OSDA Big Homework Report
Neural FCA

Mishan Aliev

December 13, 2022

Introduction

There was used openly available dataset "Acute Inflammations Data Set" from UCI Machine Learning Repository (see [1]). This dataset contains 6 attributes (5 binary and 1 numerical) and 2 target columns. It consists of 120 rows; each instance represents a potential patient.

The main goal of this dataset is to help predict two diseases of urinary system: an acute inflammations of urinary bladder (or cystitis in medical terms) and an acute nephritis.

Dataset

In this research I will focus only on one target value, namely the prediction of cystitis. This is an integer value that can take two values: 0 (no disease) and 1 (its presence). The dataset contains 60 examples of sick people and 60 non-sick people. There are following features:

1. Temperature of patient - displays a temperature of the patient; int values in the segment $[35.5, 41.5]$;
2. Occurrence of nausea - indicates the presence of an occurrence of nausea; "yes" or "no";
3. Lumbar pain - indicates the presence of a lumbar pain; "yes" or "no";
4. Urine pushing (continuous need for urination) - indicates the presence of a continuous need for urination; "yes" or "no";
5. Micturition pains - indicates the presence of micturition pains; "yes" or "no";
6. Burning of urethra, itch, swelling of urethra outlet - indicates the presence of burning of urethra, or itch, or swelling of urethra outlet; "yes" or "no".

Binarization strategies

In this research I used two types of binarization of only numerical attribute "Temperature of patient":

1. **1st type.** Based on [2], I divide all data in column into 2 groups (so, it answers the question whether the patient has a temperature):
   - "no" if temperature $\in [35.5, 37.2]$;
   - "yes" if temperature $\in [37.3, 41.5]$.

   This type of binarization used for models 1, 3.1, 4, 5.
2. **2nd type.** Based on [3], I divide all data in column into 3 groups and then binarize them using One-Hot Encoding. So, the groups are
   - temperature $\in [35.5, 36.4]$;
   - temperature $t \in [36.5, 37.5]$.
   - temperature $t \in [37.6, 41.5]$.

   And this type of binarization used for models 2, 3.2, 6, 7.

Prediction quality measure

I prefer to use the F1 score because it maintains a balance between precision and recall for the classifier and it gives a better measure of the incorrectly classified cases than the accuracy metric (see [4]).

Another technique to select best concepts

Initially, we select the best concepts based on the F1 score, let's try to select best concepts according to accuracy and see results (see models 3.1 and 3.2).

Various nonlinearities to put in the network

By default, in the neural FCA library, ReLU is used as an activation function, let's see how other very popular versions of nonlinearities will manifest themselves in the network. I tested Leaky ReLU and hyperbolic tangent (other nonlinearities are not so common).

Application of modern methods

For comparison with neural FCA, I use modern methods Logistic Regression and Random Forest Classifier with basic parameters. Since the dataset is quite small, their accuracy should be enough. Ordinary neural networks could also be used, but they will definitely just learn our small data and a good comparison will not work.

Results and conclusions

It is important to note that in my research I take the minimum possible count of best concepts (we always try to use minimum count to describe our data) and train neural networks 50000 epochs (experimentally, I got that it makes no sense to train fewer epochs). So, results for models can be seen in Tab. 1.

Model	Binarization type	Best concepts measure	Best concepts count	F1 score	Nonlinearity
1	1st type	F1 score	7	≈ 0.67	ReLU
2	2nd type	F1 score	12	≈ 0.47	ReLU
3.1	1st type	Accuracy	9	≈ 0.52	ReLU
3.2	2nd type	Accuracy	14	≈ 0.59	ReLU
4	1st type	F1 score	7	1	Leaky ReLU
5	1st type	F1 score	7	1	hyperbolic tangent
6	2nd type	F1 score	12	1	Leaky ReLU
7	2nd type	F1 score	12	1	hyperbolic tangent

Table 1: Results table

So, we see from table that our first approaches with model 1 and model 2 we got average results, but quite normal for a completely unused library for neural FCA. Then we try to use another metric to choose best concepts (models 3.1 and 3.2) but we got similar results as expected. And in the end in the last 4 models, the nonlinearity in the network was changed (models 4, 5, 6, 7), which had an incredible impact on the results: we got perfect F1 score everywhere, which means that the new nonlinearities fit well with our data.

For our modern methods with default parameters we also get F1 score = 1 but they have an advantage compared to our models, as they work faster.

Thus, neural FCA has shown good results with the right choice of parameters, but it is slower than modern methods, which can be corrected over time with more study of neural FCA.

Code

My code and dataset can be find in my GitHub repository [5].

