Analyzing raw log files to find execution anomalies

Viktor Jovanoski, Jan Rupnik,
Mario Karlovčec, Blaž Fortuna

SiKDD - October 2017

Overview

- Anomaly detection
- Log file analysis
- Algorithm
- Evaluation
- Future work

Anomaly Detection

- Anomaly: **any data point that significantly deviates from the remaining data**
- Goals:
  - Detect anomalous sudden changes (short term)
  - Detect slow degradation (long term)
  - Root-cause analysis
- Applicable to many fields, we monitor specific IT system

Anomaly Detection - challenges

- Diverse data types – numeric, discrete, text, sequences, ...
- Running on stream/batch
- Actionability – users require understandable explanations
- Proper quantity – not too many, not to few
- “Concept drift”– reality changes over time

GENERAL STREAMING ALGORITHM

```
while input data available do
    parse data and extract record
    calculate anomaly score
    if score above threshold then
        report anomaly
    end if
    add record to model
end while
```

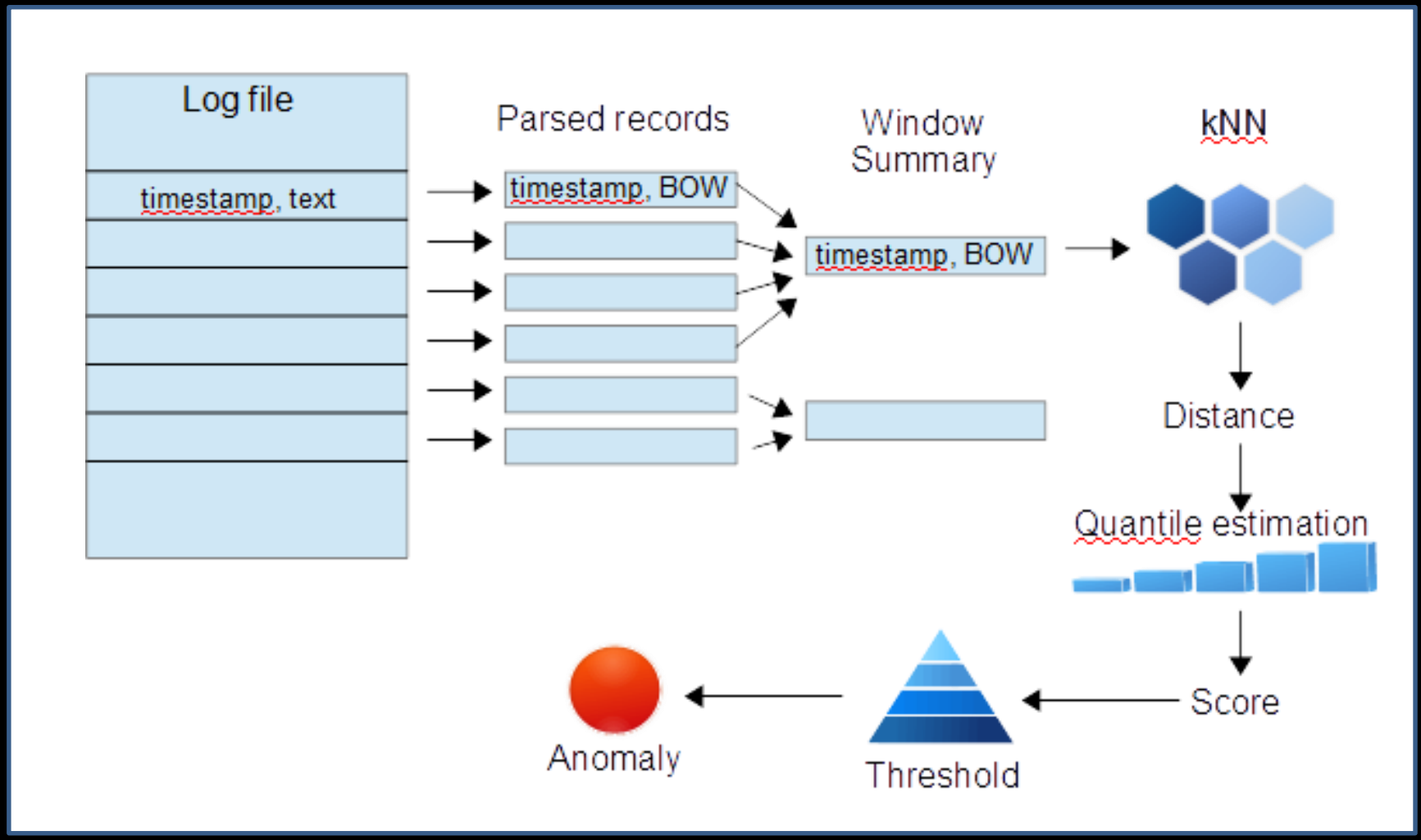
Log file analysis

- Why log files? Sometimes this is all we have.
- Unstructured text, but contain timestamp
- Parse location identifiers, detect errors and warnings
- Identify “execution profiles” – descriptions of time intervals

Algorithm

```
while input data available do
    read data from log file
    parse data and extract BOW
    insert into time-windows
    calculate distance in kNN
    if score quantile above threshold then
        create explanation using distance
        report anomaly
    end if
    add record to model
end while
```

PIPELINE


Log file
timestamp, text
Parsed records
timestamp, BOW
Window Summary
timestamp, BOW
kNN
Distance
Quantile estimation
Score
Threshold
Anomaly

Algorithm parameters

- k - # of nearest neighbors for distance comp.
- time window – the length of summary windows
- history window – how many historical examples do we compare against
- nn_rate – quantile treshold

- Must be hand-tuned to find balance between quantity, statistical quality and actionability

Anomaly explanation

- Must be ACTIONABLE
- Show difference to nearest neighbor
  - Most often, unusual processes stick out
  - Must be interpreted by an expert

Evaluation

- Not many labeled datasets
- Mostly unlabeled data
  - Previously undetected anomalies
- Precision and recall
- Test1: Synthetic data
- Test2: Real web-server logs

Evaluation – synthetic data

- Simulating parallel execution of several processes, with deterministic pattern
  - 8 extra anomalous instances, 2 per week
- 1 year of data – 1 min and 1 h windows, history = 10 days, k = 1, nn_rate = 0.003
- 1 min window – recall=1.0, precision=0.06
- 1 h window – recall=0.66, precision=0.14

Evaluation – web-server logs

- Static web site – not many content changes
- 1 month of data – 5 min windows, history = 10 days, k = 1
- Manual inspection of generated anomalies
- Those with **one strong dimensional outlier** were “correct” – new content has been published

Future work

- Full-spectrum analysis
  - Many different data sources
  - Correlation across data sources
  - Root-cause analysis
  - Prediction
  - Active learning
- Deep learning