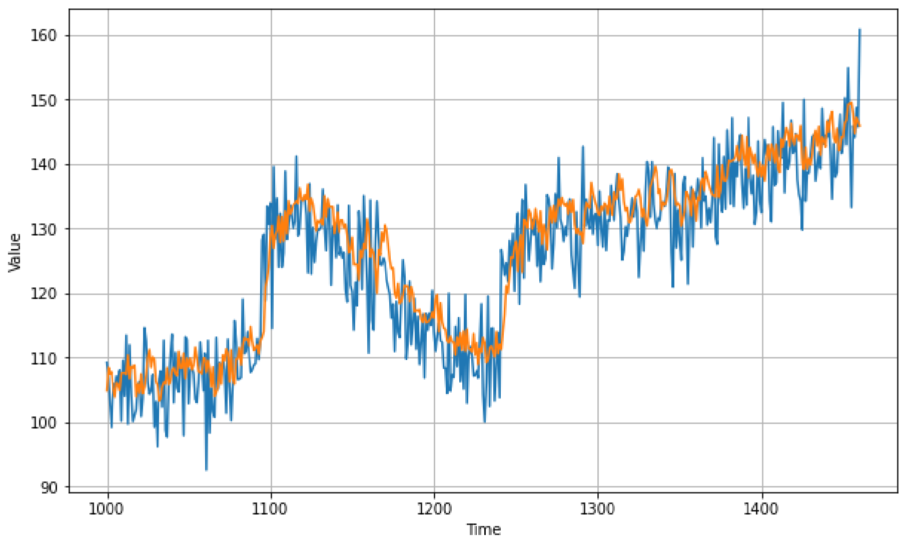
Fakulta elektrotechniky a informatiky

Department of Cybernetics and Artificial Intelligence

Subject: **Fuzzy Systems**



Exercise - Group n.4

Interpretable cognitive learning with spatial attention for high-volatility time series prediction

Keys: Fuzzy, Time Series, Forecasting, Causality

Author:
QuietXO

Obsah

1 Introduction

In this work we are going to look into fuzzy time series which is time series analysis that incorporates the concept of fuzzy logic to deal with uncertainty and vagueness in the data. In traditional time series analysis, each data point is represented as a crisp value. However, in many real-world scenarios, the data can be imprecise or uncertain. Fuzzy time series offers a flexible framework for modeling such uncertain and vague data by representing the data as fuzzy sets.

The key idea behind fuzzy time series is to use fuzzy sets to represent the values in a time series. This allows us the use of fuzzy logic to capture the inherent uncertainty and imprecision in the data. Fuzzy time series models can be used to make predictions and forecasts, identify patterns and trends, and perform classification and clustering tasks.

We may apply fuzzy time series in a wide range of domains, including finance, economics, environmental science, and engineering. It has proven to be a powerful tool for modeling and analyzing time series data in scenarios where the data is noisy or imprecise. With the increasing availability of large, complex data sets, the importance of fuzzy time series is likely to continue to grow in the future.

Adding fuzzy logic to our solution also enables for us to make it more salable and most importantly interpretable, since we can look on what grounds was the decision made.

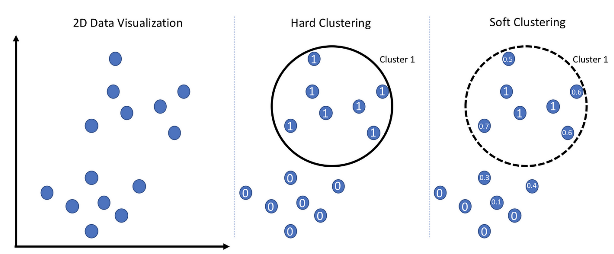
2 Related Terms

We will start by looking into explain some terms commonly found terms in this type of work, since their basic knowledge is essential for understanding of this work, but only the necessary amount and define acronyms for them.

Terms we are going to be using:
→ C-Means 2.1
→ Fuzzy Cognitive Map (FCM) 2.2
→ High-Order FCM (HFCM) 2.3
→ Fuzzy Information Granule (FIG) 2.4

2.1 C-Means

C-Means (or fuzzy C-Means) algorithm, is a soft clustering algorithm used to group data points into clusters based on similarity. It is a widely used algorithm in machine learning and data mining, and is particularly useful when the data points are not clearly separable. It is similar to the K-Means algorithm and both of them do exactly the same, but the use the fuzzy gives us advantage of knowing how much does a certain point belong to a certain cluster center [1].

