Online Recognition of Fuzzy Time Series Patterns

Gernot Herbst, Steffen F. Bocklisch

Department of Electrical Engineering, Chemnitz University of Technology
D-09107 Chemnitz, Germany
Email: {gernot.herbst, steffen.bocklisch}@etit.tu-chemnitz.de

Abstract— *This article deals with the recognition of recurring multivariate time series patterns modelled sample-point-wise by parametric fuzzy sets. An efficient classification-based approach for the online recognition of incompleted developing patterns in streaming time series is being presented. Furthermore, means are introduced to enable users of the recognition system to restrict results to certain stages of a pattern's development, e. g. for forecasting purposes, all in a consistently fuzzy manner.*

Keywords— Fuzzy classification, fuzzy automata, multivariate time series, pattern recognition.

1 Introduction

Fuzzy sets have been successfully used for a long time for a model-free or model-based representation of time series or time series sequences, e. g. for the classification and prediction of nonlinear systems behaviour [1–4]. This article aims at extending the fuzzy time series classification approach of [2, 3] to an *always aware* online recognition system for nonstationary recurring patterns in multivariate streaming time series. *Always aware* refers to the ability of the recognition system presented in this paper to constantly expect to find an (incompleted) pattern in a time series in any of its possible stages of development, which will be achieved by classification of pattern subsequences. It will be shown how the computationally expensive classification results for all possible subsequences can be obtained at almost no additional cost compared to the classification of completed patterns by relating the recognition procedure to ideas from fuzzy automata.

2 Fuzzy Modelling of Time Series Patterns

2.1 A Multivariate Parametric Fuzzy Set

In this section a parametric membership function type will be presented, which will serve as the common basis for the classifiers developed in this article. For the univariate case, the membership function is defined in (1). Based upon a potential function, it has already been successfully used in numerous applications such as process or medical surveillance [1], time series modelling and prediction [2, 3] and classifier networks [5]. One important property of this function is its capability of modelling asymmetric fuzzy sets by individual parameters for the left- and right-hand function branches.

$$\mu(x) = \begin{cases} \dfrac{a}{1 + \left(\dfrac{1}{b_l} - 1\right)\left(\dfrac{r-x}{c_l}\right)^{d_l}} & : \quad x < r \\[2ex] \dfrac{a}{1 + \left(\dfrac{1}{b_r} - 1\right)\left(\dfrac{x-r}{c_r}\right)^{d_r}} & : \quad x \geq r \end{cases} \tag{1}$$

The maximum truth value a of the fuzzy set occurs at its modal point: $\mu(x = r) = a$. For normalised fuzzy sets, $a = 1$ holds, as will be the case throughout this paper. The six parameters $b_{l/r}$, $c_{l/r}$ and $d_{l/r}$ determine the extent and shape of the class. From Fig. 1, the effect of b $(0 < b < 1)$ and c $(c > 0)$ can be understood. Semantically, they correspond to the class borders. The d parameter $(d \geq 2)$ influences the shape of μ's descent to zero, with increasing d leading to a sharper descent and $d \to \infty$ resulting in a rectangular shape. This is particularly interesting since crisp sets may be represented by (1) as well.

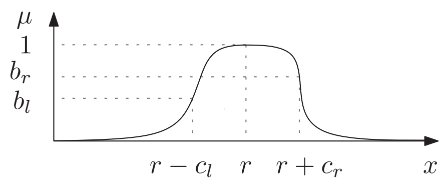

Figure 1: Normalised parametric membership function (1).

The parameters of this membership functions may be chosen based upon expert knowledge on the one hand, and automatically be computed [1] from a set of learning data on the other hand.[1] Both options benefit from the interpretability of these parameters.

A probably unique advantage of this function concept is that a multivariate membership function—based upon a conjunction of one-dimensional sets using a compensatory n-fold Hamacher intersection—may also be obtained in *parametric* form [1]. The n-fold operator is given by (2). The resulting multidimensional membership function is, for a simplified case of symmetric left- and right-hand branches, given by (3). Additionally (and omitted here for brevity), the class space may optionally be rotated in the underlying feature space for each class individually, leading to $(n - 1)$ further rotation angle parameters, which can be obtained from a principal component analysis of the learning data. Fig. 2 depicts two examples of membership functions in a two-dimensional feature space.

$$\cap^n_{Ham}\, \mu_i = \frac{1}{\dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \frac{1}{\mu_i}} \tag{2}$$

[1] The parameters of (1) are computed from data as follows: Firstly, the modal point r is being calculated by averaging the object values. The left- and right-hand side extents $c_{l/r}$ are chosen such that all learning objects, augmented by a so-called elementary fuzziness, lie within these class borders. Finally, the shape parameters b and d parameters are—if not specified manually—being computed to best fit the dispersion of the data.

