Pattern Structures Application for Text Lazy Classification

Abdulrahim Ghazal, Sergei O. Kuznetsov

Background

- Underground forums are internet platforms where hackers and hacktivists share and announce information about attacks and tools to harm entities.
- Threat Intelligence systems, collects data from these forums and stores it for human analysts.
- The amount of messages is very big and it becomes hard for analysts to monitor threats in real-time approach.
- The aim of the research is to classify messages into risky/non-risky

Data

- The information collected includes the message text, starting from 2021.

- All these details are stored in a database and processed to clean them from spam messages and stop words.

Work Done

- We used The lazy classification method suggested in the paper “Scalable Knowledge Discovery in Complex Data with Pattern Structures”.
- It relies on using the pattern structures extended hypothesis classification approach.
- It can remove the burden of extracting all knowledge from data (implications), and reduce complexity stemming from the storage and processing all nodes of formal concept lattice.
- This method can produce unclassified examples.
- We worked with several experimental settings.

Work Done

Algorithm 1: Lazy Classification with Pattern Structures

Requires: formal context (G, $\underline{D}$, δ), new test example g_t

1: $att\_g_t = \delta(g_t)$

2: for g $\in G$:

3: att_inter $= att\_g_t \sqcap \delta(g_t)$

4: inter-objects $= (att\_inter)^{\diamond}$

5: for obj in inter-objects:

6: if all obj has target attribute, classify positive

7: if all obj does not have target attribute, classify negative

8: classify undetermined (reached the end of algorithm without classification.

Experiments

- Dataset is a table of object-attribute values
- Objects are messages, attributes are binary TF-IDF values
- We have 2 variables to control:
  - the ratio of positive vs. negative examples
  - The mind_df which is the threshold below which the tf-ifd builder ignores the keyword.
- Each run represents a pair of variables (ratio,min_df)
- We will go through 5 values for each of the variables: ratio (1,5) and min_df (0.01, 0.05)
- We define saved effort as the percentage of classified examples (1 - ratio of unclassified examples)

$$att\_value(keyword) = \begin{cases} 1 & keyword \in vectorizor\ vocab \\ 0 & otherwise \end{cases}$$

Experiment	F1	saved effort
1	**98.8**	**88.8**
2	97.3	82.4
3	89.4	81.0
4	95.5	78.7
5	94.5	78.9
6	97.8	83.1
7	96.3	75.2
8	95.1	69.7
9	93.5	65.1
10	92.7	66.0
11	96.6	73.1
12	95.5	65.9
13	93.4	58.3
14	92.7	58.5
15	91.8	53.3
16	95.9	69.3
17	94.8	61.9
18	92.6	48.5
19	91.8	51.8
20	87.8	46.4
21	96.4	68.7
22	94.1	48.4
23	92.4	43.2
24	87.4	43.0
25	75.6	33.7

Experiments

- Dataset is a table of object-attribute values
- Objects are messages, attributes are interval TF-IDF values
- We have 2 variables to control:
  - the ratio of positive vs. negative examples
  - The mind_df which is the threshold below which the tf-ifd builder ignores the keyword.
- Each run represents a pair of variables (ratio,min_df)
- We will go through 5 values for min_df: ratio (1) and min_df (0.01, 0.05)
- We define saved effort as the percentage of classified examples (1 - ratio of unclassified examples)

$$[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max[b_1, b_2)]$$

Exp (ratio, min_df)	F1	saved effort
1 (1,0.01)	**88.0**	**87.7**
2 (1,0.02)	78.4	79.8
3 (1,0.03)	76.1	70.6
4 (1,0.04)	70.4	65.8
5 (1,0.05)	66.5	61.3

Experiments

- Dataset is a table of object-attribute values
- Objects are messages, attributes are min TF-IDF values
- We have 2 variables to control:
    - the ratio of positive vs. negative examples
    - The mind_df which is the threshold below which the tf-ifd builder ignores the keyword.
- Each run represents a pair of variables (ratio,min_df)
- We will go through 5 values for min_df: ratio (1) and min_df (0.01, 0.05)
- We define saved effort as the percentage of classified examples (1 - ratio of unclassified examples)

$$[a_1, \infty] \sqcap [a_2, \infty] = [min(a_1, a_2), \infty]$$

Exp (ratio, min_df)	F1	saved effort
1 (1,0.01)	**89.7**	**87.6**
2 (1,0.02)	69.5	65.3
3 (1,0.03)	75.9	74.9
4 (1,0.04)	54.0	59.7
5 (1,0.05)	65.7	62.2

Experiments

- Dataset is a table of object-attribute values
- Objects are messages, attributes are max TF-IDF values
- We have 2 variables to control:
  - the ratio of positive vs. negative examples
  - The mind_df which is the threshold below which the tf-ifd builder ignores the keyword.
- Each run represents a pair of variables (ratio,min_df)
- We will go through 5 values for min_df: ratio (1) and min_df (0.01, 0.05)
- We define saved effort as the percentage of classified examples (1 - ratio of unclassified examples)

$$[a_1, \infty] \sqcap [a_2, \infty] = [max(a_1, a_2), \infty]$$

Exp (ratio, min_df)	F1	saved effort
1 (1,0.01)	**84.3**	**87.8**
2 (1,0.02)	76.0	81.1
3 (1,0.03)	68.9	72.3
4 (1,0.04)	58.8	67.0
5 (1,0.05)	59.9	64.5

Discussion

- Binary data gave better results as it narrows the conditions for selecting objects to check.
- We can see that F1 measure decreases with the increase of min_df which increases the number of keywords.
- The most restrictive part w.r.t. Values to consider decisive in classifying will be the interval then min then max
- Many small adjustments have been made to the main algorithm to reduce complexity
- The system works in a sensible time given the number of messages and how much time it needs to classify each message.
- In the current version, the explanation is the attribute set of intersection between the test object and one of the samples that resulted in the classification results.

Thank you for attention

Questions ?