OPENSTACK LOG DATA ANALYSIS

• RAVI CHARAN
PROBLEM STATEMENT

- Massive amount of data is generated by the Openstack cloud services in the format of service logs.

- The logs contain information that is useful for pattern analysis.

- Unfortunately this information is generally exposed in semi-structured text format, not allowing direct analysis without additional munging of the data.
TRADITIONAL APPROACH

- Rule Based: Usage of Regular Expressions to extract common patterns.

- This approach requires a pre-knowledge of all text patterns and are not scalable with the services growth.

- Neural Nets: Computationally complex since the entries are approximately $1.5 \times 10^6$/per hour.
Majorly finds its application in pattern analysis by tech giants, analyzing RNA samples, and near duplicate/similar detection in documents.
THE ALGORITHMIC APPROACH

Document → Shingling → Minhashing → Locality-sensitive Hashing

Candidate pairs: those pairs of signatures that we need to test for similarity.

Signatures: short integer vectors that represent the sets, and reflect their similarity.

The set of strings of length $k$ that appear in the document.

general hashing

locality-sensitive hashing
RawData -> Clustering -> Classification -> Anomaly Detection
EXAMPLES
CONCLUSION

- Extend the clustered data for *anomaly detection*.
- Usage of genetic algorithms to generate regular expression from every cluster and mark them as the principal component.
- Jupyter notebooks: https://github.com/RavicharanN/Rally-Log-data-Analysis
- Project report: https://github.com/RavicharanN/Log-Data-Analysis-Report
THANK YOU!

Supervised by:
- DOMENICO GIORDANO
- JOSE CASTRO LEON
- SPYRIDON TRIGAZI
Jaccard index is the metric used to compare the similarity between two messages. Trivially, its given by:

\[ J(A, B) = \frac{|A \cap B|}{|A \cup B|} \]

- Message1 ={"AA BB CC"}   Message2 ={"CC DD EE"}
  - Similarity = 1/5
• **Shingling:** This is the technique that is used to preserve the order of the tokens in the messages as Jaccard similarity doesn't take into account the order of the tokens. We use a 2-shingle approach, which is, we consider 2 consecutive tokens as one word.

• **Minhash:** We generate small integers signatures for the set of shingles which reflect their similarities. These signatures are small enough to fit in the memory and this reduce the face complexity.
• **Locality Sensitive Hashing**: The general idea of LSH is to find a algorithm such that if we input signatures of 2 documents, it tells us that those 2 messages form a candidate pair. It is a method in which hash data points into buckets so that data points near each other are located in the same buckets with high probability.
Since the logs are always generated by the machine, the number of tokens (words) for all the messages in a given class are same.
SAMPLE SPACE REDUCTION

- The first step in the top-down approach of the clustering algorithm is the dividing the dataset on the basis of the number of tokens.

- Run Minhash-LSH on each of the reduced clusters to further divide them into smaller clusters.
VIOLIN SEABORN DISTRIBUTION