Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 10, Issue, 02, pp. 65478-65480, February, 2018

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

LOG CLUSTERING BASED PROBLEM IDENTIFICATION & ONLINE SYSTEMS

***Neha Patil, Dhanashri Yadav, Vishakha Pagare and Karuna Tikare**

Gokhale Education Society's, RH Sapat College of Engineering Management, Maharashtra, India

ARTICLE INFO

Article History:
Received 29th November, 2017
Received in revised form
22nd December, 2017
Accepted 17th January, 2018
Published online 28th February, 2018

Key words:

Logs, Problem Identification, Log Clustering, Diagnosis, Database.

ABSTRACT

Every systemneed to maintain record of events of everything that happen.We should call it Logs. Thislogs are useful in auditing the system & troubleshooting problems. Most of the system & platforms uses Linux based OS to provide their services because of higher availability & open source. The Linux based syslog protocol generates & collect the syslog messages & writes them in text files with their priority, facility & security. This text files (log files) are in normal text format. We convert them to relational database format with sequence of date & time for our convenience to find or search Specific meta data.

Citation: Neha Patil, Dhanashri Yadav, Vishakha Pagare and Karuna Tikare, 2018. "Log Clustering Based Problem Identification & Online Systems", *International Journal of Current Research*, 10, (02), 65478-65480.

INTRODUCTION

This Paper is made to describe it, we propose LogCluster, a log clustering based problem identification approach that considers all the characteristics of the logs of Web Services. In our work, we assign weights to log messages and group the similar log sequences into clusters. We then extract a representative log sequence from each cluster. The operation of LogCluster can be divided into two phases: construction phase and production phase. In the construction phase, we use the log sequences collected from the testing environment, cluster them to construct relational database. In the production phase, we analyze the log sequences collected from the actual production environment, cluster them and check if the clusters can be found in the knowledge base. In this way, developer only need to examine a small number of representative log sequences from the clusters that are previously unseen. Therefore, LogCluster further reduces the total number of logs need to be manually examined and improves the effectiveness of problem identification.We have evaluated our results on database.The results show that the proposed method can effectively help engineers identify problems of service systems.

Clustering Means

- Logcluster is a simple clustering implementation that is targeted specifically to the task of grouping errors in log files.
- Online service systems generate a huge number of logs every day.
- When an online system fails, engineers need to
- Examine recorded logs to gain insights into the failure and identify the potential problems.
- It is challenging for engineers to identify a service problem by manually examining the logs.

Introduction to K-means clustering

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups or database). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

The results of the K-means clustering algorithm are:

1. The centroids of the K clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster)

Rather than defining groups before looking at the data, clustering allows you to find and analyze the groups that have formed organically. The "Choosing K" section below describes how the number of groups can be determined.

Corresponding author: **Neha Patil,**
Gokhale Education Society's, RH Sapat College of Engineering Management, Maharashtra, India.

Each centroid of a cluster is a collection of feature values which define the resulting groups. Examining the centroid feature weights can be used to qualitatively interpret what kind of group each cluster represents.

Project Uses

The **K**-means clustering algorithm is used to find groups which have not been explicitly labeled in the data. This can be used to confirm business assumptions about what types of groups exist or to identify unknown groups in complex data sets. Once the algorithm has been run and the groups are defined, any new data can be easily assigned to the correct group.

This is a versatile algorithm that can be used for any type of grouping. Some examples of use cases are:

- Behavioral segmentation:
  - Segment by purchase history (row data)
  - Segment by activities on application, website, or platform
  - Define person as based on interests
  - Create profiles based on activity monitoring
- Sorting sensor measurements:
  - Detect activity types in motion sensors
  - Group images
  - Separate audio
  - Identify groups in health monitoring
- Detecting bots or anomalies:
  - Separate valid activity groups from bots
  - Group valid activity to clean up outlier detection

In addition, monitoring if a tracked data point switches between groups over time can be used to detect meaningful changes in the data.

features for each data point. The algorithms starts with initial estimates for the **K** centroids, which can either be randomly generated or randomly selected from the data set.

The algorithm then iterates between two steps:

1. Data assignment step:

Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance. More formally, if $\mathbf{c_i}$ is the collection of centroids in set **C**, then each data point **x** is assigned to a cluster based on

