80240663 Advanced Network Management
Assignment 2

Tsinghua University

November 12, 2019

Albert Millan – 2019280366

Contents

1 Log Analysis for Anomaly Detection

This paper aims to provide an analysis of two weeks‘ search engine log data and highlight the key findings. A brief summary of the data is outlined first. The following structure is identical to the one described in the assignment description paper, providing both the required graphs and an explanation at each section. Concluding remarks are presented lastly, summarizing the results and primary insights of the project.

2 Part I

The key issue was to compute the ***Randindex*** cluster evaluation metric. The complication stem from the fact that the parsing algorithms provided assign a label to each event that differs to those assigned in the dataset containing the groundtruth (e.g. ‘E40’). This is paramount to compute the true negative (TN) and positive (TP) values, that are used in the randindex formula. The proposed solution involved generating a table, mapping the clusters from the parsed method and expressing them in terms of groundtruth clusters. A sample for the first twenty initial rows from the Zookeper dataset parsed and mapped in such manner is presented in figure 1.

	663a6d5a	495b290d	7a2ec7d0	45a3ec5d	273e53f2	cb66604c
E42	1.0	0.0	4.0	0.0	0.0	0.0
E11	0.0	4.0	0.0	0.0	0.0	0.0
E24	0.0	0.0	0.0	4.0	0.0	0.0
E40	4.0	0.0	0.0	0.0	0.0	0.0
E25	0.0	0.0	0.0	0.0	2.0	0.0
E31	0.0	0.0	0.0	0.0	0.0	1.0

Figure 1: Columns represent the clusters generated by the parsing algorithm. Rows show the mapping into the groundtruth clusters. In this case, the parsed algorithm classified in group ‘663a6d5a’ elements from two distinct clusters according to the groundtruth, namely ‘E42’ and ‘E40’.

The solution was decomposed in two, computing the TP and TN independently. The value for TN is computed by adding the result of the multiplication between each non-zero element in each column (cluster) by each of the elements on the following columns that are not in the same row as

themselves. The value for TP is computed by counting the number of combination of pairs belonging to the same cluster in the groundtruth dataset. Knowing this two variables, the randindex can then be computed using the formula provided in equation 1.

$$RI = \frac{TP + TN}{TP + FP + TN + FP} \tag{1}$$

2.1 Data Analysis

The chart presented in figure 2 presents the runtime of the four algorithms on each dataset. The execution time of the **IPLoM** and **SLCT** parsing algorithms is extremely low compared to the one yielded by **LogSig** and subsequently **LKE**. In terms of clustering accuracy, the results involving the randindex shown in figure 3 indicate that IPLoM efficiently clusters logs having related events together, with little error as denoted by the large value associated with its f1 score, outlined in figure 4. Considering this reasons, it is arguably the algorithm that yields a superior performance across both runtime and clustering accuracy, hence the one that was chosen to parse the HDFS file in section 3 of this project. Further information on the advantages and disadvantages is provided below, as well as covering the process employed to maximize the f1 score and randindex.

