy rule-based systems

Although the techniques proposed in this paper are more generally applicable, we shall subsequently focus on the model class of first order TSK fuzzy rule-based systems. TSK systems of that kind are comprised of a set of rules $\mathfrak{R} = \{R_1, \dots, R_N\}$. Such a system implements a mapping of the form

$$f : \mathbb{R}^p \rightarrow \mathbb{R}, \mathbf{x} \mapsto y(\mathbf{x}) \quad \text{with} \quad y(\mathbf{x}) = \sum_{i=1}^N l_i(\mathbf{x}) \cdot \Psi_i(\mathbf{x}) . \quad (1)$$

Here, the antecedent of each rule R_i is modeled as a conjunction of p fuzzy sets with Gaussian membership function, each characterized by its center $c_{i,j}$ and width $\sigma_{i,j}$; for a given input vector $\mathbf{x} = (x_1, \dots, x_p) \in \mathbb{R}^p$, the relevance (“firing strength”) of an antecedent part is evaluated as follows:

$$\Psi_i(\mathbf{x}) = \frac{\exp \left[-\frac{1}{2} \sum_{j=1}^p ((x_j - c_{i,j})^2 / \sigma_{i,j}^2) \right]}{\sum_{k=1}^N \exp \left[-\frac{1}{2} \sum_{j=1}^p ((x_j - c_{k,j})^2 / \sigma_{k,j}^2) \right]} \quad (2)$$

The conclusion of R_i is a linear function specified by a $(p + 1)$ -dimensional weight vector $\mathbf{w}_i = (w_{i,0}, w_{i,1}, \dots, w_{i,p})$:

$$l_i(\mathbf{x}) = w_{i,0} + w_{i,1}x_1 + w_{i,2}x_2 + \dots + w_{i,p}x_p \quad (3)$$

Learning TSK models of that kind on a continuous (and potentially unbounded) stream of data $(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \mathbf{z}^{(3)}, \dots)$ in the form of input/output tuples $\mathbf{z}^{(t)} = (\mathbf{x}^{(t)}, y^{(t)})$ essentially means applying a learning algorithm \mathcal{A} that adapts the current rule model after each newly observed example (or, more generally, after a small batch of new examples). Thus, starting with a model $\mathfrak{R}^{(0)}$ at time $t = 0$, a corresponding sequence of models $(\mathfrak{R}^{(0)}, \mathfrak{R}^{(1)}, \mathfrak{R}^{(2)}, \dots)$ is produced, where $\mathfrak{R}^{(t)} = \mathcal{A}(\mathfrak{R}^{(t-1)}, \mathbf{z}^{(t)})$ is obtained by modifying certain rules in $\mathfrak{R}^{(t-1)}$, by merging different rules into a single new rule, or by creating a new rule from scratch.

For a proper handling of rules, it is important that all rules have a unique identifier (rule-ID), no matter whether freshly generated or originating from a merging process. A rule R in $\mathfrak{R}^{(t)}$ which evolved from a rule R' in $\mathfrak{R}^{(t-1)}$ with rule-ID id keeps the same rule-ID. A history \mathfrak{H}_{id} is a sequence of rules all sharing the rule-ID id .

3 Challenges in the visualization of evolving systems

The visualization of an evolving fuzzy system poses a number of challenges, both of conceptual and technical nature. Preceding a more elaborate discussion in the following sections, this section is meant as a brief outline of some key issues to be considered in this regard. To this end, it is useful to look at the visualization problem on different levels of complexity:

- the visualization of a single rule;
- the visualization of a fuzzy system consisting of a set of rules;
- the visualization of an evolving fuzzy system, i.e., a continuous sequence of sets of rules.

A single rule is characterized by its antecedent (2) and its conclusion (3). Geometrically, the former is naturally represented by an ellipsoid, while the latter is a linear function. The problem in both cases is the possibly high dimensionality p : As soon as $p > 3$, a direct representation is not longer possible. Instead, the dimensionality of the objects needs to be reduced in one way or the other, for example using dimensionality reduction techniques such as PCA or ICA [12, 10].

As for a set of rules, different visualization techniques highlight different aspects of the rule systems. Often, one is interested in the relationship between the rules, as represented, for example, in terms of their spatial positions and distances [1, 7, 18], the overlap of neighboring rules [7] or the interaction between rules during inference [1]. Gabriel *et al.* [7] and Rehm *et al.* [18] make use of multidimensional scaling in order to project the high-dimensional system to a two-dimensional plane, trying to maintain the pairwise spatial distances between rules as much as possible. As an important disadvantage of such techniques, note that they produce *global* transformations, which do not necessarily preserve parts of the system (e.g., single rules) that remain unchanged.

In the case of evolving systems, we have to consider not only a single rule system \mathfrak{R} but a growing sequence of systems $\mathcal{E} = (\mathfrak{R}^{(0)}, \mathfrak{R}^{(1)}, \mathfrak{R}^{(2)}, \dots)$, possibly with a varying number of rules. A single rule $R_i^{(t)}$ can then be considered within two different contexts, a spatial one (the rule system $\mathfrak{R}^{(t)}$) and a temporal one (its history \mathfrak{H}_i). In both contexts, the rule has relationships with other rules that can be considered for visualization. Needless to say, representing both contexts simultaneously is not only challenging from a technical point of view but also demanding from a user's perspective. A possible solution is to enhance the visualization by a dynamic component, namely to enable the user to “navigate” in time while restricting the visualization itself to the spatial context.

Since our main aim is to visualize the evolution of a fuzzy model over time, the histories \mathfrak{H}_i of a rule are more interesting for us than the spatial neighborhoods $\mathfrak{R}^{(t)}$. Correspondingly, the visualization techniques to be introduced in Sections 7, 8 and 9 will focus more on \mathfrak{H}_i than on $\mathfrak{R}^{(t)}$.

Apart from the challenges discussed above, a few technical problems need to be solved. A learning algorithm \mathcal{A} incrementally produces rule systems $\mathfrak{R}^{(0)}, \mathfrak{R}^{(1)}, \mathfrak{R}^{(2)}, \dots$ but does not necessarily provide explicit information about the history \mathfrak{H}_i of a rule R_i , i.e., about the connection between rules at different points of time. The task, then, is to “track” a rule R_i from its first appearance till its disappearance. One possibility to do so is to build the history using a functionality offered by the algorithm \mathcal{A} , namely a functionality that provides information about relations between the rules in adjacent sets $\mathfrak{R}^{(t-1)}$ and $\mathfrak{R}^{(t)}$. Evidently, this requires access to the implementation of \mathcal{A} and necessitates a separate implementation of the rule tracking functionality for every new learning algorithm.

Another possibility, which is independent of the algorithm \mathcal{A} , is to extract the history from the sequence of rule systems $\mathfrak{R}^{(0)}, \mathfrak{R}^{(1)}, \mathfrak{R}^{(2)}, \dots$. Obviously, this approach is more general, as it allows for considering the algorithm \mathcal{A} as a black box. On the other hand, it requires a technique for solving a non-trivial “assignment problem”, namely for finding the relationships between rules at different points of time: Which rule at time t corresponds to which other rule at time $t + 1$?

4 Similarity and distance measures for fuzzy rules

As will be seen in subsequent sections, visualization techniques for evolving fuzzy systems typically require a proper definition of similarity or distance between rules. In fact, such measures provide a basis for visualizing relations between individual rules and, therefore, the evolving process of rules and rule systems. Furthermore, they are essential for the monitoring system introduced in Section 6.

4.1 Antecedence similarity

Recall that the antecedence part M_k of a rule R_k is a conjunction of fuzzy sets $\mu_{k,i}$ with normalized Gaussian membership function, one for each input variable x_i (cf. Section 2). We define the similarity between two antecedence parts M_k and M_l of two rules R_k and R_l by

$$\mathfrak{S}(M_k, M_l) = \min \left(\mathfrak{s}(\mu_{k,1}, \mu_{l,1}), \mathfrak{s}(\mu_{k,2}, \mu_{l,2}), \dots, \mathfrak{s}(\mu_{k,p}, \mu_{l,p}) \right), \quad (4)$$

where $\mathfrak{s}(\mu_{k,i}, \mu_{l,i})$ is a standard similarity between fuzzy sets, namely the size of their intersection (pointwise minimum of membership degrees) normalized by the size of the larger of the two:

$$\mathfrak{s}(\mu_{k,i}, \mu_{l,i}) = \frac{|\mu_{k,i} \cap \mu_{l,i}|}{\max(|\mu_{k,i}|, |\mu_{l,i}|)} \quad (5)$$

The size $|\mu|$ of a Gaussian fuzzy set μ is defined by the area under the membership function. Thus, its computation comes down to solving an integration problem for which no closed-form solution exists. Therefore, we exploit the connection between the cumulative distribution function of the normal distribution and the error function $\operatorname{erf}(z) = 2/\sqrt{\pi} \int_0^z e^{-\tau^2} d\tau$, for

which efficient numerical algorithms are implemented. Thus, the area under a normalized Gaussian function with mean c and standard deviation σ can be efficiently determined as follows:

$$FN_{c,\sigma}(z) = \frac{\sigma\sqrt{2\pi}}{2} \left(1 + \operatorname{erf} \left(\frac{z-c}{\sigma\sqrt{2}} \right) \right) \quad (6)$$

4.2 Distance between the rule centers

Another similarity measure based on rule antecedence parts is the distance between rule centers, where the center of a rule R_k is defined as

$$c_k = (c_{k,1}, c_{k,2}, \dots, c_{k,p}) \quad , \quad (7)$$

with $c_{k,i}$ the center of the i^{th} fuzzy set in the antecedence of the k^{th} rule. By using the Euclidean metric, the distance between two rules is

$$\mathfrak{D}(R_k, R_l) = \|c_k - c_l\| \quad . \quad (8)$$

For the purpose of visualization, it is desirable to have the measure normalized to the range $[0, 1]$. In rule chains (see Section 9), the computation of distances is restricted to adjacent rules $R_i^{(t)}$ and $R_i^{(t+1)}$ from the same history \mathfrak{H}_i (the history of the i^{th} rule). Therefore, the normalization is done as follows:

$$\mathfrak{DN}(R_i^{(t)}, R_i^{(t+1)}) = \frac{\mathfrak{D}(R_i^{(t)}, R_i^{(t+1)})}{\max_{(R_i^{(\tau)}, R_i^{(\tau+1)}) \in \mathfrak{H}_i \times \mathfrak{H}_i} \mathfrak{D}(R_i^{(\tau)}, R_i^{(\tau+1)})} \quad (9)$$

4.3 Angle similarity

The angle similarity is a measure of similarity between the conclusion parts of two rules. The conclusion part (3) of a TSK fuzzy rule R_i defines a p -dimensional hyperplane $U_i = \{(x_1, \dots, x_n, l_i(\mathbf{x})) \mid \mathbf{x} \in \mathbb{R}^n\}$ with normal vector $\mathbf{v}_i = (w_{i,1}, \dots, w_{i,p}, -1)$. Using this normal vector, we can compute the angle between two conclusions U_i and U_j by

$$\alpha(U_i, U_j) = \arccos \left(\frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\|} \right) \quad , \quad (10)$$

and finally the corresponding rule angle similarity [16, 15] as

$$\mathfrak{S}_\alpha(R_i, R_j) = \begin{cases} 1 - \frac{2}{\pi}\alpha & \text{if } \alpha \leq \frac{\pi}{2} \\ \frac{2}{\pi}(\alpha - \frac{\pi}{2}) & \text{else} \end{cases} \quad . \quad (11)$$

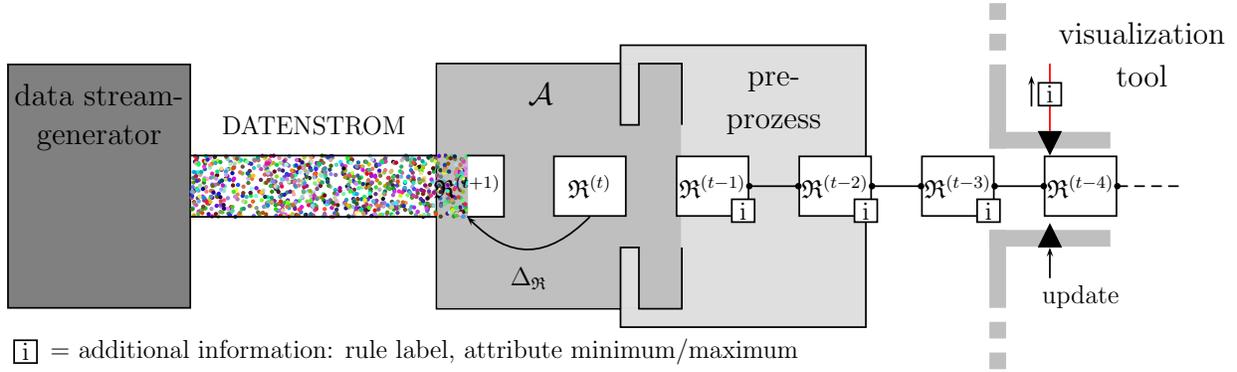


Figure 1: Illustration of the rule system visualization pipeline.