List of references

1. https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations
2. https://www.tuasaude.com/en/how-to-tell-if-you-have-a-fever/
3. https://en.wikipedia.org/wiki/Human_body_temperature
4. https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2
5. https://github.com/thecrazymage/Neural-FCA

Appendix

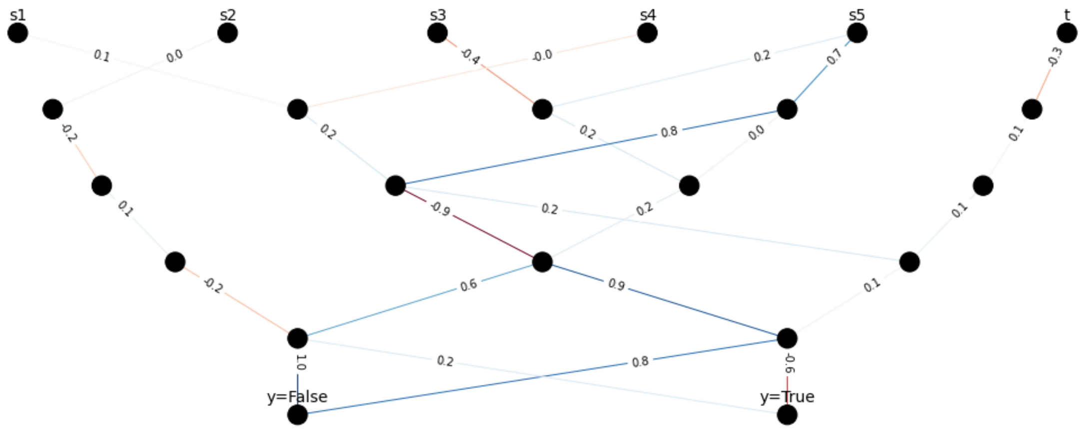