ISBN: 978-989-95079-6-8

$$\mu(\boldsymbol{x}) = \frac{a}{1 + \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{b_i} - 1\right)\cdot\left|\frac{x_i - r_i}{c_i}\right|^{d_i}} \quad (3)$$

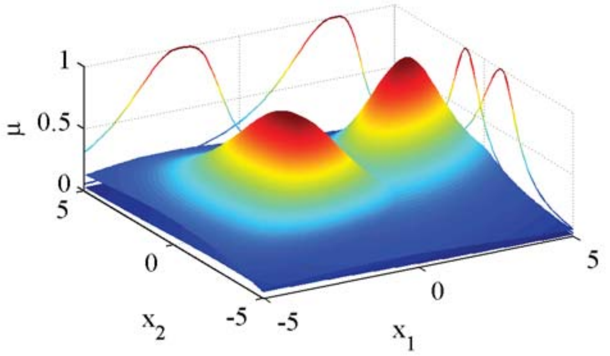

Figure 2: Two examples of two-dimensional fuzzy classes.

2.2 *A Fuzzy Model for Time Series Patterns*

The membership function concept introduced in section 2.1 forms the basis of a classifier model for multivariate time series sequences [2]. It allows to compute a continuous degree of similarity $\mu \in [0, 1]$ between a measured time series pattern and the fuzzy pattern model. Therefore pattern matching is being treated as a fuzzy classification task. The model described hereafter may be employed for equidistantly sampled univariate or multivariate time series.

Each point $\boldsymbol{x}(i)$ of a time series pattern (length: L sample points) is being described by a membership function $\mu_{P,i}(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^n$, $i = 1, \ldots, L$. This model therefore exhibits a soft tolerance towards noise and other sources of imprecision. To compare a time series sequence against the model, i. e. to classify a pattern given by L points $\boldsymbol{x}(1), \ldots, \boldsymbol{x}(L)$, the individual classification results of each point are combined to an overall degree of similarity $\mu \in [0, 1]$ by means of a fuzzy conjunction:

$$\mu = \mu_{P,1}(\boldsymbol{x}(1)) \cap \cdots \cap \mu_{P,L}(\boldsymbol{x}(L)) \quad (4)$$

If we recall the ideas of section 2.1 which led to the multivariate parametric membership function, it appears reasonable to reuse the same conjunction operator given by (2). In that manner, the overall classifier for the entire time series pattern could also be thought of as *one* fuzzy class created from the conjunction of $(L \cdot n)$ univariate fuzzy sets, forming an $(L \cdot n)$-dimensional membership function. Thusly classification of time series sequences would very well remain in line with the multivariate classification paradigm behind section 2.1. One key advantage of this model—especially in comparison to many *black box* models such as neural networks—is its support for partial classification, i. e. for the classification of any possible subsequence, simply by intersecting the underlying classification results for the sample points available. We will make use of this advantageous property later on.

Fig. 3 depicts an example of an univariate ($n = 1$) pattern (length $L = 286$ sample points) described by this fuzzy time series model, learned from a set of 14 instances of this pattern (*Coffee* dataset, available from [6]). By means of this figure we may also come to the interpretation of the pattern classifier as a corridor with soft boundaries. A measured pattern will, depending on its degree of similarity, more or less follow this corridor. The properties of the underlying membership function ensure that no pattern will be absolutely rejected ($\mu = 0$) by this classifier, thusly enabling a smooth, non-switching behaviour of the recognition system.

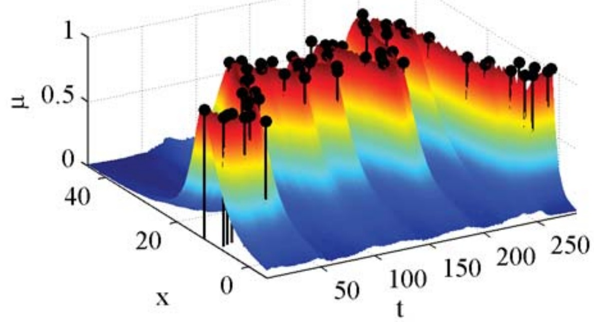

Figure 3: Fuzzy time series model along with a noisy pattern.

While the presented classifier aims at the recognition of recurring patterns that are similar in terms of absolute values, variations such as offsets or scaling may, at least to a certain extent, be covered by suitable preprocessing steps like normalisation. As in section 2.1, the time series pattern model may either be formulated based upon expert knowledge—eased by the interpretability of the individual parameters—or be result of a learning step using a set of pattern instances of the same phenomenon, employing the learning algorithm from section 2.1 for each point of the pattern. The latter is especially interesting in conjunction with clustering or motif mining algorithms. As there is no transform involved, the original properties of the pattern ensemble (energy content, statistics etc.) are mostly being preserved by this model. Furthermore, short-term forecasting of a time series with a partially elapsed pattern is made possible in a straightforward manner based upon the knowledge represented by the model.