4.4 Conclusion intercept difference

In addition to the angle similarity of two conclusions, we also consider the difference in their y -intercept, which is not captured by this similarity. Although this difference might indeed be negligible on a global scale, one should keep in mind that the influence of a conclusion is localized by the rule antecedence; and locally, the constant term may clearly make a difference. Therefore, we define the conclusion y -intercept difference by

$$\mathfrak{D}_i \left(R_i^{(t)}, R_i^{(t+1)} \right) = \frac{\left| w_{i,0}^{(t)} - w_{i,0}^{(t+1)} \right|}{\max_{(R_i^{(\tau)}, R_i^{(\tau+1)}) \in \mathfrak{S}_i \times \mathfrak{S}_i} \left| w_{i,0}^{(\tau)} - w_{i,0}^{(\tau+1)} \right|}, \quad (12)$$

with $w_{i,0}$ denoting the y -intercept of the conclusion of the rule R_i .

5 System architecture

In this section, the general system architecture of our visualization tool FISVis (Fuzzy Inference System Visualizer) will be presented. We used FLEXFIS [14] as a learning algorithm for evolving fuzzy rule-based system, to which we added a functionality for tracking rules R_i over time.

5.1 Rule system visualization pipeline

The information flow from the data generator to the visualization tool is illustrated in the rule system visualization pipeline (RV-pipeline) shown in Figure 1. The data stream $(\mathbf{Z}^{(0)}, \mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \dots)$, where $\mathbf{Z}^{(t)}$ is a batch of examples, flows into the algorithm \mathcal{A} , which produces a sequence of rule systems $(\mathfrak{R}^{(0)}, \mathfrak{R}^{(1)}, \mathfrak{R}^{(2)}, \dots)$ with $\mathfrak{R}^{(t)} = \mathcal{A}(\mathfrak{R}^{(t-1)}, \mathbf{Z}^{(t-1)})$. In a preprocessing step, a tracking system attaches a history label to every rule. Furthermore, minimum and maximum values of the attributes are added, which are used for normalization

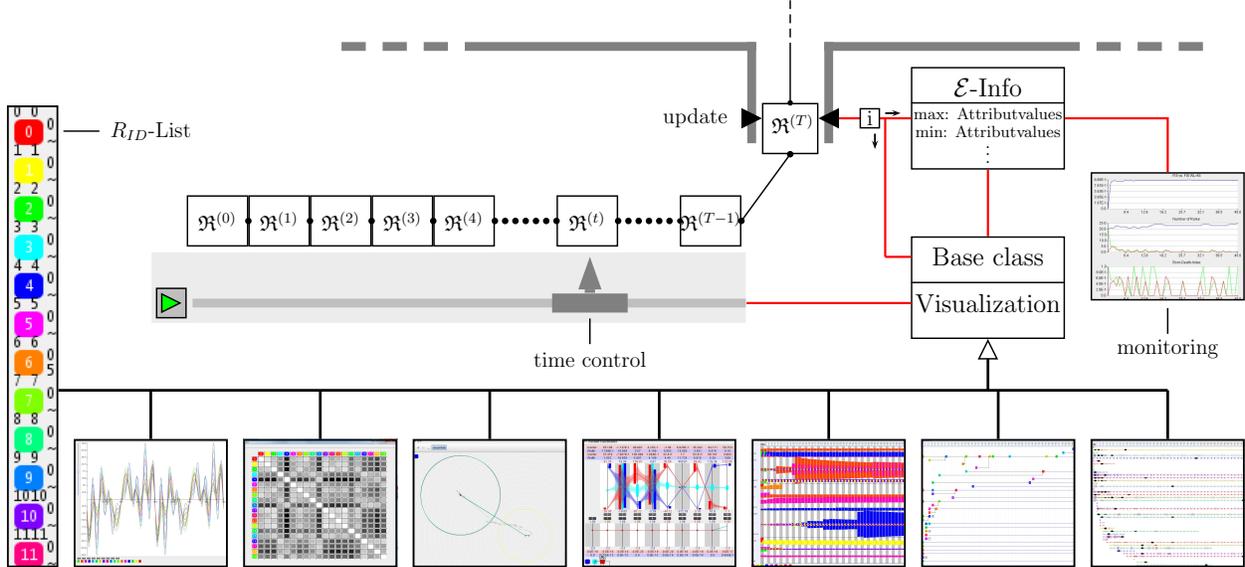


Figure 2: Schematic illustration of the FISVis architecture.

purposes later on. Once the preprocessing step is finished, the rule systems are fed into the visualization tool, where they are collected and trigger an update of the visualization.

5.2 System architecture

The FISVis tool is made up of four main elements (Figure 2): the visualization modules, a list of all histories \mathfrak{H} , a time control component containing a list of all rule systems \mathfrak{R} , and a monitoring system. Technically, the history list is just a list of integers, which represents the histories. Some visualization modules like the temporal dynamic parallel coordinates (Section 7) and the vertical parallel coordinates (Section 8) can be fed by drag-and-drop operations from the history list. The user-friendly drag and drop GUI operation carries the information about which histories in the corresponding module should be displayed, whereby the user is able to arrange a customized set $\mathcal{H}_s \in \mathbb{N}$ of interesting histories (labels) to be analyzed with the corresponding visualization.

Depending on the kind of visualization module (i.e., temporally dynamic or not), the concrete rules are selected from the preprocessed rule systems, which are in turn stored in the rule system list. Every incoming rule system $\mathfrak{R}^{(T)}$ is added to this list. The time control is a pointer to one of the rule systems stored in the list. The rule system $\mathfrak{R}^{(t)}$ identified by the time control represents the current time point and the visualization modules, working with temporally dynamic views, show $\mathcal{R}_s \subseteq \mathfrak{R}^{(t)} \cap (\bigcup_{i \in \mathcal{H}_s} \mathfrak{H}_i)$, whereas \mathcal{R}_s depends on the user selections for the corresponding module.

Two modes are available for the time control: an online mode and scroll mode. In the online mode, the time control always points to the most recent rule system. In the scroll mode,

the time control pointer can be moved forward or backward through the rule system list, which is equivalent to navigation through time. The fourth element is a monitoring system, which is a collection of plots from specific quantities to be discussed in Section 6. Every incoming rule system triggers an update of the plots, thereby delivering a quick overview of the evolving process.

6 System monitoring

A simple method for visualizing an evolving process is plot the curve of a summary statistic that quantifies a specific property of a fuzzy system $\mathfrak{R}^{(t)}$. Simple examples of such statistics include the total number of rules at a certain point of time t , the number of new rules created and the number of rules that vanished at time t . Three further quantities will be introduced in the following.

6.1 Rule coverage

A first measure is the Shannon entropy [19] applied to rule coverage:

$$H = - \sum_{i=1}^N p_i \log_2 p_i \quad (13)$$

with

$$p_i = \frac{1}{|X|} \sum_{k=1}^{|X|} \Psi_i(s_k) \quad , \quad (14)$$

where $\Psi_i(s_k)$ is the firing strength of the i^{th} rule for the k^{th} data point in a sample X (cf. Section 2); correspondingly, p_i is the average firing strength of the i^{th} rule. Thus, according to the properties of the Shannon entropy, H is large if the data is balanced in the sense of being uniformly covered by the rules. As opposed to this, H is small if large portions of the data are represented by only a few rules.

For computing the entropy of a rule system $\mathfrak{R}^{(t)}$, we make use of the training data $X = Z^{(t)}$ from which the rule system was generated (i.e., $\mathfrak{R}^{(t)} = \mathcal{A}(\mathfrak{R}^{(t-1)}, Z^{(t)})$).

6.2 Data coverage

Another useful quantity is obtained by computing the *data coverage*

$$M = \frac{1}{|X|} \sum_{x \in X} \max_{R \in \mathfrak{R}} f_R(x) \quad , \quad (15)$$

where $f_R(x)$ is the non-normalized membership function of rule R . The data coverage measures how well a set of data is covered by the rule system: The higher the data coverage,

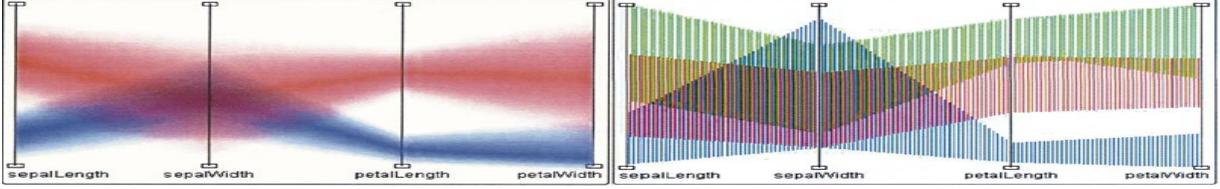


Figure 3: Visualization of fuzzy points in parallel coordinates proposed in [5].

the more samples are well covered by at least one rule. Like in the case of the entropy, the data coverage of the rule system $\mathfrak{R}^{(t)}$ is computed on the basis of the training data $X = Z^{(t)}$.

6.3 System changes

The aim of analyzing the evolution of a rule system is supported by measuring the similarity between consecutive rule systems $\mathfrak{R}^{(t-1)}$ and $\mathfrak{R}^{(t)}$, for example by matching these systems in a bidirectional way:¹

$$BM_S(\mathfrak{R}^{(t-1)}, \mathfrak{R}^{(t)}) = \frac{1}{n+m} \left(\sum_{i=1}^n \max(w_{i,1}, \dots, w_{i,m}) + \sum_{j=1}^m \max(w_{1,j}, \dots, w_{n,j}) \right)$$

where $w_{i,j} = S(R_i^{(t-1)}, R_j^{(t)})$ is the similarity between rules $R_i^{(t-1)}$ and $R_j^{(t)}$, either the antecedence or angle similarity, $n = |\mathfrak{R}^{(t-1)}|$ and $m = |\mathfrak{R}^{(t)}|$.