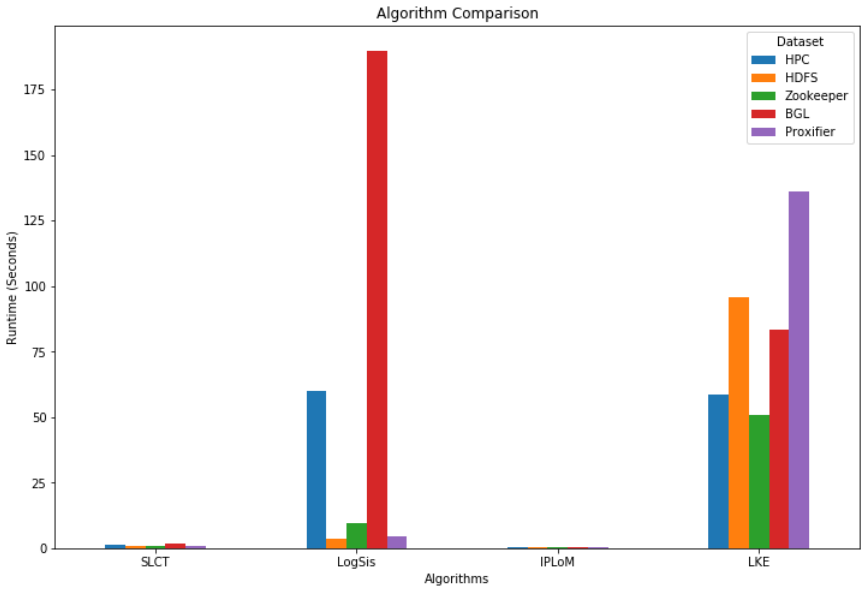

Figure 2: Runtime Comparison of the IPLoM, LogSig, LKE and algorithms.

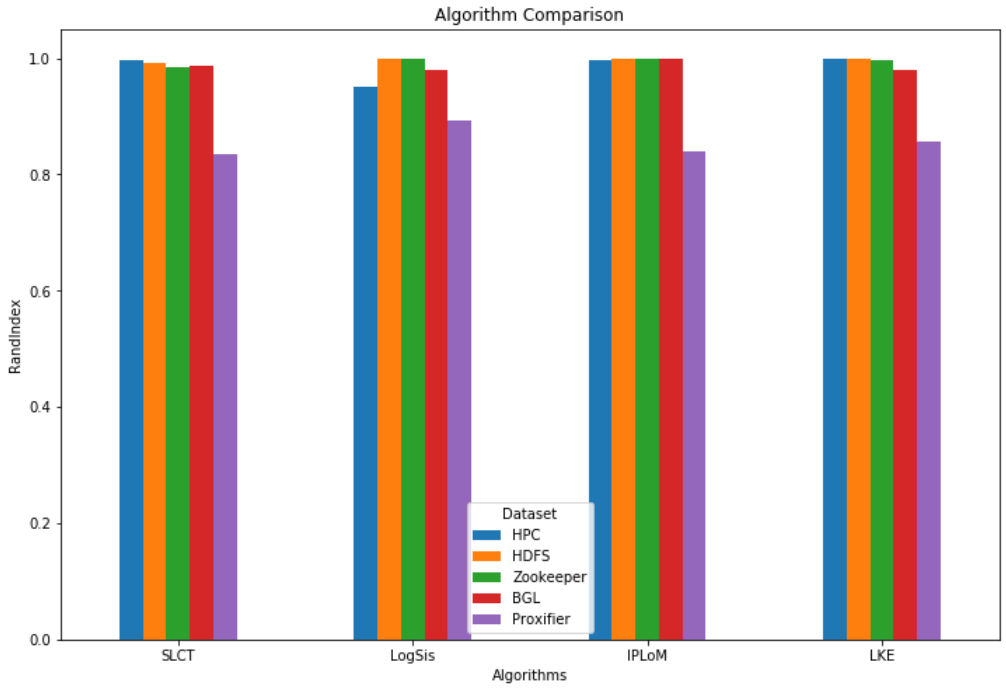

Figure 3: RandIndex comparison of the IPLoM, LogSig, LKE and SLCT algorithms.