3 Online Pattern Recognition

3.1 *Task Description*

The task of online pattern recognition—as we would like to understand it in the context of this work—can be described by means of Fig. 4. In a streaming univariate or multivariate time series $\boldsymbol{x}(t)$, we are looking for instances of a known pattern. *Online* pattern recognition firstly implies that an occurrence of the pattern has to be permanently expected. But in addition to that, we aim at detecting incompleted patterns, which would enable short-term predictions of the time series based on known patterns. For applications such as machine diagnosis, for instance, this facilitates preventive maintenance actions before a minor damage develops into a severe one.

At the current point in time t_{now} (as sketched in Fig. 4), the model containing the desired pattern has to be adjusted in its relative position τ such that the subsequence $\boldsymbol{x}([t_{\text{now}} - \tau] \ldots t_{\text{now}})$ optimally matches the partial model. τ then represents the time already elapsed in the pattern.

Summarisingly, an online pattern recognition system has to check if a pattern in a time series can be found at the current point in time, and if so, at which stage of development τ.

ISBN: 978-989-95079-6-8

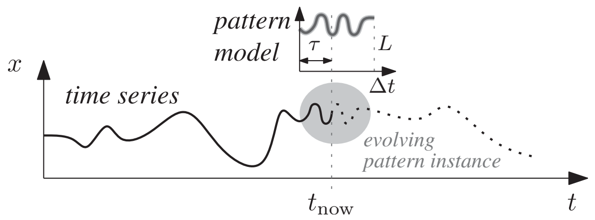

Figure 4: Online matching of evolving patterns.

3.2 Fuzzy Online Recognition

For reasons outlined above, we would like to treat the pattern matching problem as a fuzzy classification task, with time series subsequences (length τ, $\tau \leq L$) being input to a suitable classifier representing a pattern of length L. For each point in time t, this would yield classification results $\mu(t,\tau)$ for every possible value of τ. Semantically, $\mu(t_{\text{now}},\tau)$ represents the similarity of the time series subsequence $\boldsymbol{x}([t_{\text{now}}-\tau]\dots t_{\text{now}})$ to the first part (length τ) of the pattern model/classifier.

In a truly fuzzy sense[2], *each* pattern would be found at *every* point in time in *all* possible stages of development τ, expressed by non-zero classification results $\mu(t,\tau) > 0\ \forall\tau$, even though $\mu(t,\tau)$ may, of course, be negligibly small. At any point in time t, fuzzy classification of evolving time series patterns therefore results in a *fuzzy set* $\mu(t,\tau)$—instead of just *one* truth value $\mu(t)$, as in a "conventional" fuzzy classification task, cf. Fig. 5. Representing the recognition result as a fuzzy set reflects the fuzziness of the recognition task. It is a joint fuzzy representation of the following information:

1. Could the pattern be found at the current point in time?
2. What is the pattern's state of development τ?

Both questions can—in real-world time series—not be answered precisely and unambigously. $\mu(t,\tau)$ commensurates with this vagueness. Furthermore, it may be interpreted as a fuzzy specification of τ, resembling the concept of fuzzy numbers.[3]

3.3 Representation as Fuzzy Automaton

In section 2.2, a suitable classifier model for patterns in equidistantly sampled time series has been introduced. As this model supports partial classification, it is well suited for the online pattern matching problem. In an online recognition system, however, obtaining $\mu(t,\tau)$ for every possible value of τ by classifying all respective subsequences $\boldsymbol{x}([t-\tau]\dots t)$ would be a rather time-consuming procedure. More specifically, it appears incommensurate to recompute the whole fuzzy set $\mu(t,\tau)$ if only one new datum $\boldsymbol{x}(t)$ is being presented to the recognition system at time t. We therefore would like to reuse the classification result $\mu(t-1,\tau-1)$ of the subsequence $\boldsymbol{x}([(t-1)-(\tau-1)]\dots t-1)$ when computing $\mu(t,\tau)$ for $\boldsymbol{x}([t-\tau]\dots t)$.

[2] As an unofficial definition, we would like to understand "true" fuzziness as a property of models and methods which do not (explicitely or implicitely) perform crisp decisions or cut-off operations. For instance, a "truly" fuzzy set has infinite support, thusly all elements x of the underlying universe of discourse belong to this set with $\mu(x) > 0$. As a counterexample, triangular fuzzy sets feature crisp and finite extents, and would therefore not be regarded as "truly" fuzzy models.

[3] Although not being a fuzzy number in the sense of Dubois and Prade [7], especially owing to the very likely multimodality, it matches the spirit of a fuzzy number as a fuzzified representation of a real-valued number (here: τ) quite well.