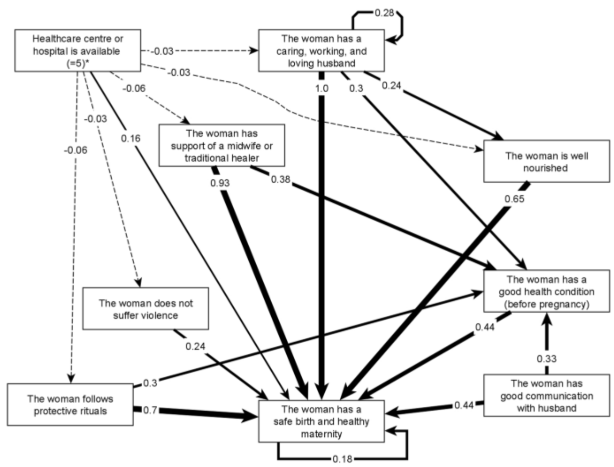


Obr. 1: Comparison between Hard (K-Means) and Soft (C-Means/Fuzzy) Clustering.

2.2 Fuzzy Cognitive Maps

Fuzzy cognitive maps (FCMs) are a type of mathematical model used to represent and analyze complex systems. They are particularly useful for modeling systems where the relationships between the various components are uncertain or vague. FCMs are based on the concept of fuzzy logic, which allows for the representation of imprecise and uncertain information. This way we can show how the stuff in relation affects each other, in which direction and most importantly, by what amount by affing fuzzy logic to it.

FCMs can be used to model a wide range of complex systems, including social, economic, and environmental systems. They are particularly useful for analyzing the behavior of systems over time, as well as predicting the effects of changes to the system.

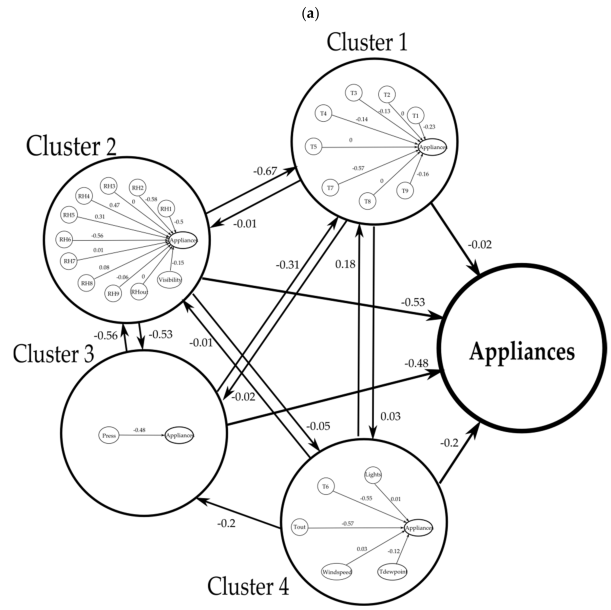


Obr. 2: Representation of an example FCM showing safety relation giving birth.

2.3 High-Order FCMs

High-order fuzzy cognitive maps (HFCMs) are an extension of FCMs that allow for the representation of higher-order interactions between the components in a system. In a traditional FCM, the nodes represent the components of the system, and the edges represent the causal relationships between them. HFCMs extend this concept by allowing the nodes to represent higher-order relationships, such as the interactions between groups of components.

Simply put, HFCMs work on the same principal as FCMs do, but they are built from FCMs instead of just simple data. While FCMs remember the short-term patterns of Time Series, HFCMs try to connect them together and create a long-term pattern [2].

