Assignment # 2 — Log Analysis for Anomaly Detection

Deadline: Nov 24 23:59 Grade: 20%

Background and Data

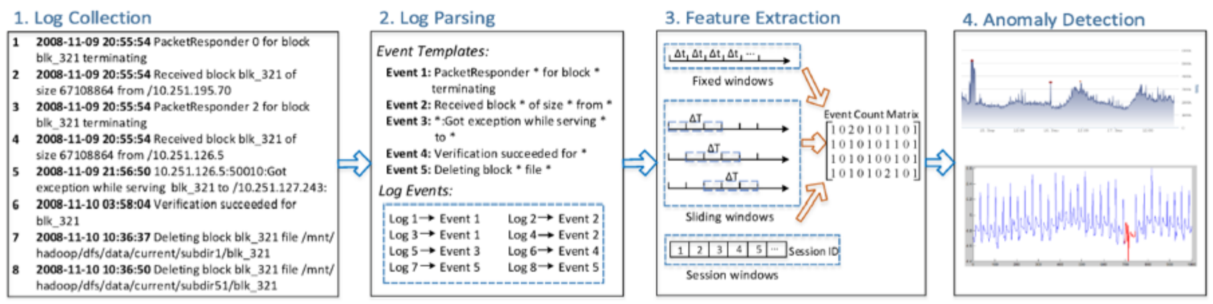
Anomaly detection plays an important role in management of modern large-scale distributed systems. Logs are widely used for anomaly detection, recording system runtime information and errors. Traditionally, operators have to go through the logs manually with keyword searching and rule matching. The increasing scale and complexity of modern systems, however, makes the volume of logs explode, which renders the infeasibility of manual inspection. To reduce manual effort, we need anomaly detection methods based on automated log analysis.

Raw log messages are usually unstructured texts. To enable automated mining of unstructured logs, the first step is to perform log parsing, whereby unstructured raw log messages can be transformed into a sequence of structured events. Then we are able to do anomaly detection based on these sequences.

Typically, one log message will record a specific system event with a set of fields: timestamp, verbosity level (e.g., INFO, WARN, ERROR), and free-format message content. A sample of the raw log file is shown below:

```
2008-11-09 20:35:32,146 INFO dfs.DataNode$DataXceiver: Receiving block
blk_-1608999687919862906 src: /10.251.31.5:42506 dest: /10.251.31.5:50010
```

The figure below shows a popular framework for log-based anomaly detection. The anomaly detection framework mainly involves four steps: log collection, log parsing, feature extraction, and anomaly detection.



```
1   2008-11-09 20:55:54 PacketResponder 0 for block
    blk_321 terminating
2   2008-11-09 20:55:54 Received block blk_321 of
    size 67108864 from /10.251.195.70
3   2008-11-09 20:55:54 PacketResponder 2 for block
    blk_321 terminating
4   2008-11-09 20:55:54 Received block blk_321 of
    size 67108864 from /10.251.126.5
5   2008-11-09 21:56:50 10.251.126.5:50010:Got
    exception while serving blk_321 to /10.251.127.243:
6   2008-11-10 03:58:04 Verification succeeded for
    blk_321
7   2008-11-10 10:36:37 Deleting block blk_321 file /mnt/
    hadoop/dfs/data/current/subdir1/blk_321
8   2008-11-10 10:36:50 Deleting block blk_321 file /mnt/
    hadoop/dfs/data/current/subdir51/blk_321
```

Event Templates:

Event 1: PacketResponder * for block * terminating
Event 2: Received block * of size * from *
Event 3: *:Got exception while serving * to *
Event 4: Verification succeeded for *
Event 5: Deleting block * file *

Log Events:

```
Log 1→ Event 1    Log 2→ Event 2
Log 3→ Event 1    Log 4→ Event 2
Log 5→ Event 3    Log 6→ Event 4
Log 7→ Event 5    Log 8→ Event 5
```

This assignment is divided into two parts:

1) The first part is **Log Parsing** (second step in above figure).

The purpose of log parsing is to extract a group of event templates; whereby raw logs can be structured;

2) The second part involves **Feature Extraction and Anomaly Detection** (third and fourth steps in above figure).

In this part, we will identify whether or not a new incoming log sequence is an anomaly.

Part 1:

[1] introduces a number of data-driven approaches for automated log parsing. In this part, we offer some datasets (https://github.com/logpai/logparser/tree/dev/data; include five system logs: BGL, HDFS, HPC, Proxifier, Zookeeper, each file has 2000 labeled logs). In each dataset, “rawlog.log” is the raw log to parse. “template*XX*.txt” refer to logs belong to *XX* template.

You need to learn the paper [1] and do the following jobs using the given datasets:

- Grasp **four** log parsing algorithms: **LogSig**, **IPLom**, **SLCT** and **LKE**
- Use toolkit to run the four log parsing algorithms (toolkit: https://github.com/logpai/logparser/tree/dev)
- Plot the **runtime** with a bar chart when four algorithms parse these datasest.
- Try to adjust parameters for best **RandIndex**[2] (a metrics for evaluation clustering) and plot a bar chart when four algorithms parse these datasets.
- Try to adjust parameters for best **F-score** and plot a bar chart when four algorithms parse these datasets.
- Describe your own experience or findings in doing those jobs. For example, advantages and disadvantages of these algorithms.

Part 2:

[3] introduces a number of data-driven approaches for automated anomaly detection based on system log. In this part, we provide HDFS logs to do anomaly detection follow the framework mentioned above.

You need to learn the paper [1,3] and do the following jobs using the **HDFS logs**.

- Grasp the principles of **two** unsupervised anomaly detection models: **Invariants Mining, PCA**
- Based on part1, choose one log parsing methods, and then use toolkit to run the three anomaly detection models (toolkit: https://github.com/logpai/loglizer).
- Try to adjust parameters for better **F-score, precision, recall,runtime** and plot them with bar charts when parsing logs.
- When running **Invariants Mining**, add some codes and display **three relationships** (please check the paper for more information), *e.g.*, n(A) = n(B), where n(∗) represents the number of logs which belong to corresponding template ∗. And explain why, *e.g.*, template A is “Interface *, changed state to down”, while template B is “Interface *, changed state to up”.
- Describe your own experience or findings in doing those jobs. For example, in which situation, these models do not detect anomalies correctly.

data

https://www.dropbox.com/s/akef557hnla0h9v/ANM-data.zip?dl=0

https://cloud.tsinghua.edu.cn/f/c8806b4c81ee45afa03c/?dl=1

submit

A zip file , which includes template files in part#1 and an assignment report.

Reference

[1] He P, Zhu J, He S, et al. An Evaluation Study on Log Parsing and Its Use in Log Mining[C]// Ieee/ifip International Conference on Dependable Systems and Networks. IEEE Computer Society, 2016:654-661.

[2] https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

[3] He S, Zhu J, He P, et al. Experience Report: System Log Analysis for Anomaly Detection[C]// IEEE, ISSRE 2016.