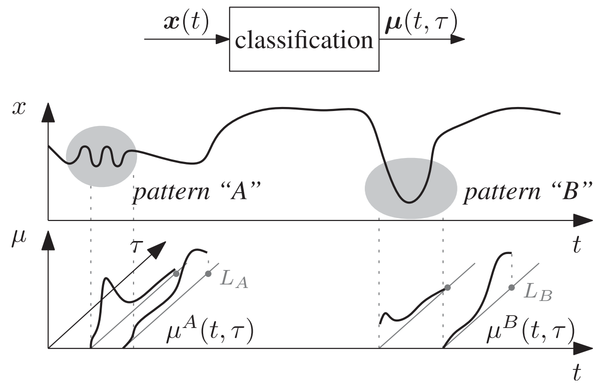

Figure 5: Fuzzy online recognition of two patterns "A" and "B" (length L_A and L_B). At each point in time t, the maximum truth values of the recognition results $\mu^{A/B}(t,\tau)$ w. r. t. τ point to the relative positions τ associated to the best match of the partially elapsed patterns. The fuzzy sets $\mu^{A/B}$ are shown at two different stages of development τ for each pattern.

For discrete time patterns of length L, the pattern recognition result $\mu(t,\tau)$ consists of L truth values ($\tau = 1,\dots,L$). This result may be represented by means of a fuzzy state automaton [8], as sketched in Fig. 6. Being a fuzzy automaton, each state—corresponding to a certain value of τ—is active to a certain degree $\mu(t,\tau)$. At each point in time t, the automaton is triggered with a discrete event in the form of a new input value $\boldsymbol{x}(t)$, leading to new activation levels for each state. An activation level $\mu(t,\tau)$ depends on the current input value $\boldsymbol{x}(t)$ and the former activation level of the previous state:

$$\mu(t,\tau) = f(\mu(t-1,\tau-1), \boldsymbol{x}(t)) \tag{5}$$

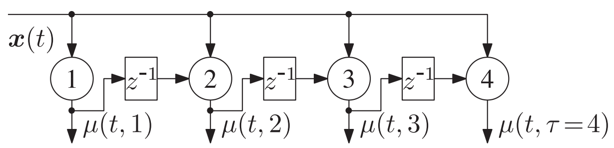

Figure 6: (Informal) fuzzy state automaton representation of a time series pattern recognition system, here for a pattern consisting of four sample points.

Recalling the internals of the time series pattern classifier from section 2.2, the overall classification result was gained from a compensatory Hamacher conjunction (cf. (4)) of individual classification results $\mu_{P,i}$ of time series sample points $\boldsymbol{x}_i$. Thusly $\mu(t-1,\tau-1)$ consists of a conjunction of $\tau-1$ truth values. $\mu(t,\tau)$ could therefore be obtained from a conjunction of $\mu(t-1,\tau-1)$ and the partial classification result $\mu_{P,\tau}(\boldsymbol{x}(t))$ for the τ^{th} point of the time series pattern. However, this conjunction should reflect the weight of the $\tau-1$ truth values that contributed to $\mu(t-1,\tau-1)$, so that $\mu_{P,\tau}(\boldsymbol{x}(t))$ does not outweigh all previous information. A suitable conjunction operator will be derived in the following section, which will enable us to rewrite (5) to such a weighted conjunction:

$$\mu(t,\tau) = \mu(t-1,\tau-1)\ {}^{(\tau-1)}\!\cap^{1}\ \mu_{P,\tau}(\boldsymbol{x}(t)) \tag{6}$$

ISBN: 978-989-95079-6-8

Equation (6) forms the basis for the mode of operation of the fuzzy automaton from Fig. 6, and will enable us to perform online recognition of time series patterns much more efficiently.

3.4 Weighted Compensatory Hamacher Conjunction

The compensatory Hamacher conjunction of n truth values μ_i ($i = 1, \ldots, n$) has already been introduced in (2). By rearranging the components of (2) into two sets of truth values, we may formulate the compensatory Hamacher conjunction of $n = n_1 + n_2$ truth values as:

$$\begin{aligned} \mu_1 \cap \cdots \cap \mu_n &= \frac{1}{\frac{1}{n_1+n_2}\left(\sum_{i=1}^{n_1}\frac{1}{\mu_i} + \sum_{j=n_1+1}^{n}\frac{1}{\mu_j}\right)} \\ &= \frac{1}{\frac{1}{n_1+n_2}\left(\frac{n_1}{\mu_1 \cap \cdots \cap \mu_{n_1}} + \frac{n_2}{\mu_{n_1+1} \cap \cdots \cap \mu_n}\right)} \\ &= (\mu_1 \cap \cdots \cap \mu_{n_1}) \ {}^{n_1}\!\cap^{n_2} \ (\mu_{n_1+1} \cap \cdots \cap \mu_n) \end{aligned} \tag{7}$$

Based upon (7) we may now define the (n_a, n_b)-weighted compensatory Hamacher conjunction of two truth values μ_a and μ_b, which was in a similar manner also derived in [9]:

$$\mu_a \ {}^{n_a}\!\cap^{n_b} \ \mu_b = \frac{1}{\frac{1}{n_a+n_b}\left(\frac{n_a}{\mu_a} + \frac{n_b}{\mu_b}\right)} \tag{8}$$

Thusly it becomes possible to compute the conjunction of new truth values with existing conjunction results without having to store the individual truth values leading to the existing results and while exactly retaining the weight of each truth value that contributed to the overall result.