Figure 1: Neural network with fitted edge weights for model 1

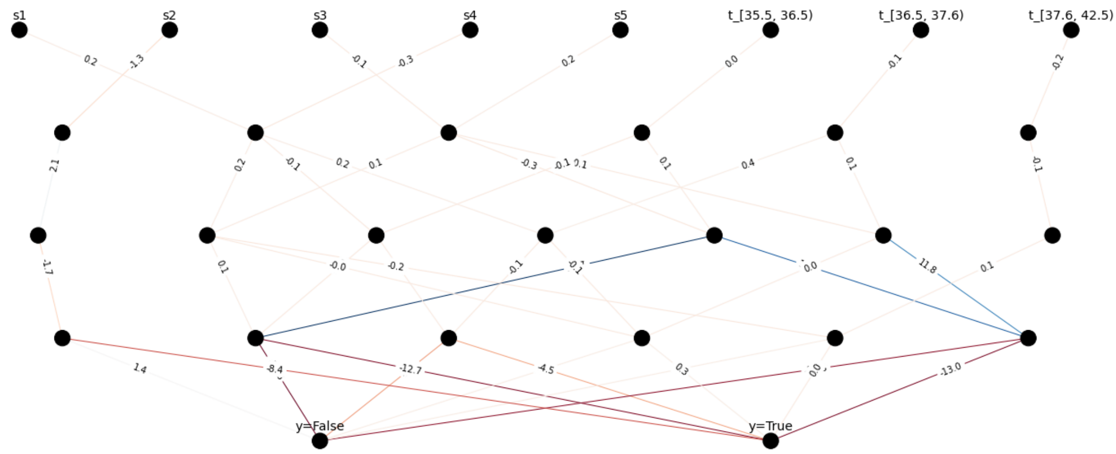

Figure 2: Neural network with fitted edge weights for model 2

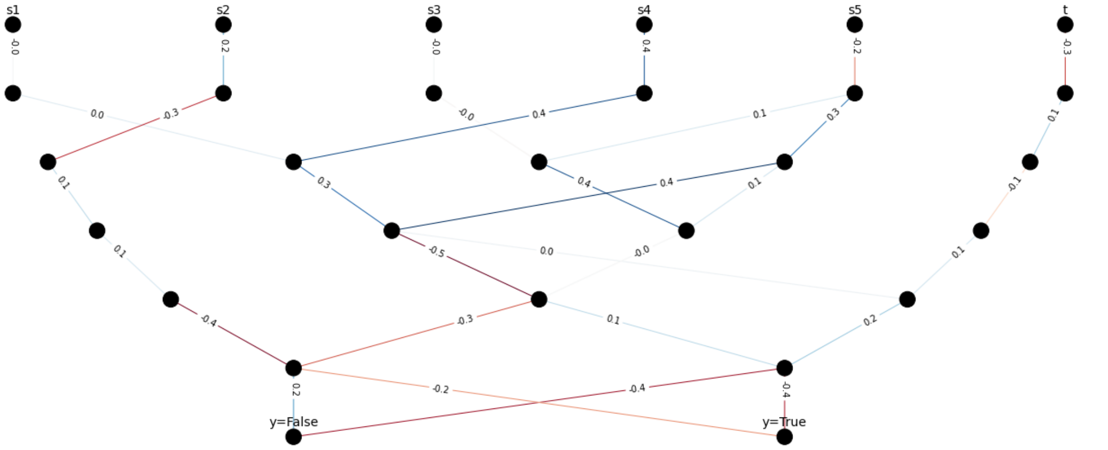

Figure 3: Neural network with fitted edge weights for model 3.1

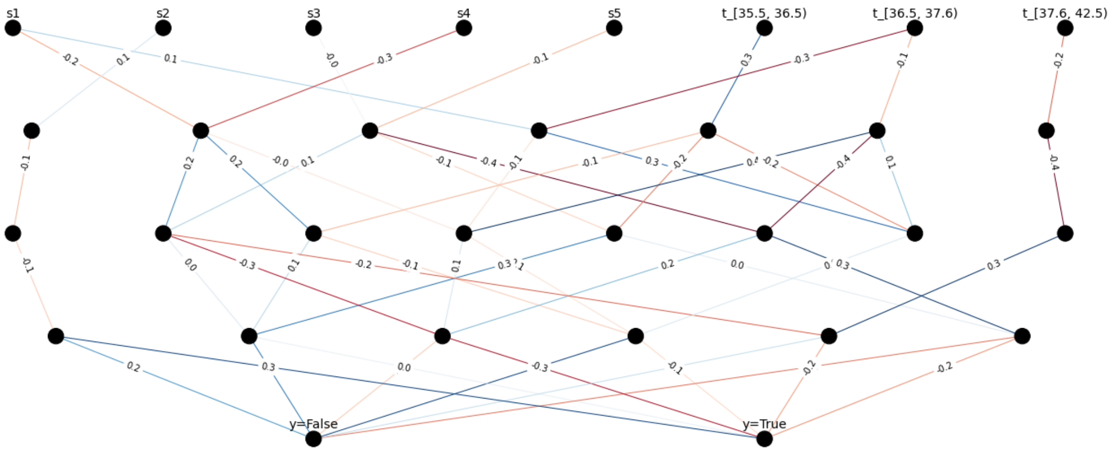

Figure 4: Neural network with fitted edge weights for model 3.2

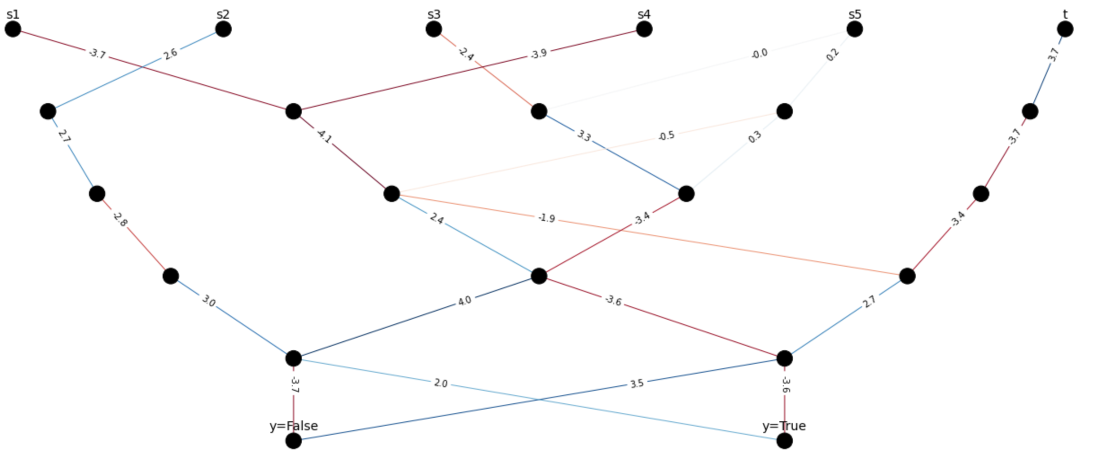