7 Dynamic visualization of TSK rules using parallel coordinates

Parallel coordinate (PC) plots, reintroduced by Alfred Inselberg [11], are well-known displays for visualizing high-dimensional data. An n -dimensional parallel coordinate system consists of n axes arranged in parallel. A data point is represented as a polygonal line with n station points, each one lying on the corresponding axis. The advantage of parallel coordinates is the ability to represent high-dimensional data in a two-dimensional space.

Berthold and Hall [5] use parallel coordinates to visualize high-dimensional fuzzy points of the form $\mu = \mu_1 \wedge \mu_2 \wedge \dots \wedge \mu_n$. Fuzzy points represented in parallel coordinates do not appear as a polygonal line but as a kind of polygonal tube (Figure 3), where the position and width of the tube is determined by the fuzzy sets μ_i . Using semi-transparent polygons and oscillating lines, different overlapping rules can be discerned.

¹It would arguably be more correct to solve a linear assignment problem [6]. However, since the computation needs to be repeated at every time step, this is computationally too expensive.

Since the antecedence of a fuzzy rule can be seen as a fuzzy point, we adopt the visualization technique of Berthold and Hall for our approach of visualizing evolving fuzzy systems. As our aim is to visualize the evolutionary process, it is important to highlight changes between two systems $\mathfrak{R}^{(t-1)}$ and $\mathfrak{R}^{(t)}$, or locally between two rules $R_i^{(t-1)}$ and $R_i^{(t)}$. Small differences between two rules $R_i^{(t-1)}$ and $R_i^{(t)}$ are difficult to see when being displayed next to each other. Therefore, we interpret the visualizations of all rules in \mathfrak{H}_i as a frame series in a stop motion animation. Changes from $t - 1$ to t appear as a movement, which can be easily noticed by humans. This beneficial property of animation is frequently used when visualizing processes-related data [20]. The FISVis tool enables the user to spin forward and backward in time, so that interesting evolution time points can be analyzed by the user.

7.1 Antecedence visualization

Fuzzy points only contain information about their position and spread in the corresponding dimensions. Thus, there are no formal reasons for drawing any kind of connection between adjacent axes. Instead, the only motivation for specific connections like a tube with piecewise linear edges is to improve readability.

In addition to a crisp linear tube (Figure 4 (a)), we propose a *Bézier tube*, in which edge are defined by Bézier curves (Figure 4(b)). This representation, which is inspired by the work of Zhou *et al.* [21], avoids large overlapping parts between the axes of two or more rules (Figure 5(b)). Thus, a better distinction of different rules is achieved, and the impression of overlapping is reduced.

An even more extreme solution is to represent fuzzy rules as completely disconnected bars (Figure 5(c)). This view is recommended when exploring single dimensions of the rules, or when visualizing rules and data points at the same time (Figure 6). In order to keep overlapping rule parts visible, tubes and bars are displayed in a semi-transparent manner. Bars can also be displayed as so-called *splitbars*, which will be explained in detail in section 8.

7.2 Conclusion visualization

Parallel coordinates are also appropriate for visualizing the conclusion of a TSK fuzzy rule. The coefficients of the corresponding linear function l can be interpreted as a point $\mathbf{w} = (w_0, w_1, \dots, w_p)$ in a $(p+1)$ -dimensional space and displayed in parallel coordinates. However, since this *Cartesian* representation makes it difficult to infer the hyperplane U defined by \mathbf{w} , it is more appropriate to represent the normal vector $\mathbf{v} = (w_1, w_2, \dots, w_p, -1)$ of the

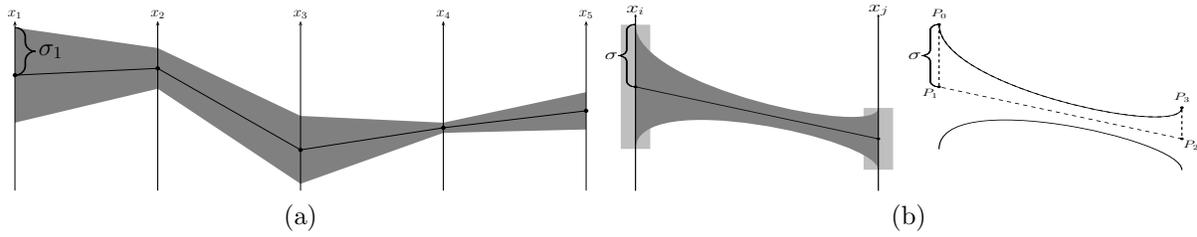


Figure 4: The left picture shows a fuzzy rules antecedence represented as a *linear tube* in parallel coordinates. On the axes, the tube has the width $2\sigma_i$ and is centered over c_i . Tube pieces between two neighbored axes own linear borders. In the Bézier tube representation (right picture), the edges of a tube piece are inside arched Bézier curves. The control points of the upper curve are $P_0 = c_i$, $P_1 = c_i + \sigma_i$, $P_2 = c_j + \sigma_j$ and $P_4 = c_j$, and the control points of the lower curve are $P_0 = c_i$, $P_1 = c_i - \sigma_i$, $P_2 = c_j - \sigma_j$ and $P_4 = c_j$.

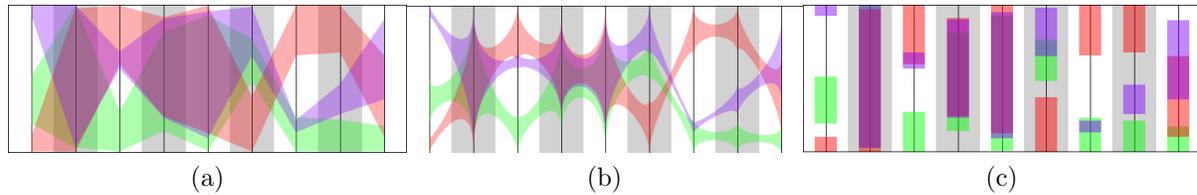


Figure 5: Comparison of tube, Bézier tube and bar representation of fuzzy rule antecedence.

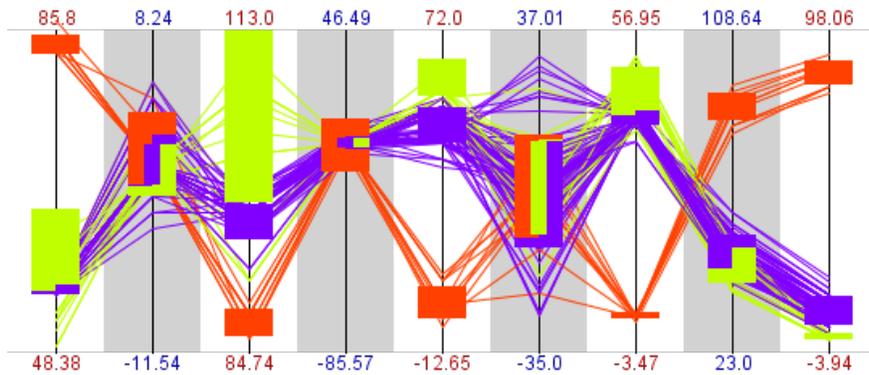


Figure 6: Illustration of three fuzzy rule antecedences displayed as **splitbars**. In addition to the rules, the current data is displayed.

hyperplane U in spherical coordinates $(r, \phi_1, \phi_2, \dots, \phi_p)$:

$$\begin{aligned} r &= \sqrt{w_1^2 + w_2^2 + \dots + w_p^2 + 1} \ , \\ \tan \phi_1 &= \frac{w_1}{w_2} \ , \\ \tan \phi_2 &= \frac{\sqrt{w_1^2 + w_2^2}}{w_3} \ , \\ &\vdots \\ \tan \phi_p &= \frac{\sqrt{w_1^2 + w_2^2 + \dots + w_p^2}}{-1} \ . \end{aligned}$$

This representation has several advantages. The normal vector uniquely defines the orientation of the hyperplane. Moreover, from spherical coordinates, the observer is directly able to see how \mathbf{v} is rotated against the single axis, and hence can easily imagine the orientation of the hyperplane. The angles are bounded by intervals: $\phi_1 \in [-180^\circ, 180^\circ]$ and $\phi_i \in [0^\circ, 180^\circ]$ for $i = 2, 3, \dots, p$. A representation in terms of spherical coordinates also facilitates the comparison of two or more conclusions, since the differences between the angles are in proportion to rotations of the hyperplanes (in contrast to the Euclidean distance in Cartesian space). For the radius r of the spherical coordinates,

$$r = \sqrt{w_1^2 + w_2^2 + \dots + w_p^2 + 1} = \sqrt{D_{\nabla l} + 1} \ , \quad (16)$$

with ∇l the gradient of the conclusion l and $D_{\nabla l}$ the directional derivative in ∇l . Thus, the radius offers information about the greatest increase, which is also provided by ϕ_p . In fact, $\pi - \phi_p$ is the angle between the hyperplane U and the p -dimensional subspace in which the data is embedded, and

$$\tan \phi_p = -\sqrt{D_{\nabla l}} \ . \quad (17)$$

Therefore, instead of displaying r , we consider the y -intercept w_0 . The latter provides information of the hyperplane's location, whereas the angles are telling something about its orientation. Eventually, we obtain $(w_0, \phi_1, \phi_2, \dots, \phi_p)$ as a representation of the conclusion. Since the y -intercept w_0 has a meaning that differs from the angles $(\phi_1, \phi_2, \dots, \phi_p)$, we display w_0 on the first axis of the parallel coordinates, separated from the angles, which are displayed as a polygonal line on the other axes (Figure 7).

As a disadvantage of the conclusion representation $(w_0, \phi_1, \phi_2, \dots, \phi_p)$, note that it completely ignores the position and size of the rule, which is determined by the antecedence (Figure 8). The influence of the conclusion only extends over an area around the center of the antecedence. It is hence interesting to look at the location of the conclusion in the antecedence-centered reference system. A simple quantity containing this information is $l_i(\mathbf{c})$, which is the intercept of the y -axis displaced from the origin into \mathbf{c} . FISVis provides both, the y -intercept w_0 and the centered y -intercept $l_i(\mathbf{c})$, as both quantities might be of interest.

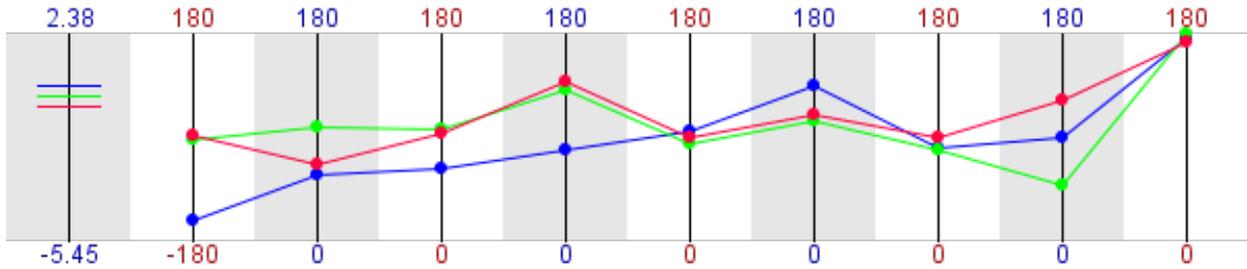


Figure 7: Three conclusions in ‘spherical’ parallel coordinates.