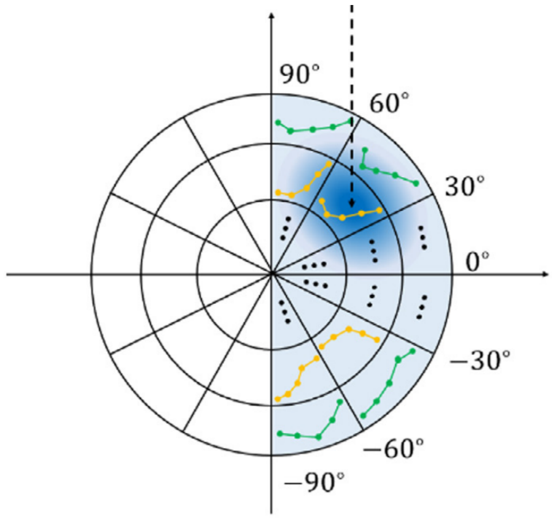


Obr. 3: Dummy example of a High-order Fuzzy Cognitive Map.

2.4 Fuzzy Information Granule

In the field of fuzzy logic and fuzzy systems, a fuzzy information granule (FIG) refers to a group of objects or data points that are similar or related in some way. Fuzzy information granulation is the process of grouping together objects or data points that share similar characteristics, based on the degree of similarity or dissimilarity between them [3].

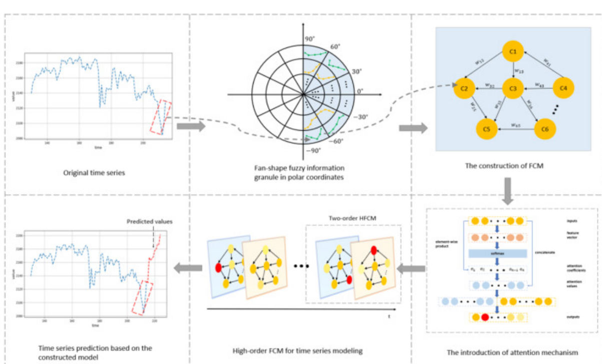
In our case we use them to create trends which we do not want to repeat more then once, since it would be a useless waste of the computational power [2].



Obr. 4: Comparison between Hard (K-Means) and Soft (C-Means/Fuzzy) Clustering.

3 Our Implementation

We were aiming to re-create a work which can be seen in Figure 5 in Python, but we deemed the approach questionable, mostly because some of the formulas didn't really add up, we still use the idea behind this approach, and re-created it on weather data for now, since we need to figure out a way for optimization of data moving out of training range.



Obr. 5: The framework of the proposed approach [2].

3.1 Libraries

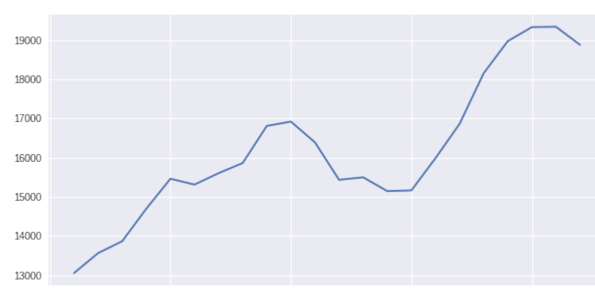
-matplotlib.pyplot: Is used for visualisation of data in 2D array
-pandas: Shows us the data in a table for better orientation
-numpy: Optimized matrix library for python, so we do not waste computing resources
-math: For doing some more advanced calculation as or mean squared error
-pyFTS: A Git-Hub open-source library which includes multiple most commonly used fuzzy logic operations

3.2 Datasets

For our datasets, we picked weather information in Seattle [5], which was successful Section 4.1 and also Google stock price [4] data, which failed and will be described in Section 4.2, since we have to figure out a way of optimization.

3.3 pyFTS Functionality

We are going to demonstrate this library on the in-built dataset called Enrollment which can be seen in Figure 6.

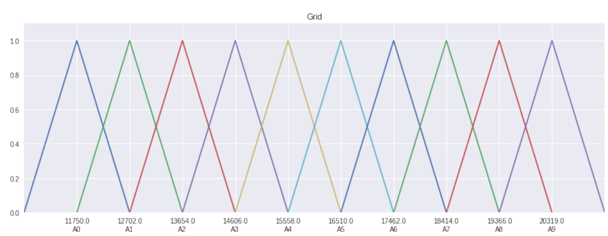


Obr. 6: Enrollment Dataset.