$$\underset{c_i \in C}{\operatorname{argmin}}\ dist(c_i,\ x)^2$$

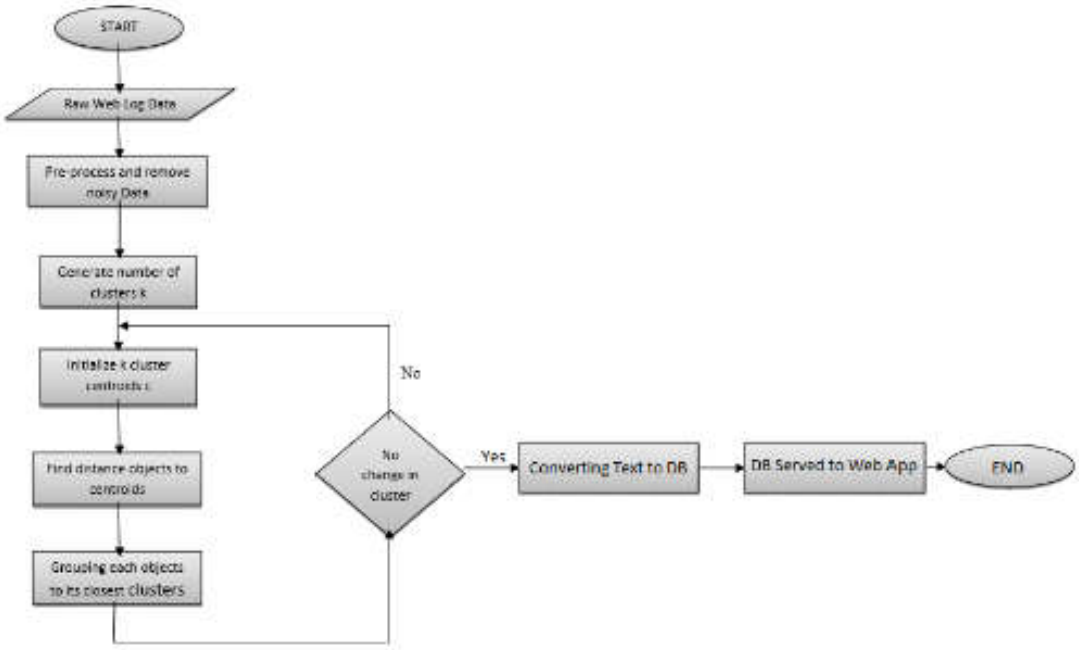
where **dist**(·) is the standard ($\mathbf{L}_2$) Euclidean distance. Let the set of data point assignments for each $\mathbf{i}^{th}$ cluster centroid be $\mathbf{S_i}$.

2. Centroid update step:

In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|}\sum_{x_i \in S_i} x_i$$

The algorithm iterates between steps one and two until a stopping criteria is met (i.e., no data points change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached). This algorithm is guaranteed to converge to a result. The result may be a local optimum (i.e. not necessarily the best possible outcome), meaning that assessing more than one run of the algorithm with randomized starting centroids may give a better outcome.



Algorithm

The **K**-means clustering algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters **K** and the data set. The data set is a collection of

Why Choosing K

- The algorithm described above finds the clusters and data set labels for a particular pre-chosen **K**. To find the number of clusters in the data, the user needs to run the **K**-means clustering algorithm for a range of **K** values and compare the results. In general, there is no method for determining exact value of **K**, but an accurate estimate can be obtained using the following techniques.
- One of the metrics that is commonly used to compare results across different values of **K** is the mean distance between data points and their cluster centroid. Since increasing the number of clusters will always reduce the distance to data points, increasing **K** will always decrease this metric, to the extreme of reaching zero when **K** is the same as the number of data points. Thus, this metric
- Cannot be used as the sole target. Instead, mean distance to the centroid as a function of **K** is plotted and the "elbow point," where the rate of decrease sharply shifts, can be used to roughly determine **K**.
- A number of other techniques exist for validating **K**, including cross-validation, information criteria, the information theoretic jump method, the silhouette method, and the G-means algorithm. In addition, monitoring the distribution of data points across groups provides insight into how the algorithm is splitting the data for each **K**.

Working

The self-made Command Line (Shell Scripting) application will ready to get row data to process on data to analyze & remove impurities like repetitions of entries & application uses *K* means algorithm to form the clusters. And writes them in form of relational database. This database should access on any operating system because of it's modular design. That output is launched on web based application. Which can easily access by any version of web browser.

Conclusion

In this project of Log clustering is done backup of large sizeof log data in cluster based backup. Preprocessed web log data is clustered using improved K-Means clustering algorithm to identify internet users behavior & finally got out put on web page to easily access.

REFERENCES

Active Monitoring, available at: https://en.wikipedia.org/ wiki/Synthetic_monitoring.

Ford Lumban Gaol, Exploring The Pattern of Habits of Users Using WebLog Squential Pattern, IEEE, Second *International Conference on Advances in Computing, Control, and Telecommunication Technologies*, 2010, pp 161-163.

Han, J. and Kamber, M. 2000. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers.

Hybrid model of rule based and clustering analysis for big data security

Masseglia, F., P. Poncelet, and M. Teisseire. 1999. Using data mining techniques on web access logs to dynamically improve hypertext structure. *In ACM SigWeb Letters*, 8(3): 13-19.

Renáta Iváncsy and István Vajk, 2006. Frequent Pattern Mining in Web Log Data, *Acta Polytechnica Hungarica*, Vol. 3, No. 1, pp 77 – 90.

Shang, W., M. Nagappan, A. E. Hassan, Z. M. Jiang, 2014. Understanding Log Lines Using Development Knowledge. In Proc. IEEE *International Conference on Software Maintenance and Evolution*, (ICSME 2014), pp.2130.

****