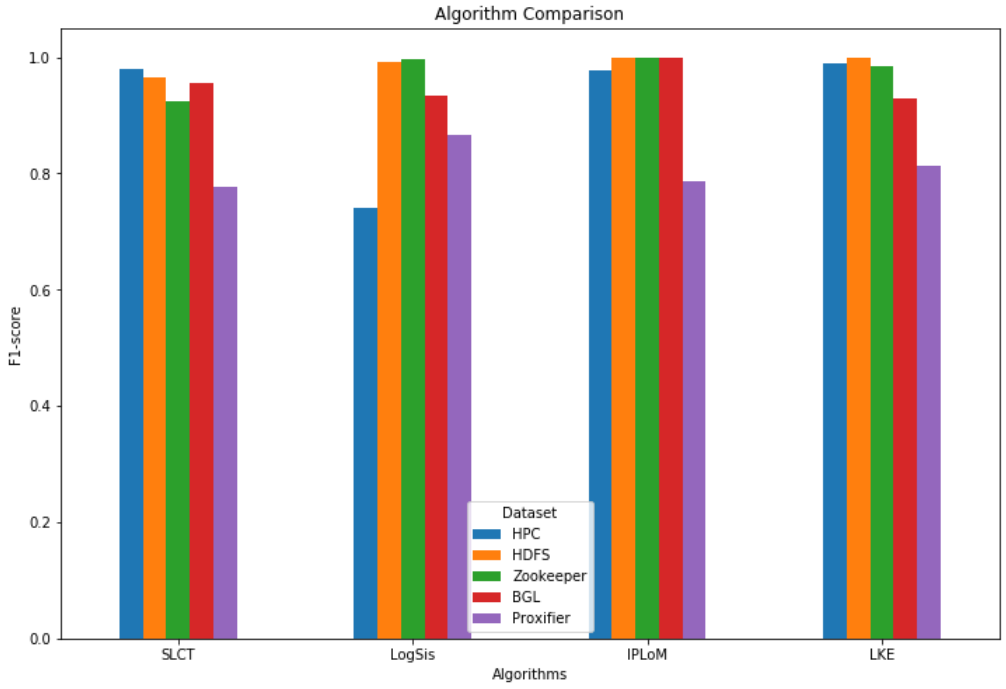

Figure 4: F1-score comparison of the IPLoM, LogSig, LKE and SLCT algorithms.

A summary of the key advantages and disadvantages is provided in table 1. It aims to compare the algorithms in terms of execution time, f1-score and randindex (cluster similarity) and determine which one performs best. It evidences that IPLoM outperforms the others in the three categories under which the algorithms are being evaluated, followed closely by SLCT, and subsequently LogSis and LKE.

	Runtime	F1-score	RandIndex
IPLoM			
SLCT			
LogSis			
LKE			

Table 1: Comparison of the evaluation metrics employed to determine the strengths and weaknesses of the IPLoM, LogSig, LKE and SLCT algorithms.

2.1.1 Maximizing RandIndex & F1-score

The process employed to maximize the values is grid-search. Each algorithm provides its own set of variables that can be modified, and each dataset behaves differently to the values of these variables. A sample test optimizing the values of the parameter employed in the LogSis algorithm is shown in 5. The exact same method was also used to compute the optimal parameter values for the randIndex.