Finally, we may define the vectorial weighted conjunction of two equal-sized truth vectors $\boldsymbol{\mu}_a$ and $\boldsymbol{\mu}_b$ using weight vectors $\boldsymbol{n}_a$ and $\boldsymbol{n}_b$ containing the individual weights for the respective elements:

$$\boldsymbol{\mu}_a \ {}^{\mathbf{n_a}}\!\cap^{\mathbf{n_b}} \ \boldsymbol{\mu}_b = \begin{pmatrix} \vdots \\ \frac{1}{\frac{1}{n_{a,i}+n_{b,i}}\left(\frac{n_{a,i}}{\mu_{a,i}} + \frac{n_{b,i}}{\mu_{b,i}}\right)} \\ \vdots \end{pmatrix} \tag{9}$$

3.5 Update Equations for the Fuzzy Automaton

Recalling (6) and employing the weighted conjunction operator from (8), we may now formulate the update equation for each fuzzy state of the automaton representation of the recognition results $\mu(t, \tau)$ from section 3.3. Equation (6) holds for $\tau \geq 2$, as the first state of the automaton does not have a predecessor state to obtain $\mu(t-1, \tau-1)$. Since the weight of the non-existing zero-th state would be zero anyway, (6) can (for $\tau = 1$) be thought of to reduce to solely the classification result for the first sample point, cf. (10). All further results $\mu(t, \tau \geq 2)$ are computed by a weighted conjunction of antecedent recognition results and the classification results $\mu_{P,\tau}$ of the new sample point $\boldsymbol{x}(t)$ available at each point in time.

$$\mu(t, 1) = \mu_{P,1}(\boldsymbol{x}(t)) \tag{10}$$

To obtain a vectorial form of the update equations, we define $\boldsymbol{\mu}_P(t)$ as the vector containing the individual membership values of $\boldsymbol{x}(t)$ to all classes constituting the fuzzy time series pattern model, and $\boldsymbol{\mu}_\tau(t)$ as the state vector containing all levels of activation of the automaton, and thusly the overall pattern recognition results:

$$\boldsymbol{\mu}_P(\boldsymbol{x}(t)) = \begin{pmatrix} \mu_{P,1}(\boldsymbol{x}(t)) \\ \vdots \\ \mu_{P,L}(\boldsymbol{x}(t)) \end{pmatrix} \tag{11}$$

$$\boldsymbol{\mu}_\tau(t) = \begin{pmatrix} \mu_{\tau,1}(t) \\ \vdots \\ \mu_{\tau,L}(t) \end{pmatrix} = \begin{pmatrix} \mu(\tau, 1) \\ \vdots \\ \mu(\tau, L) \end{pmatrix} \tag{12}$$

Summarisingly, the recursive update equation for the recognition result (the fuzzy set $\mu(t, \tau)$) or its vectorial representation $\boldsymbol{\mu}_\tau(t)$ can be condensed from (5), (6) and (9) to (13):

$$\begin{aligned} \boldsymbol{\mu}_\tau(t) &= f(\boldsymbol{\mu}_\tau(t-1)), \boldsymbol{\mu}_P(\boldsymbol{x}(t))) \\ &= \left(\begin{pmatrix} 1 \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0}^T & 0 \\ \boldsymbol{I} & \mathbf{0} \end{pmatrix} \cdot \boldsymbol{\mu}_\tau(t-1)\right) \ {}^{\mathbf{n}_\tau}\!\cap^{\mathbf{1}} \ \boldsymbol{\mu}_P(\boldsymbol{x}(t)) \end{aligned} \tag{13}$$

$$\boldsymbol{n}_\tau = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ L-1 \end{pmatrix} \tag{14}$$

Incorporating the weight vector $\boldsymbol{n}_\tau$, this finally results in:

$$\boldsymbol{\mu}_\tau(t) = \begin{pmatrix} \frac{1}{\frac{1}{1}\cdot\left(\frac{0}{1} + \frac{1}{\mu_{P,1}(\boldsymbol{x}(t))}\right)} \\ \vdots \\ \frac{1}{\frac{1}{i}\cdot\left(\frac{i-1}{\mu_{\tau,i-1}(t-1)} + \frac{1}{\mu_{P,i}(\boldsymbol{x}(t))}\right)} \\ \vdots \\ \frac{1}{\frac{1}{L}\cdot\left(\frac{L-1}{\mu_{\tau,L-1}(t-1)} + \frac{1}{\mu_{P,L}(\boldsymbol{x}(t))}\right)} \end{pmatrix} \tag{15}$$

Apart from the $(L-1)$ weighted conjunctions of truth values that are required at each time step, the online recognition of a (partially elapsed) fuzzy pattern therefore does not require more underlying classification steps and computations of $\mu_{P,\tau}$ than would be needed just for the fuzzy classification and recognition of only completed patterns in streaming time series.