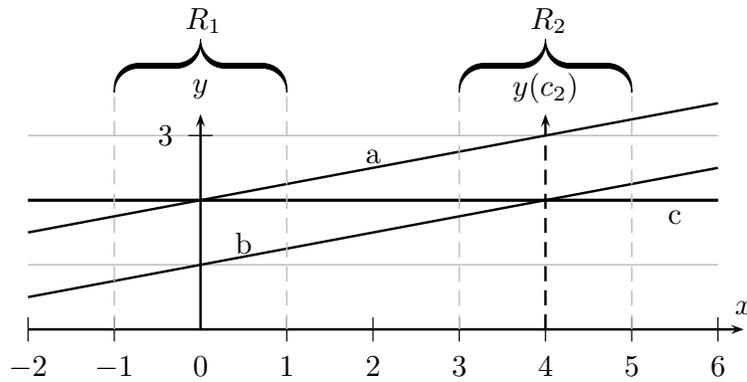


Figure 8: Illustration of two rules with three possible conclusions a , b and c . The y -intercept as well as the angle similarity between a , b and c are independent of the position for R_1 or R_2 , but a and c are more similar from the perspective of R_1 than from R_2 . Otherwise, the conclusions b and c have the same similarity for R_2 as a and c for R_1 , but the y -intercept difference of a and c is 0, whereas the y -intercept difference of b and c is 1. Using the rule centered y -intercept eliminates this effect.

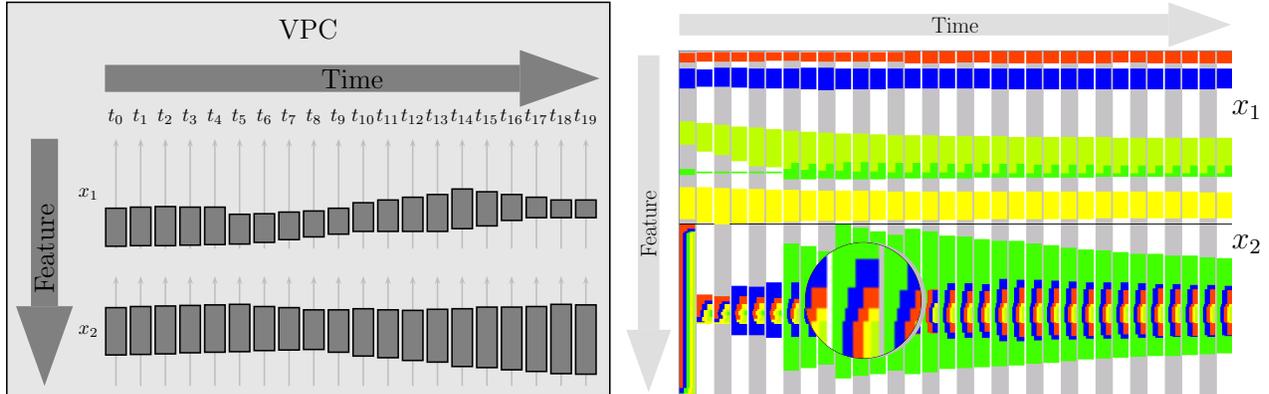


Figure 9: The left picture shows vertical parallel coordinates for two attributes x_1 and x_2 against time. The right picture illustrates a vertical parallel coordinate time line of five two-dimensional rules. All rules exhibit overlap in the second feature x_2 . This causes an overlap of all painted bars representing the rules in the second dimension. For enabling an identification of single bars, the overlapping parts of the bars are displayed in a split bar representation.

8 Vertical parallel coordinates time line

The visualization of fuzzy rules through parallel coordinates provides a local view, both spatially and temporally. However, for visualizing an evolving process, for which time naturally plays an important role, a temporally global view is compulsory. In this section, we therefore introduce the *vertical parallel coordinate time line* as a technique supporting such a view. Vertical parallel coordinates (VPC) are normal parallel coordinates with stacked axes. Instead of using the stop motion technique described in Section 7, we produce one coordinate system for every time point and arrange all VPCs along a horizontal time line (Figure 9, left).

To tackle the problem of overlapping bars, we developed a technique to divide the overlapping parts into n horizontal bars with width b/n (Figure 9, right): b is the size of a non-split bar and n is the number of bars overlapping in the corresponding part. The splitting algorithm consists of two parts. First, a list of separators is built, which indicate the start and end of a split bar. Then, the split bars are painted using the separator list (Figure 10). The pseudocode for the split generation and plotting is given in Algorithms 1 and 2.

8.1 Visualizing merging processes with VPC

Evolving a rule system does not only involve the modification of existing rules and the creation of new rules from scratch, but also the merging of rules. Apart from being useful for providing a temporal global overview of the evolution of single rules \mathcal{R}_S , a VPC time line can also be used as a tool for analyzing the history of merging processes and the merging

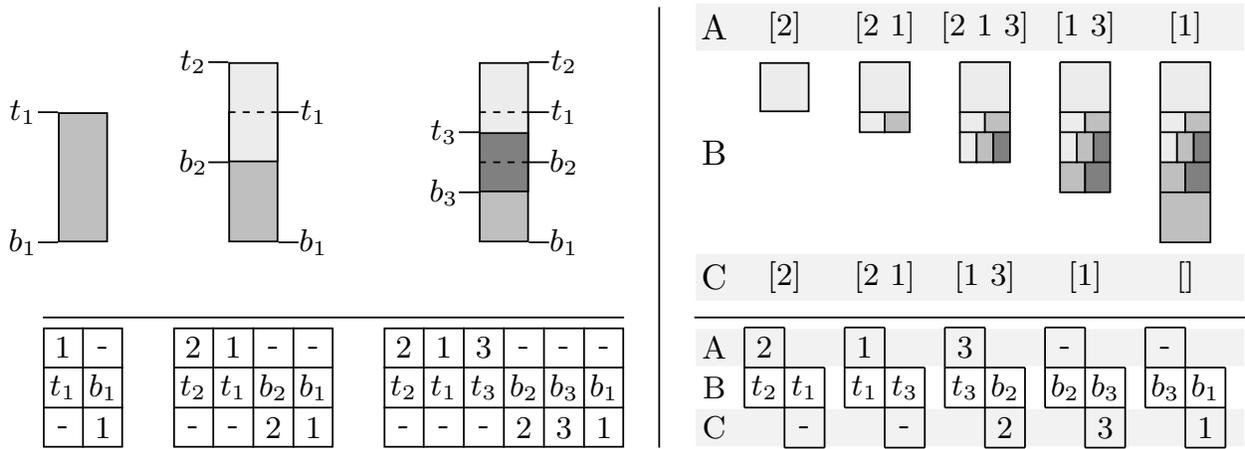


Figure 10: Illustration of the *SplitList* algorithm (left) and *PaintSplit* algorithm (right). The *SplitList* algorithm builds a separator list for three partially overlapping bars. The *PaintSplit* algorithm takes the separator list and draws the split bar.

Algorithm 1 SplitList builds a separator list for partially overlapping bars.

```

1: procedure SPLITLIST( $\mathcal{R}_S$ )
2:   empty list splitList;
3:   for  $R \in \mathcal{R}_S$  do
4:     top  $\leftarrow$  {R.label, null} ▷ A (see Fig. 10)
5:     value  $\leftarrow$  {R.center + R.spread, R.center - R.spread} ▷ B
6:     bottom  $\leftarrow$  {null, R.label} ▷ C
7:     for  $k \leftarrow 0$  to 1 do
8:       separator  $\leftarrow$  {top[k], value[k], bottom[k]}
9:        $j \leftarrow 0$ 
10:      for  $i \leftarrow 0$  to size(splitList) do
11:        if splitList(i)[1] > separator[1] then
12:           $j \leftarrow i$ 
13:          break
14:        end if
15:      end for
16:    end for
17:    add(splitList,separator,j)
18:  end for
19:  return splitList
20: end procedure

```

Algorithm 2 PaintSplit takes the separator list and draws the split bar.

```

1: procedure PAINTSPLIT(splitList)
2:   labelList
3:   for  $i \leftarrow 0$  to  $\text{size}(\text{splitList})-1$  do
4:      $\text{add}(\text{labelList}, \text{splitList}(i)[0])$  ▷ A
5:      $\text{paintSplitBox}(\text{splitList}(i)[1], \text{splitList}(i+1)[1], \text{labelList})$  ▷ B
6:      $\text{remove}(\text{labelList}, \text{splitList}(i+1)[2])$  ▷ C
7:   end for
8: end procedure

```

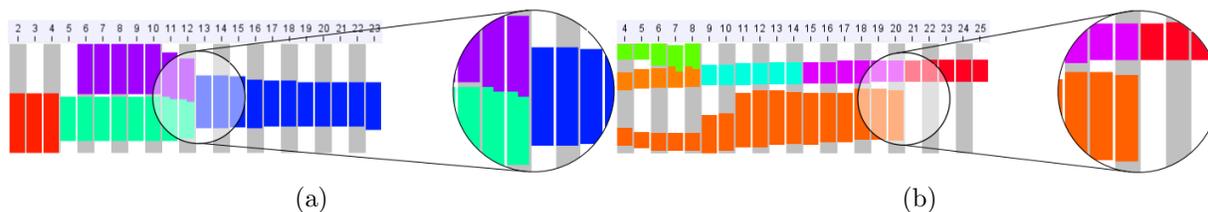


Figure 11: Both pictures show a merging process. (a) Two rules converge and merge. The resulting rule is a weighted average of the two parent rules. (b) The orange and purple rule converge, but the resulting red rule is more or less identical to the purple rule. Thus, the purple rule is much more relevant than the orange one which has hardly any influence on the merging process, resulting in the rule shown in red.

itself.

For example, the FLEXFIS algorithm [14] does not only mix two rules by averaging the centers and widths of the corresponding fuzzy sets, but also captures the relevance of a rule, which is measured in terms of the number of instances covered by that rule [15]. An example of two merging processes is shown in Figure 11. In the first case (Figure 11(a)), both rules seem to have nearly the same relevance, whereas in the second case (Figure 11(b)), the purple rule seems to have a much higher relevance than the orange one, because the resulting red rule is very similar to the purple one.

In FLEXFIS, a merging of two rules is initiated when their antecedences overlap to a certain degree [15]. Although the conclusion is not considered for the initiation of a merging process, it is taken into consideration for selecting the merging strategy. Apart from a “real” merging strategy, where in addition to the rules antecedences also the conclusions are mixed, FLEXFIS provides another strategy for the case of a certain dissimilarity in the rule conclusion [15]. Then, the conclusion of the more relevant rule is taken as the conclusion of the new rule.