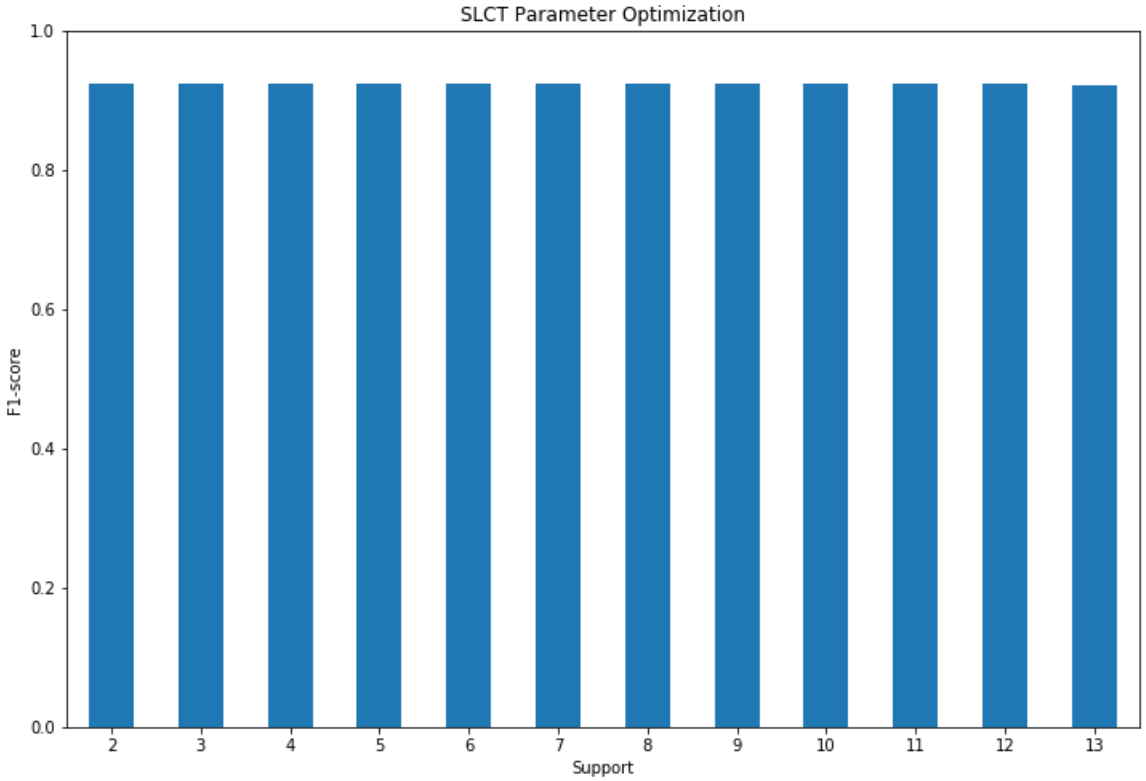

Figure 5: F1-score output given the value of the parameter 'Support' in the SLCT algorithm and Zookeeper dataset.

3 Part II

This section explore anomaly detection with three distinct algorithms, namely ***Invariants Mining, PCA, & Log Clustering***. The algorithms where applied on a HDFS dataset, which was initially trimmed into five chunks and parsed accordingly using IPLoM parsing algorithm. The files where then concatenated together to get the structured file. The three algorithms were in turn executed in this dataset, and the output was subsequently generated.

3.1 Invariants Mining

The following are three relationships for invariant mining. The terms are defined in terms of the template:

1. A := 'Receiving block src / dest /'

2. B := 'PacketResponder $< * >$ for block $< * >$'

3. C := 'Received block of size $< * >$ from /'

4. D := 'BLOCK* NameSystem.addStoredBlock blockMap updated is added to size $< * >$

Hence, the three relationships across devised include:

$$n(A) = n(B) \tag{2}$$

$$n(B) = n(C) \tag{3}$$

$$n(C) = n(D) \tag{4}$$

3.2 Data Analysis

The following graphs compare the precision (figure 6), recall (figure 7), F-score (figure 8) and runtime (figure 9). The LogClustering and PCA models exhibit a higher precision than the Invariants Mining (IM) algorithm, however, the recall of the former two is lower than the latter. A better measure to analyze the performance of the algorithms is in this case the f1-score, which takes into account both the precision and the recall. This graph indicates that the performance of the PCA and IM is similar, above that yielded by the LogCluster method. In terms of runtime, the IM algorithm takes significantly larger proportion of time than the PCA and LogClustering methods. Notice that the LogClustering was only computed with a sample employing a fifth of the data from the parsed dataset. Taking this three metrics into consideration, overall, it is concluded that the method that yielded the best results in a time-efficient manner is the PCA.

3.3 Findings

A key metric that considerably enhanced the three models' performance was the train to test ratio of the dataset given. Initially, it was set to 50/50%, yielding accurate results when the number of entries was low ($\approx$100000). Notwithstanding, the models experienced a drastic drop in the recall and f-score metric when larger datasets where used. Tuning the ratio to 75/25% was paramount to enhance the anomaly predictive accuracy.