3.3.1 Definition of the Universe

Here we look at the definition of the Universe of Discourse U and Linguistic variable creation. The Universe of Discourse (U) partitioners are responsible for identifying U, split the partitions and create their fuzzy sets. There are several ways to partition U and this has a direct impact on the accuracy of the predictive model. For this example we are using grid partitioning, where all sets are equal. The default membership function is triangular and is shown in Figure 7.

To create this, we will be using the function *pyFTS.partitioners.GridPartitioner*, in which we define how we want to split our data.

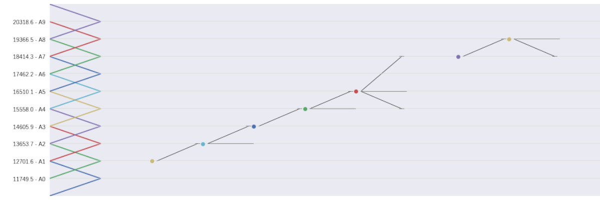


Obr. 7: Enrollment data membership function.

3.3.2 Model fitting

Starting with fuzzyfication, we turn our data into some sort of patterns using the *fuzzyfy* function on our already created sets in Universe of Discourse. Afterwards we use *pyFTS.common.generate_non_recurrent_flrs* function, which will create non-repeating rules for us.

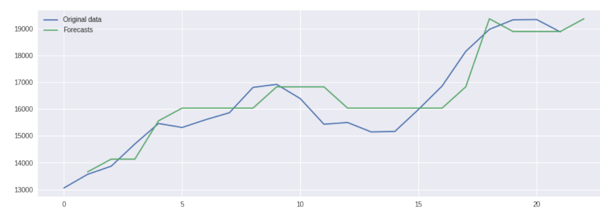
After we prepared everything, we can create the model by using the *pyFTS.models.ConventionalFTS* function, and train it using the *fit* function of this model, which will give us a decision map as seen in Figure 8.



Obr. 8: Decision map of enrollment model.

3.3.3 Forecasting procedure

We prepare data the same way we did for the model fitting, but this time it can be either one value we'd like to know an output for or set of testing data, to see how precisely did our model learn something. For this we will be using the *predict* function of our model, which will return as many values as we send into this function.



Obr. 9: Prediction based on enrollment model.

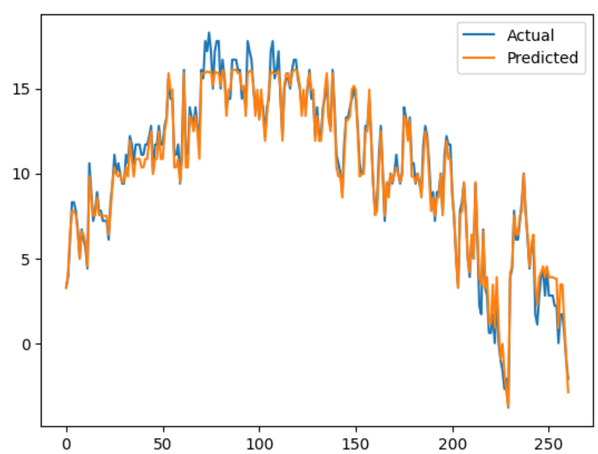
4 Experiments

In this part we will try to a certain degree re-create the model based on the original article and afterwards using the built-in Fuzzy Time Series model function and compare their performance on real world data and not just demo dataset. Firstly, we will take a look at data which is under normal conditions in a certain range (weather temperatures) and after that at data for which the range cannot be fixed (Google stock).

4.1 Weather Data

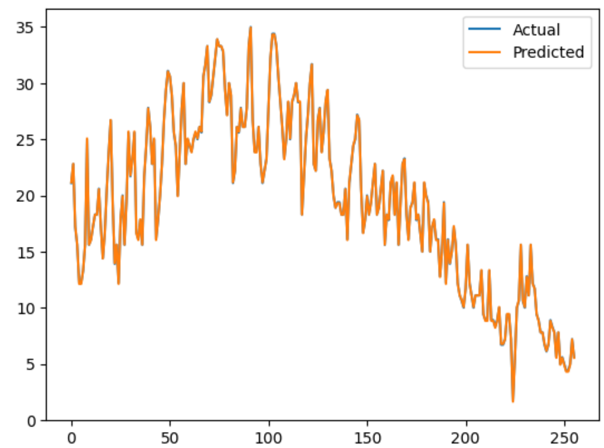
This data contains day-by-day forecast of precipitation, maximum temperature, minimum temperature, wind and type of weather (drizzle, rain, sun, snow, etc.) from which we picked maximum temperature to be our one key information, since that we decided to predict.