Practically, it is not always possible to distinguish between the two types of merging processes, i.e., the real merging and the degenerate case in which the conclusion of the resulting rule is completely determined by a single dominating rule. In this regard, a combination

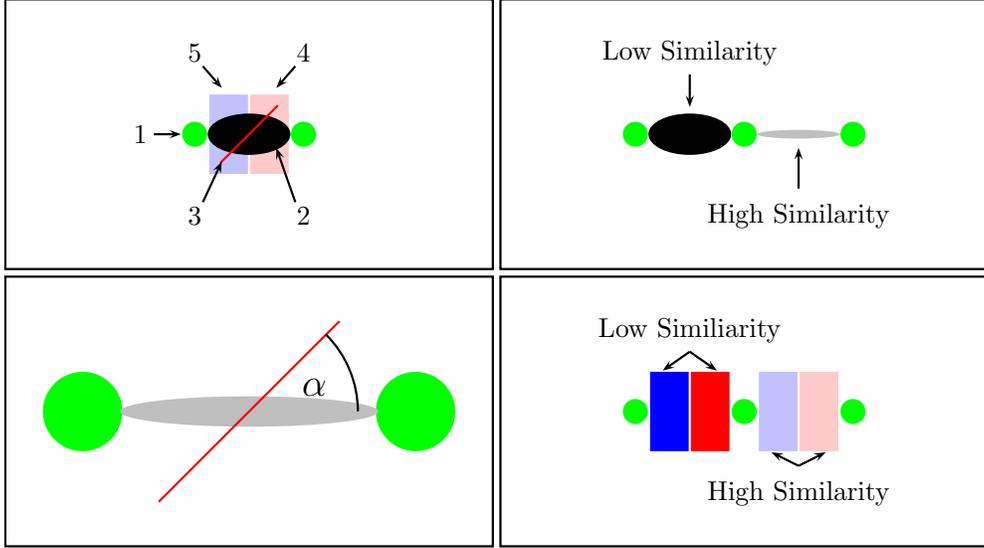


Figure 12: The upper left panel shows all five elements of a rule chain. The elements pointed at by the numbered arrows are: (1) rule pearl, (2) antecedence similarity link, (3) angle line, (4) angle similarity block, (5) conclusion shift block. The top right panel describes the meaning of different heights and color intensities of ellipsoid links. The bottom right panel describes the meaning of different color intensities of rectangular links. Generally, both types of links can be used for visualizing any given similarity or distance measure. The lower left panel shows the rotated line segment for visualizing the angle between the conclusions of two adjacent rules.

of the VPC and the PC visualization techniques is useful. If a merging is observed in the VPC visualization, the analysis of the rule conclusions in the PC visualization can provide important additional information.

9 Rule chains

In this section, we describe rule chains which have already been introduced in [9] as an adequate way for visualizing specific aspects of the evolution of a fuzzy rule-based system. Essentially, a rule chain seeks to capture the changes of a single rule between two consecutive time points. A rule history \mathfrak{H} is visualized as a horizontal pearl chain, where every pearl represents a rule at a certain time point. The first time point is located at the left and the most recent time point at the right end. Pearls along a chain are connected with different types of links. Focusing on a measure of interest, links between adjacent pearls can represent the corresponding similarity or distance between consecutive rules. As illustrated in Figure 12, there are three types of links that correspond to the measures introduced in Section 4, respectively.

In addition to the horizontal time point position, the vertical position of a pearl determines

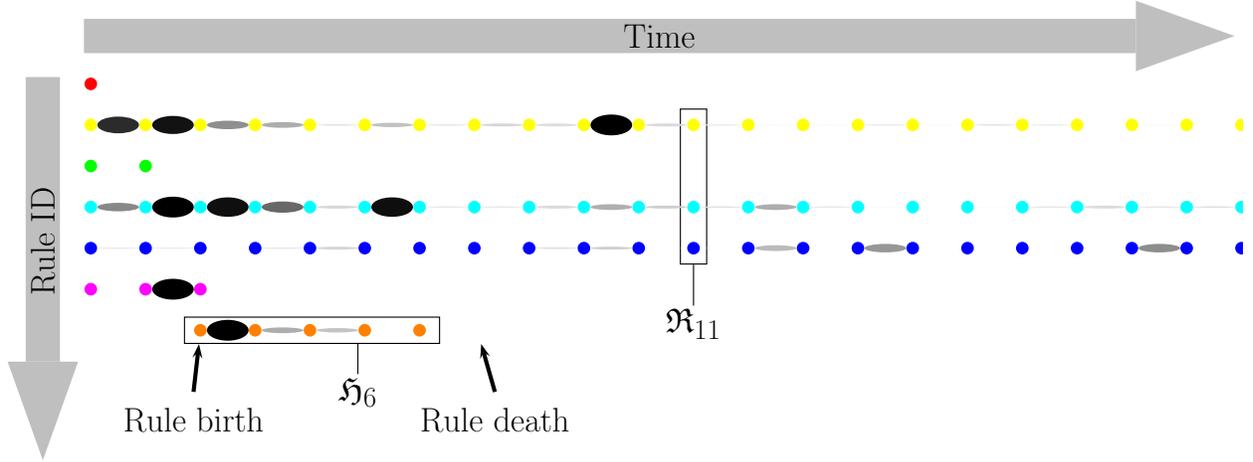


Figure 13: This figure shows a cut-out from a rule chain system. All rules, symbolized by pearls, belonging to the same rule system are aligned vertically. Rules belonging to the same rule history are organized in horizontal chains. The pearl links are associated with similarity measures. Here, only the antecedence similarity links are shown.

the history of the associated rule. The antecedence similarity link corresponds to the antecedence similarity measure or the Euclidean distance of centers. The degree of similarity is connected to the color intensity and shape of the ellipse. The lower the similarity, the wider and darker is the ellipse. The angle similarity block visualizes the angle similarity, and the conclusion shift block displays differences of y -intercepts of the related rule conclusions. The color saturation of blocks is the stronger, the lower the similarity of linked rules is. Thus, the three types of links highlight different aspects of change between two rules. The angle line is mathematically positively rotated by α against the horizon to display the angle between two temporally adjacent conclusions.

For visualizing the evolving system as a whole, the horizontal rule chains are stacked (Figure 13). A single rule chain gives a quick overview of a rule's lifetime and development, and stacking them provides a holistic view on the development of the whole rule chain system.

9.1 Concept drift detection with rule chains

An important requirement for evolving (fuzzy) systems is a quick discovery and adequate reaction to so-called *concept drift* [8]. Roughly speaking, a concept drift is a (gradual) change of the data-generating process in the course of time, that is, a change of the probability distribution that generates input/output tuples emitted by the data stream.

Rule chains provide an adequate (visual) means for discovering concept drift and monitoring the fuzzy system's reaction to this drift. In fact, as a reaction to a change of the data generating process, the fuzzy system is expected to adapt a possibly large number of rules (both, antecedence and/or conclusion parts). Likewise, new rules will typically be created

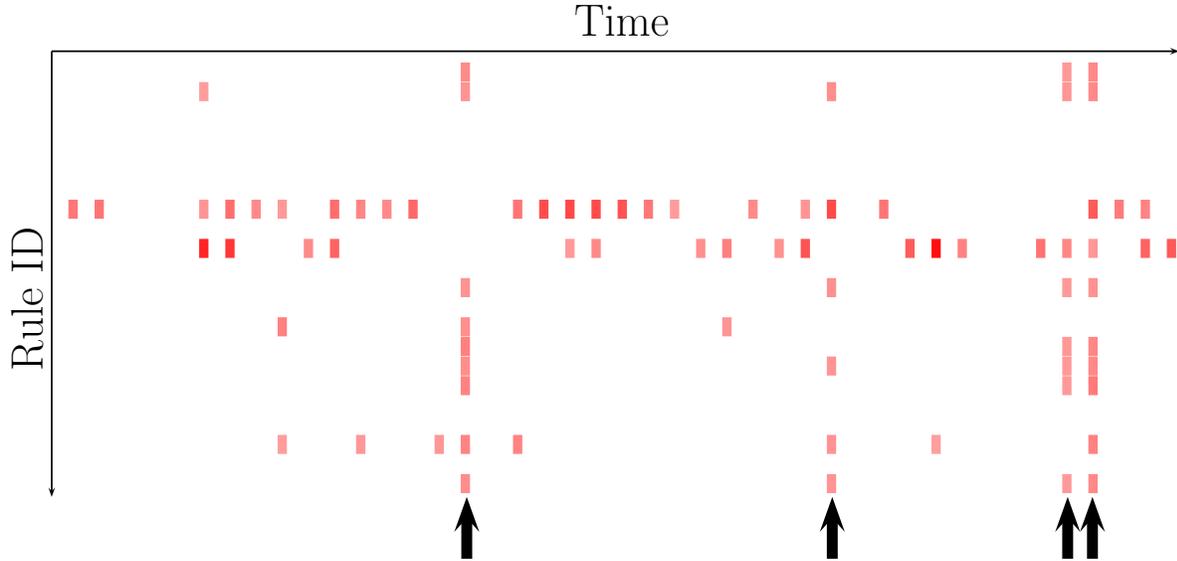


Figure 14: This figure highlights time-specific vertical patterns across various rules in the rule chains which indicate significant changes in the model and, therefore, suggest a possible concept drift in the data. Redundant information of contiguous blocks with high temporal similarity is suppressed by color thresholding. The bold arrows are marking time points at which an abrupt change of the conclusion parts of many rules can be observed.

and existing ones will be deleted.

Changes of that kind naturally produce observable patterns in the visualized rule chain system. More precisely, a simultaneous change of the antecedence or conclusion parts of many rules will produce noticeable vertical lines in the rule chain system (Figure 14). Such patterns can be amplified by defining thresholds for the minimal change to be drawn. The simultaneous appearance or disappearance of many rules produces long vertical edges in the displayed rule chains. This enables the user to recognize potential concept drifts in the data. More correctly, since we are visualizing the model and not the data evolution, the observation of such patterns should only be taken as an indication (and not as a proof) of a possible concept drift.

10 Experiments

This section is meant to illustrate the three visualization techniques introduced above, using different types of synthetic data. In contrast to real data, synthetic data allows for conducting controlled experiments and, since the “ground truth” is known, for judging the plausibility of the results. Moreover, while our visualization tool is in principle independent of the learning algorithm, we again used FLEXFIS [14] as a concrete instantiation; we set the built-in forgetting factor to 0.9 for the first and third experiment and to 0.99 for the second

experiment. This factor controls the forgetting of the inverse Hessian matrix during the recursive weighted least squares optimization of the rule consequences (the smaller its value, the more quickly the influence of previous examples decays).

10.1 The data generation process

To generate a data stream, we concatenate several blocks of synthetic data sets. Every data set is associated with a time point that specifies its position q in the concatenation. A single data set is generated by specifying the position q and a set of clusters $\{C_1, C_2, \dots, C_m\}$. The latter has the same size for all data sets (blocks) belonging to the same data stream. A cluster C_i is specified by four parameters: (i) the number of instances N_i to be created for this cluster; (ii) the center $\mathbf{c}_i = (c_{i,1}, c_{i,2}, \dots, c_{i,n})$ and (iii) the width $\boldsymbol{\sigma}_i = (\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,n})$ of a normal distribution, based on which the instances $\mathbf{x} = (x_1, x_2, \dots, x_n)$ belonging to the cluster are generated; (iv) a coefficient vector $(w_{i,0}, w_{i,1}, \dots, w_{i,n})$ used to calculate the output value of the instances through

$$y = f_i(\mathbf{x}) = w_{i,0} + w_{i,1}(x_1 - c_{i,1}) + \dots + w_{i,n}(x_n - c_{i,n}) , \quad (18)$$

where i is the index of the i^{th} cluster C_i . Thus, a data set can be characterized by $[q, (\mathbf{c}_i, \boldsymbol{\sigma}_i, N_i, \mathbf{w}_i)_{i=1}^m]$, where m is the number of clusters.

For the whole stream, only a few data sets lying on so-called *anchor positions* are specified explicitly. The data sets lying between two anchor positions are interpolated by a linear function (i.e., the parameters $\mathbf{c}, \boldsymbol{\sigma}, n, \mathbf{w}$ characterizing a data set are convex combinations of the corresponding parameters of the left and right anchor data sets). In general the data sets lying on anchor positions differ in their characterization, which in fact means they are produced by different concepts. Hence, there are temporal concept drifts within the data stream from one anchor position to the next anchor position. The position points q have to be distinguished from the “time” parameter t underlying the model generation process. The amount of data points used to learn a model depends on the settings of the learning algorithm (in our case FLEXFIS). Normally, q and t have different frequencies, and the start is shifted because of the initialization of the model. Furthermore, the position points q are in general not equidistant (measured in data points), in contrast to the model generation process t (Figure 15).