4 Post-Processing Recognition Results

4.1 Windowing of $\mu(t, \tau)$

The fuzzy set $\mu(t, \tau)$—describing the recognition results of a pattern found at the current point in time t in all possible stages of development τ—is a rather complex result containing a wealth of information. In many cases it will thusly have to be narrowed down to particularly interesting or relevant parts. In addition to that, shorter subsequences of the pattern will be found more often in a time series than the whole pattern. This results in a tendency to relatively often obtain high degrees

ISBN: 978-989-95079-6-8

of similary $\mu(t,\tau)$ for small values of τ, which will later on diminish if the time series does not develop further into an instance of the pattern. For longer patterns, a user will therefore always have to gauge and find a trade-off between the necessity of detecting patterns at an early stage of development τ (with a higher risk of a false or premature recognition result) and the importance of a reliable recognition result (which can ultimately only be achieved at the end of an evolving pattern). This trade-off cannot be universally formulated in a crisp manner, and neither does it seem advisable to do so.

These considerations lead us to the idea of letting the user of the recognition system focus on certain stages development τ of a pattern in a soft manner, such that the recognition system only reports matches of the evolving pattern for values of τ falling within certain (fuzzy) boundaries. Subsequently, we will call these boundaries the *fuzzy window of interest for* τ. We will represent this window of interest by means of a fuzzy set $\mu_w(\tau)$ defined over all possible values of τ, with $\mu_w(\tau) \to 0$ indicating low interest in a recognition result for this particular value of τ, and $\mu_w(\tau) \to 1$ indicating high interest.

To obtain a *windowed* pattern recognition result $\tilde{\mu}(t,\tau)$, the window of interest $\mu_w(\tau)$ and the recognition result $\mu(t,\tau)$ are combined as in (16) by a conjunction[4] of the two fuzzy sets. Semantically, this corresponds to a coincidence of a pattern being found at a certain stage of development and the user's subjective assessment of the importance of this particular stage.

$$\tilde{\mu}(t,\tau) = \mu(t,\tau) \cap \mu_w(\tau) \quad \forall\tau \tag{16}$$

Equation (16) in its general form includes some interesting special cases, such as the following: By definition, the truth values in $\mu(t,\tau)$ appear equally important to the user regardless of τ, so that higher values of μ will often be found for smaller values of τ, although the pattern has just begun to evolve. Following the windowing approach would also enable us to (for example: linearly) weigh a recognition result $\mu(t,\tau)$ according to the stage of development τ of the pattern by chosing a suitable $\mu_w(\tau)$. To achieve this, $\mu_w(\tau)$ has—in this special case—to be chosen as a fuzzy set with (linearly) increasing truth values, reaching $\mu_w(\tau = L) = 1$ at the end of the pattern (length L). If we, as an example, choose the algebraic product for the operator $\cap$ in (16), we would simply obtain (17) for a thusly weighted/windowed recognition result.

$$\tilde{\mu}_{linear}(t,\tau) = \mu(t,\tau) \cdot \frac{\tau}{L} \tag{17}$$

4.2 Deriving Decisions

In order to further process the (windowed) recognition results $\mu(t,\tau)$ or $\tilde{\mu}(t,\tau)$, e. g. when serving as a rationale for subsequent actions, it will often be necessary to come to a crisp decision regarding the similarity and stage of development of a pattern found in a time series. In these cases, the fuzzy set $\tilde{\mu}(t,\tau)$ has to be condensed to one or more crisp parameters.

Although many defuzzification methods for fuzzy sets are available [10], we would—for the sake of simplicity—like to restrict ourselves in this article to a maximum approach (first of maxima, FOM). The resulting crisp value τ^* from $\tilde{\mu}(t,\tau)$ points to the earliest stage of development of the pattern yielding the highest degree of similarity $\tilde{\mu}^*$ to the respective time series subsequence, both of which may be represented as a fuzzy singleton $\tilde{\mu}^*(t,\tau^*)$ for each point in time t.

5 Example

Fig. 7 shows a time series containing one instance of the pattern modelled and depicted in Fig. 3. For illustrative purposes, the instance has been highlighted, as the surrounding parts of the the time series are relatively similar in amplitude and shape.