We start by dataset preparation by putting our weather data into a FIG, then into FCM and those into HFCM, to re-create the structure of the base article [2] and the result can be seen in Figure 10. The results aren't that bad, but we can see that if the data is just slightly out of the range on which it was trained, this model has a problem adapting to it, since it doesn't know such number.



Obr. 10: Prediction of weather based on article [2] model with RMSE of 0,6677.

But if we would be to use the in-built function of Fuzzy Time Series, other than easier data preparation, the model itself can better handle slight anomalies and is even more precise with this type of almost closed range data as you can see in Figure 11.

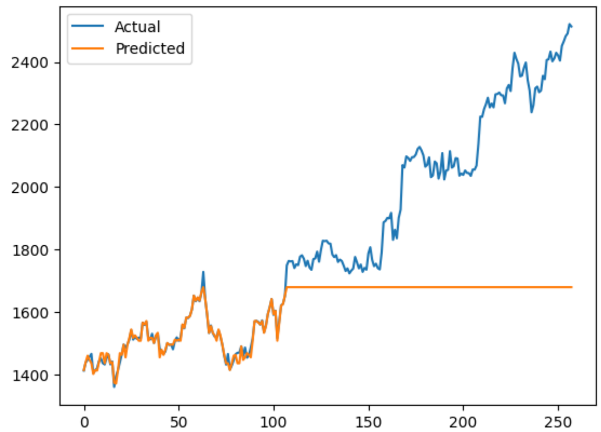


Obr. 11: Prediction of weather based on FTS model with RMSE of 0,0575.

4.2 Stock Data

Once we started to work with stock data, there we decided to use as the key information average of daily opening and closing price, but unlike weather, where the interval is relatively closed, here we can't do decisions based on the input number, but rather based on a certain pattern. It didn't matter which approach we have used, since both of them worked on only previously trained on range and therefore upon reaching numbers outside of this interval, we can see the algorithm crashing in Figure 12.

What we think could help to solve this problem, is a pattern normalization in form of Fuzzy Information Granule as proposed by article [2], since it would eliminate the numbers and we could just shift them back to their place afterwards as a step of our defuzzyfication process. In theory this sounds really promising, but it may not be as simple as just matter of primordial down-shifting to the 0 axis and putting it back to it's original position with up-shifting after the defuzzyfication process.



Obr. 12: Prediction of Google stock price.

5 Conclusion

We may conclude, that fuzzy time series prediction is a powerful technique, that is becoming popular in recent years due to its ability to effectively handle uncertain and imprecise data by applying fuzzy logic to model, is better explainable than some other options such as neural networks and is pretty accurate at predicting time series data. This technique has demonstrated superior performance compared to traditional time series methods in a variety of real-world applications.

Although while fuzzy time series prediction has its limitations, such as the need for careful parameter tuning, the requirement for sufficient training data and data normalization so it can learn the patterns, the benefits of the technique make it a valuable tool for data analysis and decision-making. As the field of fuzzy logic and fuzzy systems continues to advance, it is likely that fuzzy time series prediction will continue to play an important role in solving complex problems and making predictions based on uncertain and imprecise data.

Literatúra

[1] Stelios Krinidis and Vassilios Chatzis. A Robust Fuzzy Local Information C-Means Clustering Algorithm. *10.1109/TIP.2010.2040763*, 2010. 2

[2] Ding Fengqian and Luo Chao. Interpretable cognitive learning with spatial attention for high-volatility time series prediction. *10.1016/j.asoc.2022.108447*, 2022. 3, 6

[3] Lotfi A. Zadeh. Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *10.1016/S0165-0114(97)00077-8*, 1997. 3

[4] Shreenidhi Hipparagi. Google Stock Prediction. *https://www.kaggle.com*, 2023. 4

[5] ANANTH R. WEATHER PREDICTION. *https://www.kaggle.com*, 2023. 4