10.2 Visualizing conclusion evolution

In this first experiment, we focus on the visualization of rule conclusions with parallel coordinates and compare the “Cartesian” representation with the “spherical” one. The synthetic data stream used in this experiment is shown in Table 1. The stream consists of two clusters having the same size and width but different locations. Only the coefficients defining the output are changing over time. On this data stream, FLEXFIS produces two rules and adapts their conclusions over time.

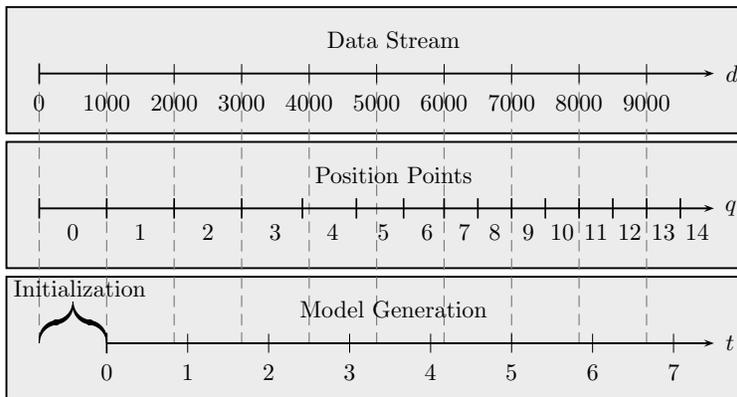


Figure 15: Illustration of different system internal time scales. All three axes are equally scaled in order to refer to the same number of data points (dashed vertical lines). The tick distance of the t -axis is 1200 data points and the tick distance of the q -axis varies between 500 and 1000 data points. This constellation is the result of a data stream defined by three anchor points: $[0, (\mathbf{c}_1, \boldsymbol{\sigma}_1, 500, \mathbf{w}_1), (\mathbf{c}_2, \boldsymbol{\sigma}_2, 500, \mathbf{w}_2)]$, $[2, (\mathbf{c}_1, \boldsymbol{\sigma}_1, 500, \mathbf{w}_1), (\mathbf{c}_2, \boldsymbol{\sigma}_2, 500, \mathbf{w}_2)]$, $[7, (\mathbf{c}_1, \boldsymbol{\sigma}_1, 350, \mathbf{w}_1), (\mathbf{c}_2, \boldsymbol{\sigma}_2, 150, \mathbf{w}_2)]$ and an algorithm that uses 1000 data points for initialization and 1200 data points for an incremental update.

Table 1: Specification of the data stream of the first experiment. For both clusters at any time: $N = 500$, $\sigma_1 = 6$, $\sigma_i = 1$ with $i \in \{2, 3, 4\}$. The center of C_1 at any time is $c = (20, 20, 20, 50)$ and the center of C_2 is $c = (80, 80, 80, 50)$.

| | C_1 | | | | | C_2 | | | | |
|-----|--------------|------|------|------|------|--------------|-------|-------|-------|-------|
| q | coefficients | | | | | coefficients | | | | |
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 15 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | -1.0 | 0.0 | 0.0 |
| 30 | 0.0 | 0.0 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | -10.0 | 0.0 | 0.0 |
| 45 | 0.0 | 10.0 | 10.0 | 10.0 | 0.0 | 0.0 | -10.0 | -10.0 | -10.0 | 0.0 |
| 60 | 0.0 | 10.0 | 10.0 | 10.0 | 10.0 | 0.0 | -10.0 | -10.0 | -10.0 | -10.0 |

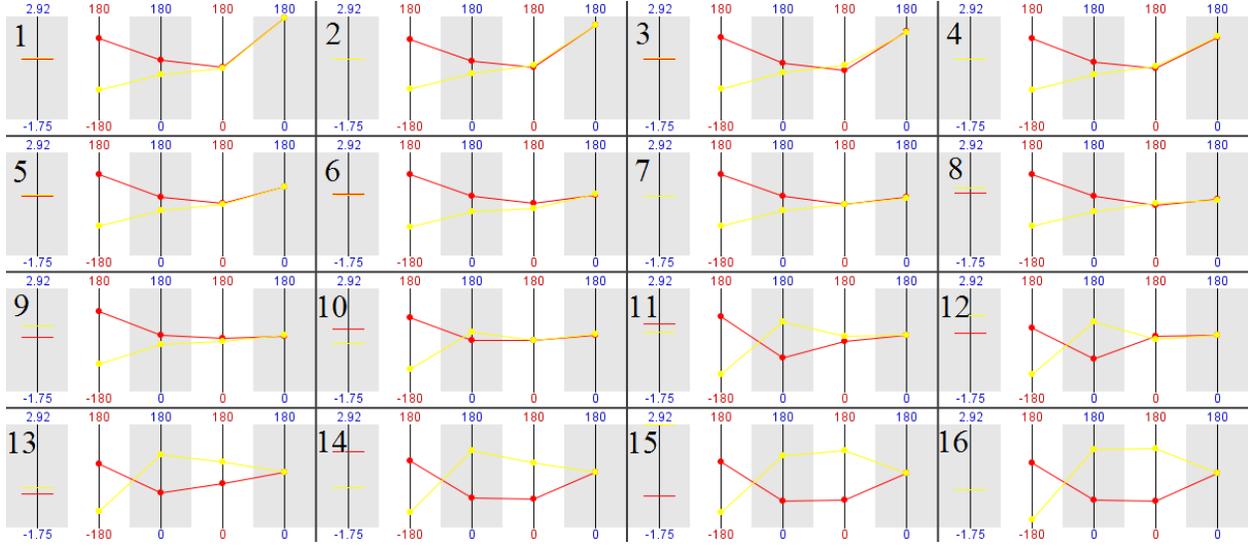


Figure 16: Display of two conclusions in *spherical* parallel coordinates at every 6th generation from 1 to 91.

Figure 16 shows the rule conclusions of every 6th generation from 1 to 91 in “spherical” parallel coordinates ordered from left to right and top to bottom. At the beginning ($q = 0$), the coefficients of both clusters are equal. The first picture in Figure 16 shows the two conclusions of generation 1. At this time, according to definition point 1, $w_{1,2}$ is between 0 and 1 and $w_{2,2}$ between -1 and 0, with $w_{i,j}$ the j^{th} coefficient of the i^{th} cluster, resulting in angles $\psi_{1,1} = 90^\circ$ and $\psi_{2,1} = -90^\circ$. In order to refer to the angle α between the two hyperplanes, the last angle ψ_4 has to be considered. If $\psi_{1,4} \approx \psi_{2,4} \approx 180^\circ$, then $\alpha \approx 0$ is independent of the other ψ . From generation 1 to 21 (pictures 1 to 4), the angles $\psi_{1,4}$ and $\psi_{2,4}$ change to 135° in correspondence to the data concept, where the coefficients $w_{1,2}$ and $w_{2,2}$ change to 1 and -1 respectively. At this point, the hyperplanes over both clusters are orthogonal and have an angle of 45° to the data subspace. From generation 22 to 44 (pictures 5 to 8), the angle ψ_4 of both rules changes to nearly 90° . The coefficients $w_{1,2}$ and $w_{2,2}$ change to 10 and -10 , respectively, which means that the angle between the hyperplanes and data subspace become nearly 90° , and the angle between both hyperplanes tend to zero. As of generation 45, the coefficients w_1, w_3, w_4 also change and cause an adaptation of the rule conclusions. The angles ψ_2 and ψ_3 of both conclusions change, whereas ψ_4 stays at about 90° . This corresponds to the observation of a small angle (less than 45°) between the two conclusions for generations after $t = 44$. This can be seen by using the visualization concept of angle lines introduced in figure 12 as a matrix display (Figure 18).

The “Cartesian” parallel coordinate representation of the conclusion is shown in Figure 17, where the coefficients of the rule conclusions are plotted. While the two conclusions appear to be quite similar in the first four pictures, the similarity of the hyperplanes is high only in the first picture ($\alpha \approx 0^\circ$) but close to 0 in the fourth picture ($\alpha \approx 90^\circ$). In this case, the impression of similarity is only caused by normalization. In fact, the conclusions in the

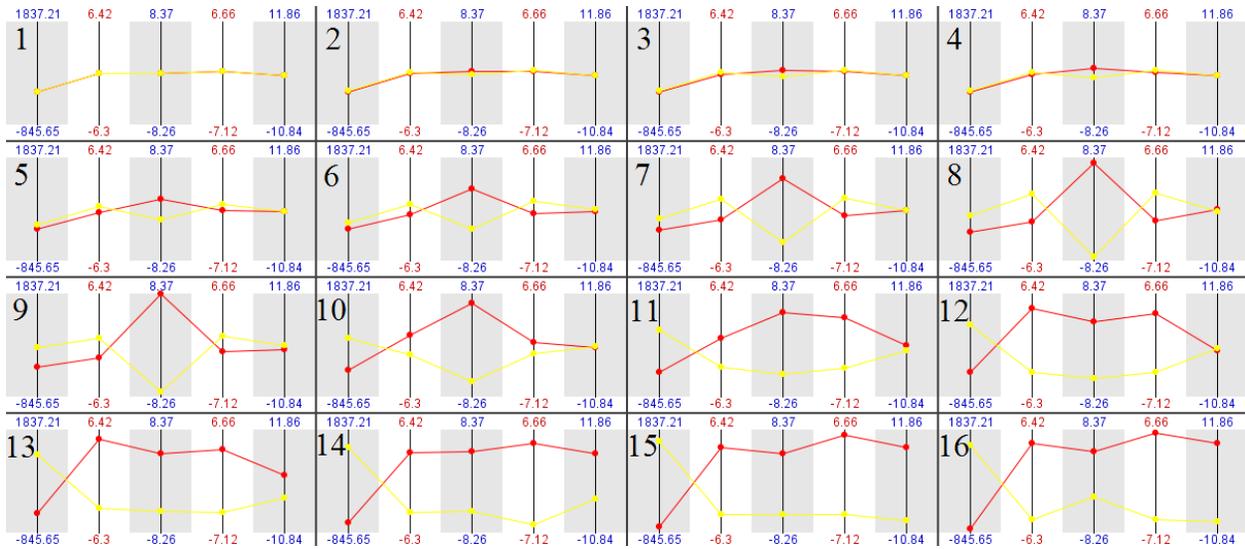


Figure 17: Display of two conclusions in *Cartesian* parallel coordinates at every 6^{th} generation from 1 to 91.

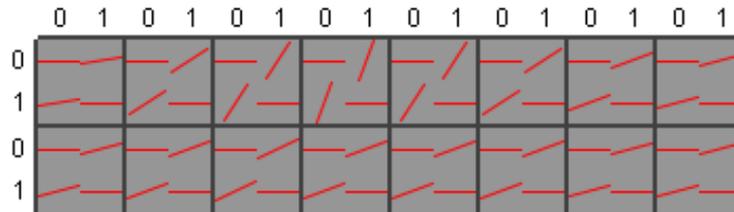


Figure 18: The red lines illustrate angles between the conclusions of rules 0 and 1. Each 2×2 matrix displays one generation. From left to right every 6th generation from 1 to 91 is shown.