Figure 5: Neural network with fitted edge weights for model 4

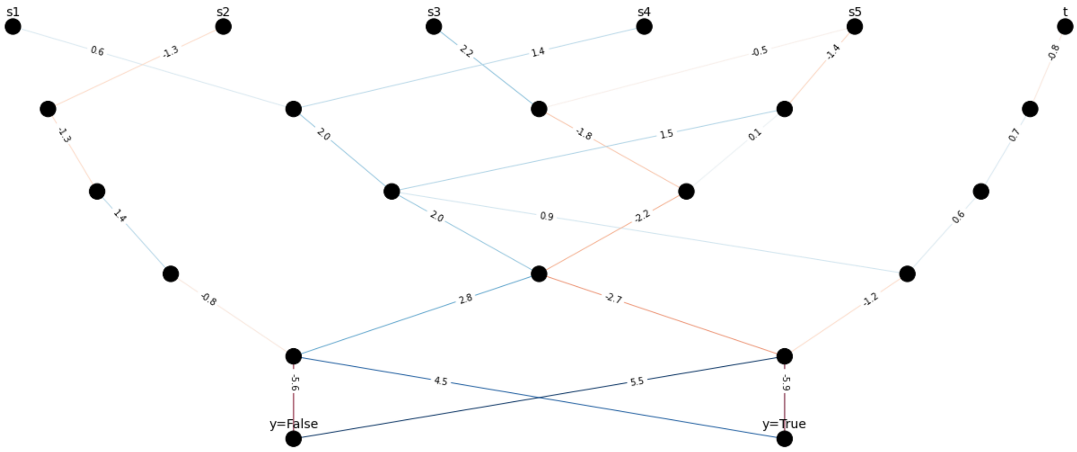

Figure 6: Neural network with fitted edge weights for model 5

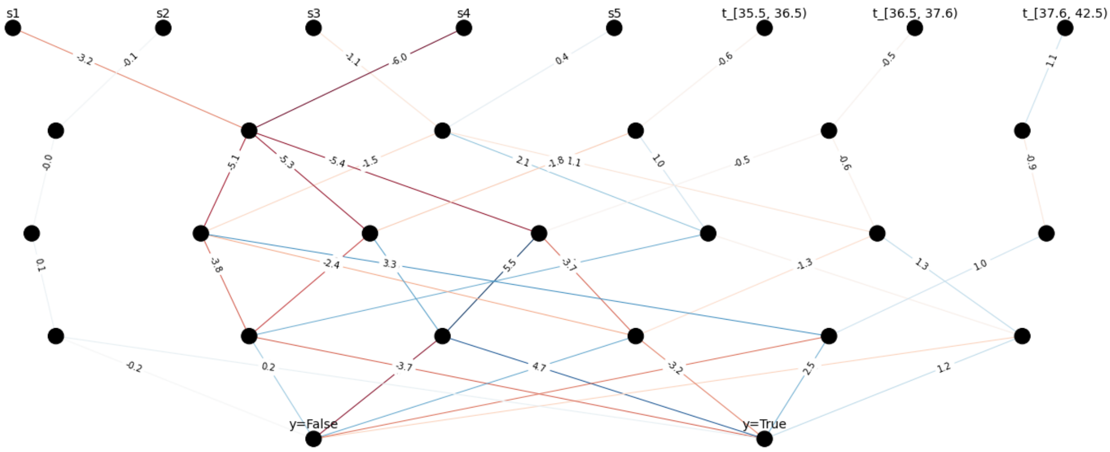

Figure 7: Neural network with fitted edge weights for model 6

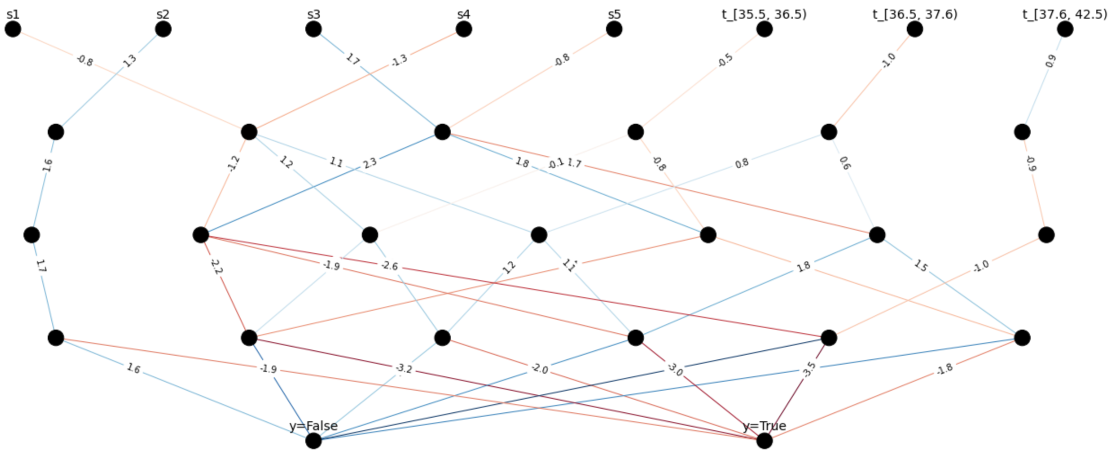

Figure 8: Neural network with fitted edge weights for model 7