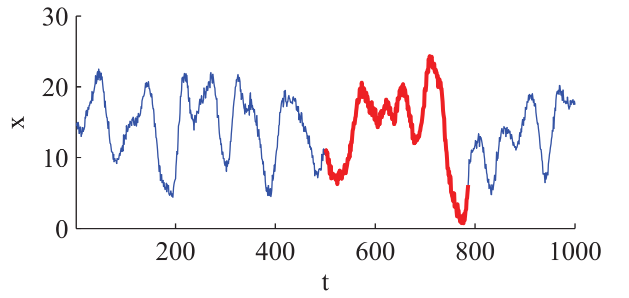

Figure 7: Univariate time series with a pattern instance.

Employing the fuzzy model from Fig. 3, we are now performing an online recognition of this pattern in the time series of Fig. 7. While advancing in time t through all points of the time series, the pattern is constantly being expected to be found in all[5] possible stages of development τ. The classification results for each point in time t—evolving fuzzy sets $\mu(t,\tau)$ defined over τ—are being shown in Fig. 8.

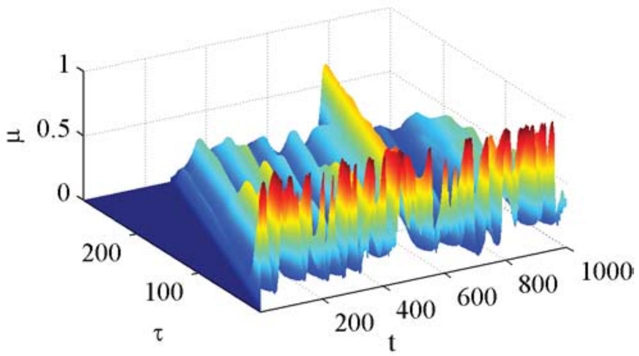

Figure 8: Results for an online recognition of the time series pattern of Fig. 3.

These recognition results $\mu(t,\tau)$ reveal high degrees of similarity of time series subsequences to the first part (i. e. small values of τ) of the pattern, and, in fact, it can be understood from Fig. 7 that the ascending first slope can (with slight variations) be found repeatedly throughout the time series. Apart from that, the progression of a detected instance is well visible in the course of $\mu(t,\tau)$. Only once a completed pattern is being found with distinctly high values for $\mu(t,\tau = L)$, which point to the pattern instance embedded in this time series.

To detect a pattern ahead of time ($\tau < L$), e. g. for forecasting purposes, the recognition result has to be well-founded, and therefore based upon a adequately large number of recognised

[4] As conjunction operator, a non-compensatory operator—such as all T-norm operators—should be chosen to ensure $\tilde{\mu}(t,\tau) \le \mu(t,\tau)$.

[5] In the first part of the time series, the pattern may only be found at early stages of development $\tau < t$, i. e. $\mu(t,\tau > t) = 0$ must hold.

ISBN: 978-989-95079-6-8

sample points τ on the one hand, whilst τ being small enough to leave sufficient scope for actual forecasting. We will tackle both problems of "premature" pattern recognition and formulating a compromise for a suitable value of τ delivering usable recognition results by following the windowing approach for $\mu(t, \tau)$ presented in section 4.1. As an example, we would like to concentrate on recognition results for pattern instances being at least half-way, but not yet fully completed. Once again employing the unimodal parametric membership function from (1), we formulate a fuzzy window of interest $\mu_w(\tau)$ for this certain stage of the pattern's development, cf. Fig. 9.[6] Due to the fuzzy nature of the presented method, slight variations of $\mu_w(\tau)$ and its parameters will not alter the overall system behaviour and recognition results in a drastic, switching manner. This eases the design of $\mu_w(\tau)$, as no "critical" decisions have to be made when choosing its parameters.

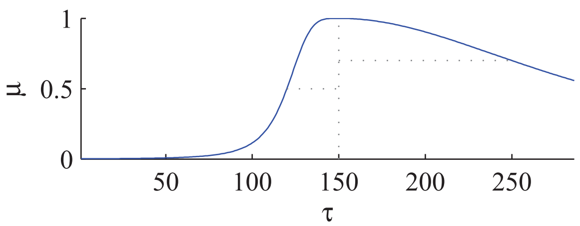

Figure 9: Fuzzy window of interest $\mu_w(\tau)$ for pattern recognition results $\mu(t, \tau)$ from Fig. 8.

For each t, the recognition result (fuzzy set) $\mu(t, \tau)$ was processed as given by (16) using a Hamacher product with $\mu_w(\tau)$. The windowed recognition results $\tilde{\mu}(t, \tau)$ are shown in Fig. 10. As a final step, a simple decision strategy was followed to generate crisp recognition results in singleton form for each point in time t. For each fuzzy set $\tilde{\mu}(t, \tau)$, the modal (maximum) value $\tilde{\mu}^*(t, \tau^*)$ was determined. Any result for τ^* with a corresponding truth value falling below a threshold $\tilde{\mu}^*(t, \tau^*) < 0.5$ was discarded. Results of this decision procedure are marked in Fig. 10. As can be seen in this figure, two candidate pattern instances are being identified at the desired stage of development τ, the first one of which vanishing quickly. The actual pattern instance embedded in the time series of Fig. 7 is being recognised reliably throughout the fuzzy window of interest.