Table 2: Specification of the data stream of the second experiment. For both clusters at any time: $N = 500$, $\sigma_1 = 6$, $\sigma_2 = 4.2$, $w_0 = 0$, $w_1 = 1$, $w_2 = 0$.

| | C_1 | | C_2 | |
|-----|-------|-------|-------|-------|
| q | c_1 | c_2 | c_1 | c_2 |
| 0 | 65 | 75 | 35 | 25 |
| 10 | 25 | 75 | 35 | 75 |
| 20 | 65 | 65 | 35 | 35 |
| 30 | 25 | 65 | 75 | 35 |
| 40 | 25 | 65 | 75 | 35 |
| 50 | 50 | 50 | 50 | 50 |

fourth picture would look dissimilar if the axis had a range from zero to one. Anyway, the axis should be normalized in all pictures (at all time points) in the same way, so taking the maximum and minimum values for normalization is clearly advisable. In the last picture, the two conclusions have the biggest Euclidean distance, however, the similarity of the hyperplanes is close to one.

For the purpose of comparing two hyperplanes, the spherical representation is preferable. The angle between hyperplane and data subspace can be efficiently characterized by ψ_p , and in conjunction with the other angles ψ , the angle α between two hyperplanes can be estimated. Using the Cartesian representation, an appraisal of the relative position of two hyperplanes becomes more difficult and less intuitive. The advantage of the Cartesian representation is the direct view on the coefficients.

10.3 Visualizing merging processes

The second data stream (Table 2) demonstrates the benefits of VPC time lines with regard to merging processes in the evolution of a fuzzy rule system. We produced a two-dimensional data stream consisting of two clusters of the same size and shape and the same output coefficients. In the beginning, both clusters are well separated, and FLEXFIS creates one rule for each of them. To enforce a rule merging process, we translate the two clusters in the two-dimensional space using a heuristic strategy, where the clusters change their direction four times (change of the drift concept).

In the VPC display in Figure 19 the evolution of the rules and four merging processes marked with letters A, B, C, D can be clearly observed. In the beginning, we have two rules (yellow, green) over C_1 and two rules (red, light blue) over C_2 . Between mark 0 and 1, C_1 and C_2 move up and down, respectively, along the second dimension. The rules expand in this dimension and converge. At $t = 6$ (mark B), the green and yellow rule are merged into the purple rule, which is merged at $t = 13$ (mark C) with the newly emerged dotted colored rule to a blue green rule. This blue green rule on the other hand merges with a newly emerged red rule into the orange rule ($t = 39$, mark D). At $t = 4$ (mark A), the red and light blue

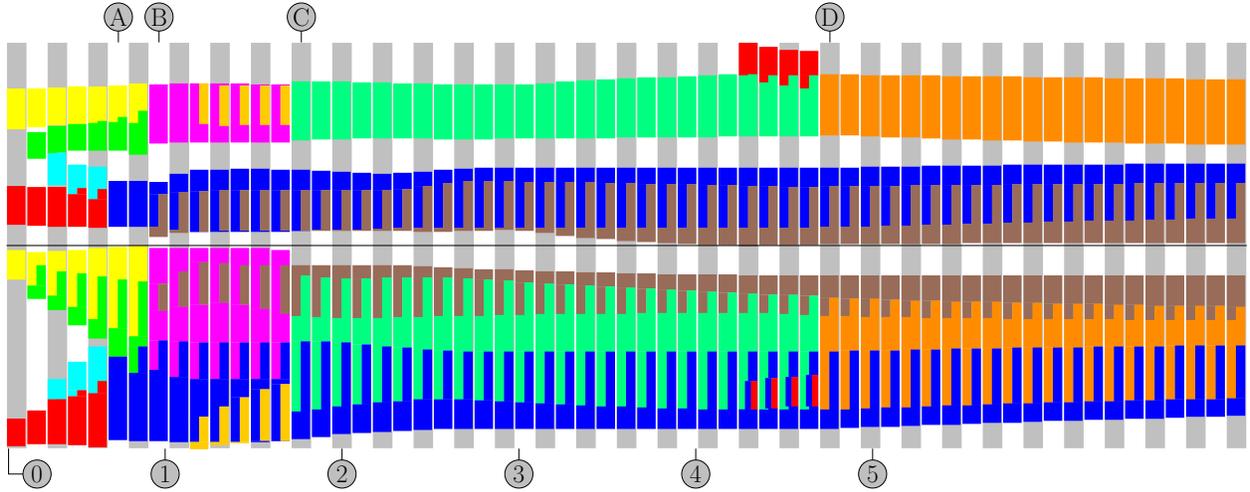


Figure 19: Visualization of the evolution of 11 rules with a vertical parallel coordinates time line. The letters A, B, C, D mark the time points where a merging process can be observed. The numbers mark the time points where the concept drift change.

rules are merged into the blue rule.

The monitoring display in Figure 20 shows that the four merging processes cause distinct and visible reactions in the BM-angle similarity $BM_{\mathcal{E}_\alpha}$ and BM-antecedence similarity $BM_{\mathcal{E}}$. At all four marks (A, B, C, D) are peaks in the course of $BM_{\mathcal{E}_\alpha}$ and $BM_{\mathcal{E}}$, which indicates a change from $R^{(t)}$ to $R^{(t+1)}$ caused through the merging of two rules. The changes of the model structure are confirmed by the plots of the number of rules, number of new rules and number of disappeared rules. At marks A, C and D, the merging of rules causes a reduction of the number of rules. Moreover, a peak in the number of new rules and a peak with double height in the number of disappeared rules can be observed. At mark B, the number of rules remains constant and the peaks in the number of new and disappeared rules have the same size. In the VPC visualization it can be seen that, next to the merging of the yellow and green rule, and new (brown) rule emerges at mark B. This explains the graph courses at mark B (Figure 20).

10.4 Visualizing reactions to concept drift

The data stream used in this experiment contains three concept drifts with two stable episodes in-between. These episodes allow the learner to re-stabilize the model. The characteristics of the data stream are summarized in Table 3. At the beginning, all three clusters are well separated and located next to each other along the first dimension. Moreover, the three regression functions share the same coefficients. Somewhat surprisingly, this initial configuration lets FLEXFIS generate more than only the expected three rules and leads to rich patterns during model formation. It seems that this initial configuration is a difficult learning problem for FLEXFIS, which explains the changes before mark 1.

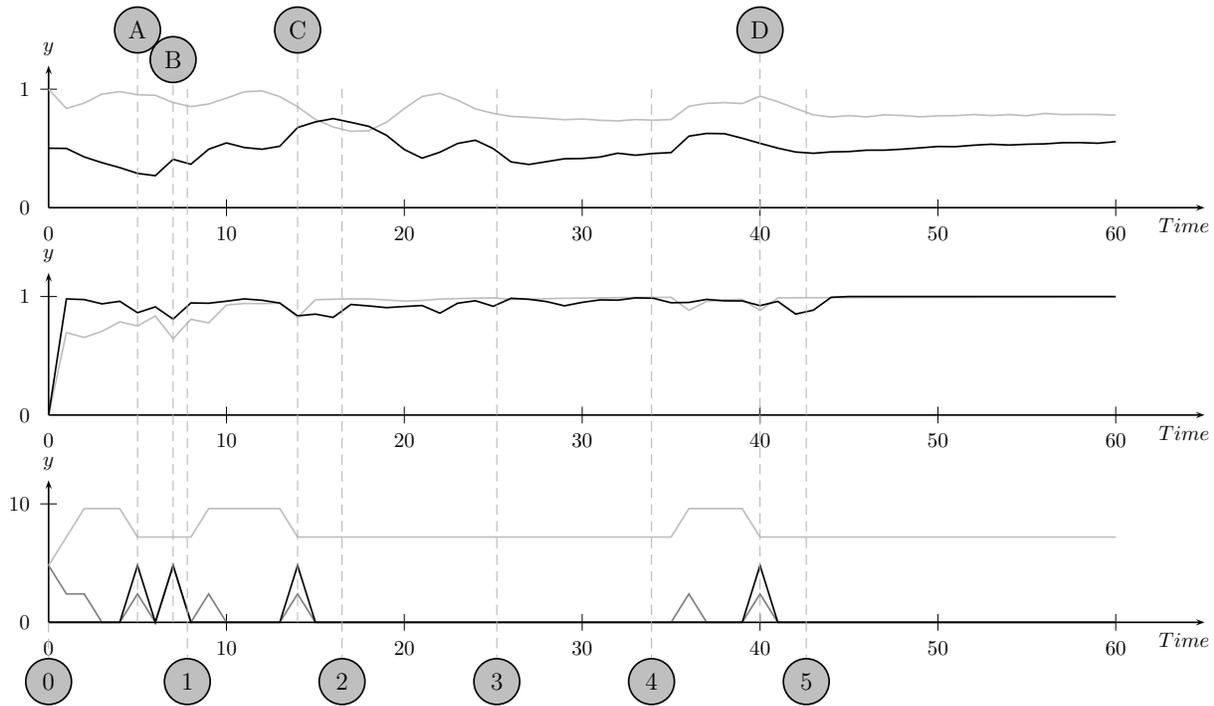


Figure 20: Monitoring of the second experiment. The upper plot shows the graphs of the rule coverage (light gray) and the data coverage (black). The middle plot contains the graphs of the BM-antecedence similarity $BM_{\mathfrak{S}}$ (light gray) and the BM-angle similarity $BM_{\mathfrak{S},\alpha}$ (black). The lower plot shows the number of rules (light gray), number of new rules (black), and number of disappeared rules (gray).

Table 3: Specification of the data stream of the third experiment. Cluster size of each cluster at any time is $n = 1000$. The cluster width is $\sigma_1 = 2$ and $\sigma_2 = 2$ for all clusters.

| | C_1 | | C_2 | | C_3 | | C_1, C_3 | | | C_2 | | |
|-----|-------|-------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|
| q | c_1 | c_2 | c_1 | c_2 | c_1 | c_2 | w_0 | w_1 | w_2 | w_0 | w_1 | w_2 |
| 0 | 10 | 50 | 50 | 50 | 90 | 50 | 0 | 10 | 0 | 0 | 10 | 0 |
| 3 | 10 | 50 | 50 | 50 | 90 | 50 | 0 | 10 | 0 | 0 | 10 | 0 |
| 6 | 10 | 50 | 50 | 50 | 90 | 50 | -10 | 0 | 0 | 10 | 0 | 0 |
| 12 | 10 | 50 | 50 | 50 | 90 | 50 | -10 | 0 | 0 | 10 | 0 | 0 |
| 16 | 10 | -100 | 50 | 150 | 90 | -100 | -10 | 0 | 0 | 10 | 0 | 0 |
| 26 | 10 | -100 | 50 | 150 | 90 | -100 | -10 | 0 | 0 | 10 | 0 | 0 |
| 30 | 10 | -100 | 50 | 150 | 90 | -100 | -10 | 0 | 10 | 10 | 0 | 10 |

The first drift is caused by a change of the regression values (between anchor positions 3 and 6 in Table 3). In Figure 21, this drift phase lies between marks 1 and 2. At mark A, a reaction of FLEXFIS is clearly seen. In the time before mark A, the rule conclusions strongly change from one system \mathfrak{R}_t to the following \mathfrak{R}_{t+1} . After mark A, only small changes in the conclusions can be observed. This behavior is also mirrored in the BM-angle similarity plot of the monitoring system in figure 22. The graph of $BM_{\mathfrak{S}_\alpha}$ jumps up at mark 2 and keeps this level till the next concept drift between marks B and C.

