FCA Toolbox Project

Made by: Vitaly Sopov

Faculty of Computer Science
Higher School of Economics
2019

1. The choice and preprocessing of dataset.

For this project the UCI's "Heart Disease" dataset was chosen (https://www.kaggle.com/ronitf/heart-disease-uci). The dataset has 303 entries, 165 of which are positive and 138 are negative. It contains 3 binary features, 5 categorical features and 5 numerical features. All numerical features were firstly converted to categorical by clustering them into buckets according to their value ranges. After that all categorical features were converted into binary by means of one-hot encoding. Application of pattern structures was attempted, but theire usage showed substandard results and thus was omitted from this report.

2. Proposed FCA algorithms.

I propose two algorithms that use FCA toolbox.
The classification rule in both cases is:

$$\frac{f_+(g)}{f_-(g)} \geq T$$

That means that for any object **g** if the ratio of positive to negative functions is no less than certain threshold **T**, then the object is classified as positive, otherwise it is classified as negative

1) For object **g** proposed function **f₊** looks like this:

$$f_+(g) = \frac{\sum_{g_+ \in G_+} |g' \cap g_+|}{|G_+|}$$

f₋ looks the same, except that the signs in the formula will be mirrored.

2) For building the second aggregation function I used the following thoughts:
For any class (let it be G+ for example) to be selected for sample g:
1) we want to maximize the number of objects with similar descriptions in this class, and the more similar any description is, the better.
2) at the same time, if for any similar description g+ in G+ we can find some "counterexamples" in G-, then the importance of this similarity between g and g+ is lowered. And we can possibly tolerate a single counterexample or a couple of them to some extent, but the more of them we see, the less important pair (g, g+) is.
3) At the same time, we should probably somehow take into account the difference in size for sets G+ and G-. In particular, for bigger G+, we will naturally find more

"similar" descriptions to our sample g. On the other hand, for any "similar" pair (g,g+) we will probably find more counterexamples in a bigger G-. So, this difference should be compensated.
The second proposed function looks like this:

$$f_+(g) = \sum_{g_+ \in G_+} \frac{2^{|g' \cap g_+| * \frac{a}{|G_+|}}}{2^{h_-(g' \cap g_+) * \frac{b}{|G_-|}}}$$

h₋(x) here denotes the number of negative descriptions which are supersets for given intersection x. **a** and **b** denote (tunable) constant parameters.

3. Experiments and comparisons.

Proposed algorithms are compared with logistic regression and random forest. As both LogReg and RF can naturally work with both binary and numerical features, they are both applied both for binarized and non-binarized data.
For testing purposes, the dataset was split into 5 more or less equal parts, to which then cross-validation was applied. For every metric the average for 5 runs was taken.

	Random Forest (num.)	LogReg (num.)	Random Forest (bin.)	LogReg (bin.)	FCA 1 (t=0.9)	FCA 1 (t=1.05)	FCA 1 (t=1.2)	FCA 2 (t=0.75)	FCA 2 (t=0.9)	FCA 2 (t=1.05)
True Positive Rate	0.38	0.383	0.463	0.457	0.53	0.46	0.373	0.523	0.47	0.417
True Negative Rate	0.597	0.607	0.353	0.353	0.283	0.37	0.413	0.297	0.373	0.393
Negative Predictive Value	0.994	1.0	0.819	0.807	0.958	0.824	0.708	0.94	0.844	0.761
False Discovery Rate	0.054	0.028	0.184	0.184	0.248	0.163	0.105	0.236	0.152	0.135
Accuracy	0.977	0.99	0.817	0.81	0.813	0.83	0.787	0.82	0.843	0.81
Precision	0.946	0.972	0.816	0.816	0.752	0.837	0.895	0.764	0.848	0.865
Recall	0.993	1.0	0.851	0.841	0.974	0.845	0.686	0.962	0.862	0.761

The results show that proposed algorithms are much better suited for binarized features, with the second proposed algorithm showing the best results and the first proposed algorithm being the second best.
At the same time, while being better than RF and LogReg when compared on equal grounds (using binarized features), they are much worse than those algorithms when applied to numerical features.

Speaking of FCA algorithms it is also important to note that they are rather sensitive to choosing thresholds. The values for threshold should be very carefully chosen to prevent big rates of false positives and false negatives.

4. Conclusion.

Overall, the proposed algorithms, while showing generally good results and outperforming conventional algorithms such as Random Forest and Logistic Regression on binarized features, lose to the same algorithms when it is possible to acquire initial numerical features.

Still, they are a good choice for datasets with fewer or less important numerical features or with application of cleverer binarization strategies.