6 Conclusions

This article dealt with the online recognition of patterns in multivariate time series. The vectorial recursive update equation for the recognition results derived in section 3.5 allows to obtain a fuzzy recognition result $\mu(t, \tau)$ for all possible stages of development τ of a pattern at almost the same computational cost as required for the fuzzy classification of only completed patterns. By formulating and applying a fuzzy window of interest, the recognition result $\mu(t, \tau)$ can be narrowed down to parts relevant for the respective application in a fuzzy manner. This also eases a subsequent crisp decision step.

Further work will concentrate on the application of the model and methods presented here to the recognition and prediction of recurring patterns in energy load time series. Besides that, an enhancement of the decision procedure appears promising. A desirable goal would be a *fuzzy* decision process delivering *parametric* fuzzy information for the stage of development of a detected pattern. This could be more directly employed in subsequent (e. g. prediction) steps without discarding the fuzziness always inherent in a real-world recognition result.

[6] The parameter were chosen manually: $r = 150$, $b_l = 0.5$, $b_r = 0.7$, $c_l = 30$, $c_r = 100$, $d_l = 4$, $d_r = 2$. For $\tau < r$, $d_l = 4$ and $c_l = 30$ ensure a quick decrease of $\mu_w(\tau)$, whereas the right-hand side parameters b_r, c_r, d_r provide a gentle decrease of μ_w for $\tau > r$.

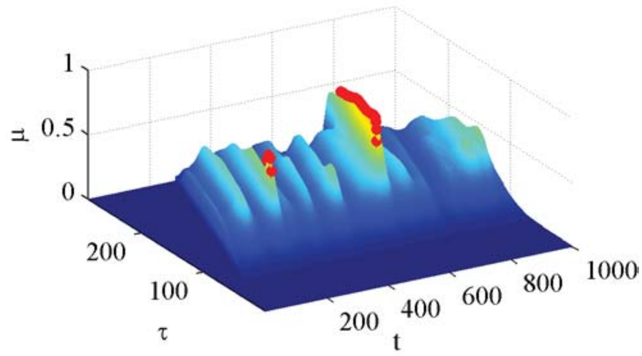

Figure 10: Windowed recognition results $\tilde{\mu}(t, \tau)$. Singleton results $\tilde{\mu}^*(t, \tau^*)$ of the decision procedure are marked with dots (slightly increased for better visibility).

References

[1] Steffen F. Bocklisch and Norman Bitterlich. *Fuzzy systems in computer science*, chapter Fuzzy pattern classification – methodology and application. Vieweg, Wiesbaden, 1994.

[2] Michael Päßler and Steffen F. Bocklisch. *Fuzzy Control: Theory and Practice*, chapter Fuzzy time series analysis, pages 331–345. Springer Verlag, 2000.

[3] Peter Bretschneider. *Ein Beitrag zur Vorhersage musterbasierter nichtlinearer stochastischer Signale*. PhD thesis, TU Ilmenau, 2002.

[4] Bernd Möller and Uwe Reuter. Prediction of uncertain structural responses using fuzzy time series. *Computers and Structures*, 86(10):1123–1139, 2008.

[5] Arne-Jens Hempel and Steffen F. Bocklisch. A conception for networks of fuzzy pattern classifiers. In *Proceedings 13th East West Fuzzy Colloquium*, Zittau, 2006.

[6] Eamonn Keogh, Xiaopeng Xi, Li Wei, and Chotirat Ann Ratanamahatana. The UCR time series classification/clustering homepage. http://www.cs.ucr.edu/~eamonn/time_series_data/, 2006.

[7] Didier Dubois and Henry Prade. Fuzzy numbers: An overview. In James C. Bezdek, editor, *Analysis of Fuzzy Information, Vol.I: Mathematics and Logic*, pages 3–39. CRC Press, Boca Raton, FL, 1987.

[8] William G. Wee and King-Sun Fu. A formulation of fuzzy automata and its application as a model of learning systems. *IEEE Transactions on Systems Science and Cybernetics*, 5(3):215–223, 1969.

[9] Ullrich Scheunert. *Fuzzy-Mengen-Verknüpfung und Fuzzy-Arithmethik zur Sensor-Daten-Fusion*. PhD thesis, Chemnitz University of Technology, 2001.

[10] Werner Van Leekwijck and Etienne E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, 108(2):159–178, 1999.

ISBN: 978-989-95079-6-8