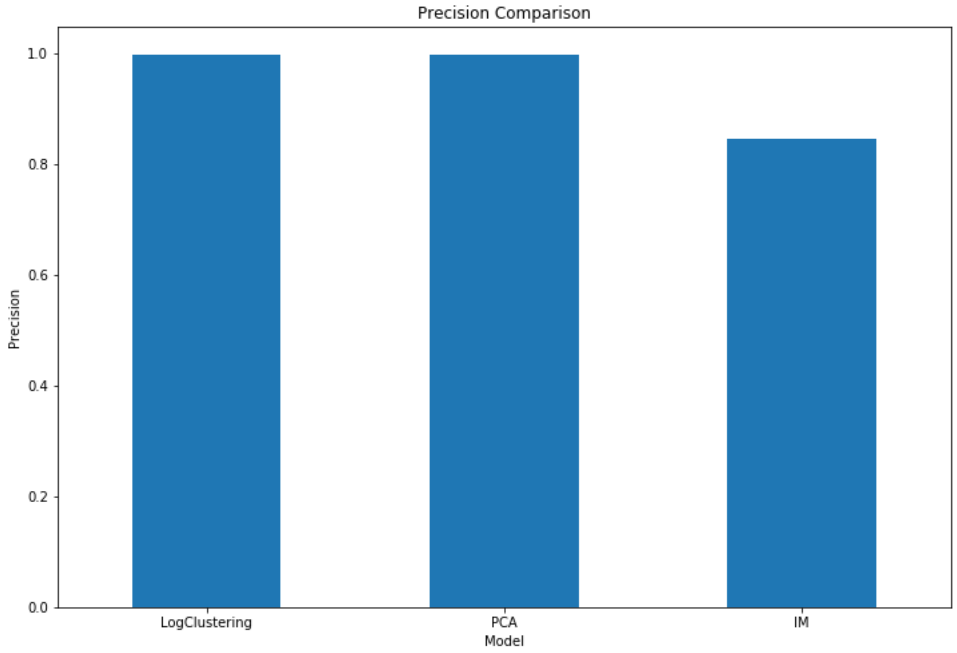

Figure 6: Precision comparison of the anomaly detection models Invariants Mining, PCA and LogCluster.

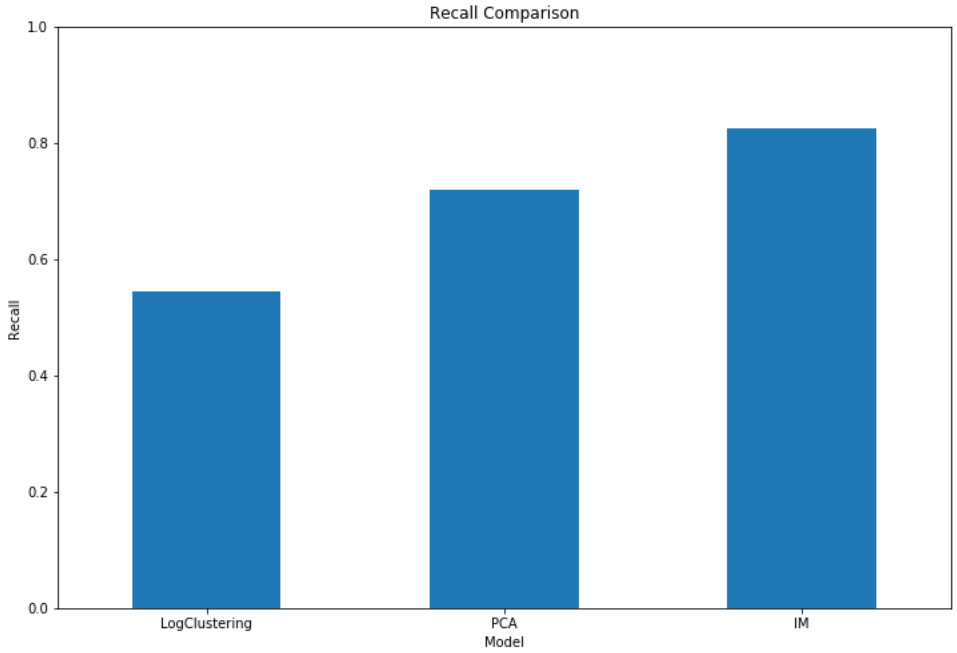

Figure 7: Recall comparison of the anomaly detection models Invariants Mining, PCA and LogCluster.