The second drift (between anchor positions 12 and 16 in Table 3) is caused by a movement of the clusters. As seen in Table 3 cluster C_1 and C_3 are sequentially moving down in the direction of the second dimension, whereas cluster C_2 is moving up. The period of this drift is delimited by marks 3 and 4 (Figure 21). The visible reaction of FLEXFIS is between marks B and C. Here, a change in the antecedent parts of the rules can be observed, which confirms the expectation from the experimental design. Also the course of the rule and data coverage change substantially during the second drift phase between mark B and C (Figure 22). To explain this change, observe that the clusters are moving away under the rules, which only follow with a delay and to a certain degree. This causes a reallocation of the instances to the rules (change of rule coverage) and weaker coverage of the instances through the rules (change of data coverage).

The last concept drift (between anchor positions 26 and 30 in Table 3) is again caused by a drift in the regression values. It starts at mark 5 and ends at mark 6. A change in the rule system caused by this drift can be seen from mark D onwards. After the third drift, the rule system does not reach a stable stage until the end of the experiment. Again, the rule chains nicely reflect the changes in the course of $BM_{\mathfrak{S}_\alpha}$ after mark D (Figure 22). All three concept drifts cause an adaptation of the rule system by the learning algorithm, resulting in clearly visible patterns in the rule chain visualization.

11 Conclusion

In this work, we introduced the visualization tool FISVis, which, apart from a basic monitoring system, provides several interactive visualization techniques for exploring evolving TSK fuzzy models: dynamic parallel coordinates, vertical parallel coordinate time lines, and rule chains. The potential and benefits of these techniques have been shown in three experiments, each focusing on specific aspects of an evolving process.

In the first experiment, we showed that a representation in “spherical” parallel coordinates of a rule conclusion is well interpretable. Furthermore, an estimation of the relative position of two rule conclusions is improved compared with a “Cartesian” representation of the conclusion coefficients. The second experiment was focused on visualizing merging processes through vertical parallel coordinate time lines. It was shown that VPC time lines provide important insight in the evolution of rules and the interaction between rules, and helps to understand rule merging processes. In the last experiment, we tried to highlight some benefits of rule chains. We discovered that vertical patterns clearly indicate possible concept

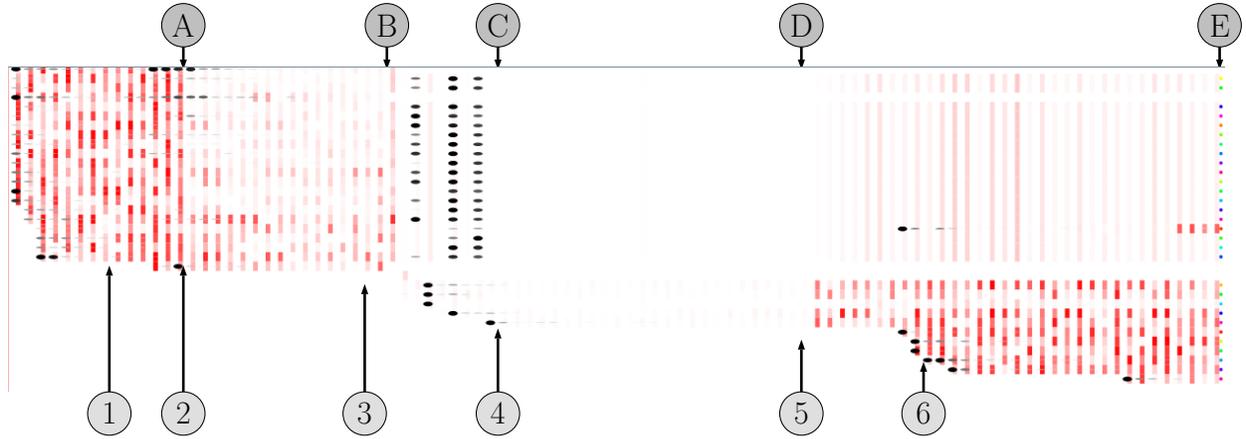


Figure 21: Visualization of an evolving fuzzy rule-based system using stacked rule chains introduced in figure 12. There are three concept drifts in the underlying data stream, the first between 1 and 2, the second between 3 and 4, and the third between 5 and 6. The letters mark the time points where a change in the rule system, as a reaction to the concept drift, becomes visible.

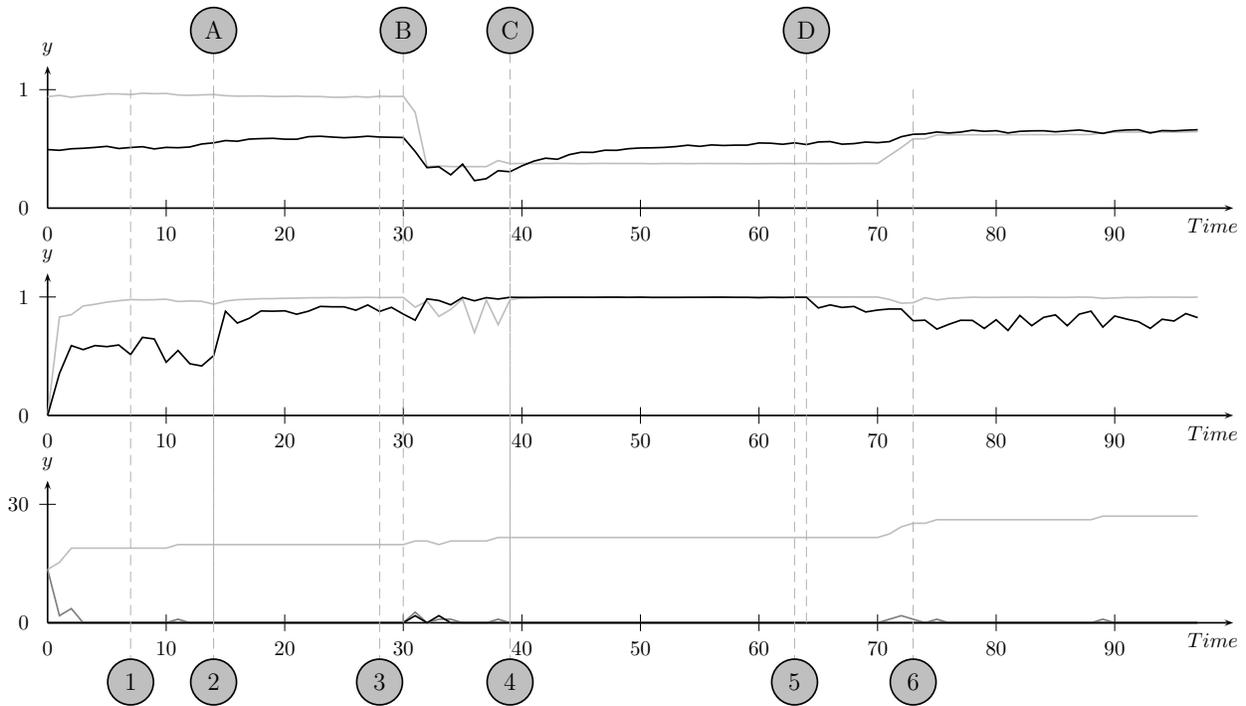


Figure 22: Monitoring of the third experiment. The upper plot shows the graphs of the rule coverage (light gray) and the data coverage (black). The middle plot contains the graphs of the BM-antecedence similarity $BM_{\mathfrak{S}}$ (light gray) and the BM-angle similarity $BM_{\mathfrak{S},\alpha}$ (black). The lower plot shows the number of rules (light gray), number of new rules (black) and number of disappeared rules (gray).

drifts and corresponding reactions of the evolving model. Moreover, rule chains provide a quick overview of the size of a rule system (number of rules) at a certain time point as well as the size of the whole evolving system; thus, the temporal local and global complexity of the system can be grasped quickly.

The different techniques are complementary in the sense of visualizing different aspects of model evolution, and the combination of these techniques clearly makes FISVis a powerful tool. This tool should be of interest for both end-users who receive early feedback about parameter settings and developers who like to study convergence or adaptation properties of their models. It would be interesting to assess the practical values of the proposed visualization concepts by a conducting real-world user studies.

The FISVis software can be downloaded at www.uni-marburg.de/fb12/kebi/research.

Acknowledgments

This project is supported by the German Research Foundation (DFG). The authors also acknowledge financial support by the LOEWE Center for Synthetic Microbiology (SYNMIKRO), Marburg. Moreover, they like to thank Edwin Lughofer for providing the FLEX-FIS system.

References

- [1] J.M. Alonso, O. Cordon, A. Quirin, and L. Magdalena. Analyzing interpretability of fuzzy rule-based systems by means of fuzzy inference-grams. In *Proc. World Conference on Soft Computing*, pages 181.1–181.8, San Francisco, 2011.
- [2] P. Angelov. *Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems*. Springer-Verlag, New York, 2002.
- [3] P. Angelov. An approach to on-line identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B—Cybernetics*, 34(1):484–498, 2004.
- [4] P. Angelov, D. Filev, and N. Kasabov, editors. *Evolving Intelligent Systems: Methodology and Applications*. John Wiley and Sons, New York, 2010.
- [5] M.R. Berthold and L.O. Hall. Visualizing fuzzy points in parallel coordinates. *IEEE Transactions on Fuzzy Systems*, 11(3):369–374, 2003.
- [6] R.E. Burkard, M. Dell’Amico, and S. Martello. *Assignment Problems*. SIAM, 2009.

- [7] T.R. Gabriel, K. Thiel, and M.R. Berthold. Rule visualization based on multi-dimensional scaling. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 66–71, Vancouver, BC, Canada, 2006.
- [8] J. Gama. A survey on learning from data streams: current and future trends. *Progress in Artificial Intelligence*, 1(1):45–55, 2012.
- [9] S. Henzgen, M. Strickert, and E. Hüllermeier. Rule chains for visualizing evolving fuzzy rule-based systems. In *Proc. CORES-13, 8th International Conference on Computer Recognition Systems*, pages 279–288, Wrocław, Poland, 2013.
- [10] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Adaptive and Learning Systems for Signal Processing, Communications and Control Series. Wiley, 2001.
- [11] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.
- [12] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2002.
- [13] D.A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann. *Mastering The Information Age—Solving Problems with Visual Analytics*. Eurographics, 2010.
- [14] E. Lughofer. FLEXFIS: A robust incremental learning approach for evolving Takagi-Sugeno fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(6):1393–1410, 2008.
- [15] E. Lughofer, J.L. Bouchot, and A. Shaker. On-line elimination of local redundancies in evolving fuzzy systems. *Evolving Systems*, 2(3):165–187, 2011.
- [16] E. Lughofer and E. Hüllermeier. On-line redundancy deletion in evolving fuzzy regression models using a fuzzy inclusion measure. In S. Galichet, J. Montero, and G. Mauris, editors, *Proceedings Eusflat-2011, 7th International Conference of the European Society for Fuzzy Logic and Technology*, pages 380–387, Aix-les-Bains, France, 2011.
- [17] G. Peters, K. Bunte, M. Strickert, M. Biehl, and T. Villmann. Visualization of processes in self-learning systems. In *Proc. TSOS-12, Tenth Annual International Conference on Privacy, Security and Trust*, pages 244–249, Los Alamitos, CA, USA, 2012. IEEE Computer Society.
- [18] F. Rehm, F. Klawonn, and R. Kruse. Rule classification visualization of high-dimensional data. In *Proc. IPMU-06, 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Paris, 2006.
- [19] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

- [20] T. D. Wilson. Review of: Steele, julie & ilinsky, noah. (eds.) beautiful visualization. looking at data through the eyes of experts.. sebastopol, ca: 2010. *Inf. Res.*, 15(3), 2010.
- [21] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. In *Computer Graphics Forum*, volume 27, pages 1047–1054. Wiley Online Library, 2008.