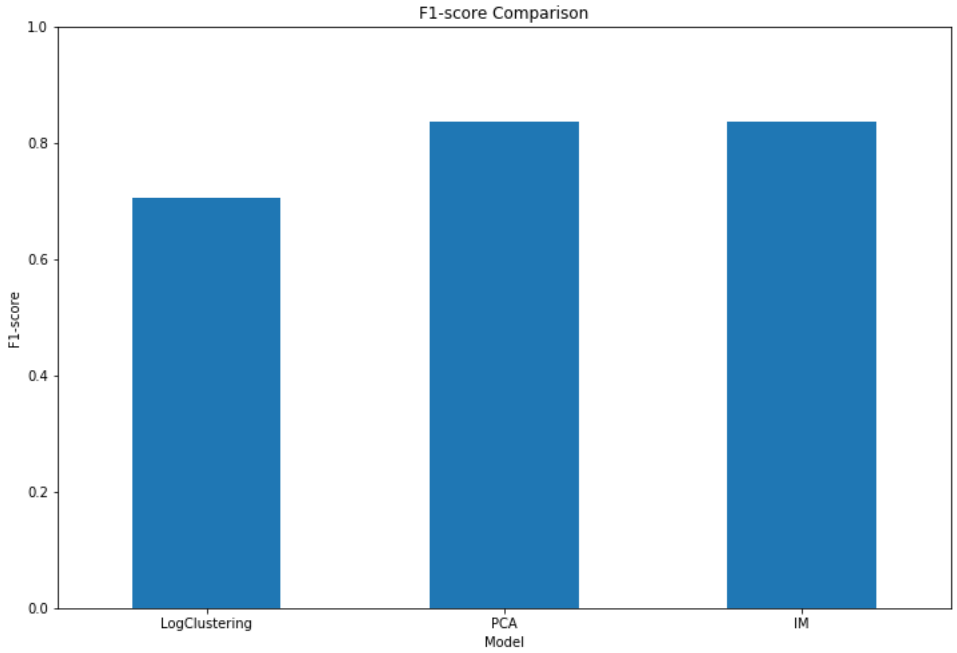

Figure 8: F1-score comparison of the anomaly detection models Invariants Mining, PCA and LogCluster.

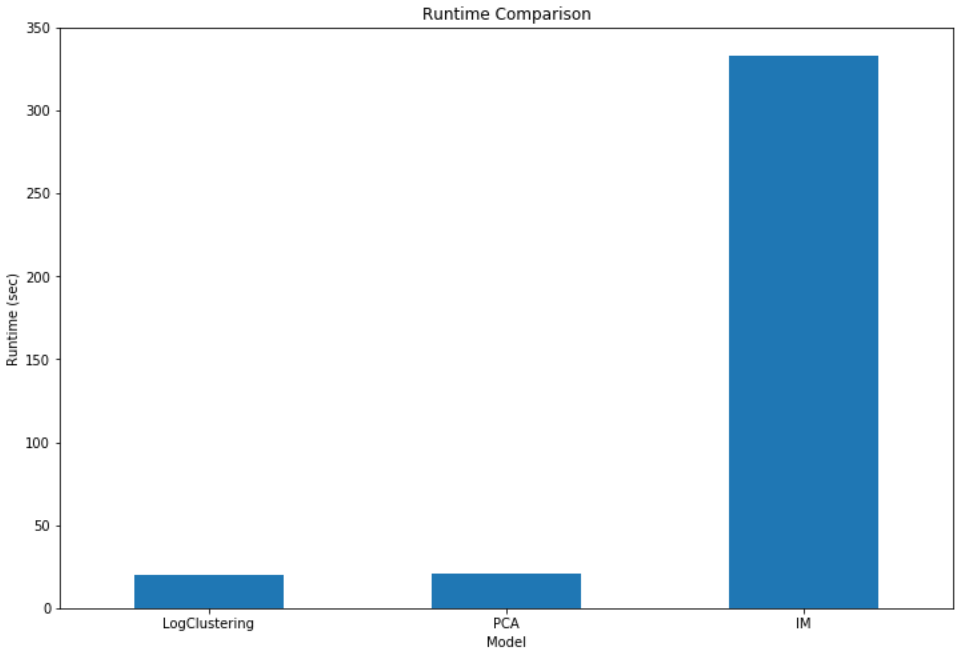

Figure 9: Runtime comparison of the anomaly detection models Invariants Mining, PCA and LogCluster.