CDR based Customer profiling using Big Data

Rajesh Kumar[1, #], Bhavneesh Tyagi[2]

[1]*Architect – Ericsson R&D, Gurgaon, India*

[2]*R&D Manager – Ericsson R&D Gurgaon, India*

[1,#]Corresponding Author, Email: choudhary.rajesh123@gmail.com

***Abstract –* CDR (Call Detail Record) is basically the data which is collected during the call. It contains various information like start time, finish time, service type (data or voice), requested URLs, source and destination telephone number or MSISDN. It also gives each event details that occur in the network against a subscriber. Call Detail Record (CDR) is most crucial source of subscriber data which is used in various telecom processes like charging, settlement, billing, network efficiency. This paper will present the use of Hadoop big data technology stack along with Proclus map reduce algorithm to perform the analysis of huge CDR data.**

***Keywords*— CDR, Hadoop, Proclus, Map-Reduce, Distributed.**

1. INTRODUCTION

This paper is proposing a data-analytical process using big data technology for automatically identifying the repeated patterns of subscriber in the telecom industry of mobile network users. This will result in drastically reducing the timing to analyse the patterns of network user, so that better offers and services can be provided to the targeted customers. It will benefit both service provider i.e. network operator and customer as better plans, offer and services can be offered in timely manner which will result in better customer satisfaction, hence better revenue for network operator. This paper will focus on designing of a system that will create the clusters of data based upon user data using map-reduce architecture (Hadoop). It will do the enrichment of customer data in order to be aggregated against a required criterion. This paper will be relying on CDR (Call Data Record) to collect and identify the repeated patterns for different customers for mobile network users.

2. WHY BIG DATA FOR CDR ANALYSIS

Big data is any data which are having three characteristics like high volume, velocity and multiple variety. CDR generated by telecom operators are very huge and having high velocity with multiple type of data characteristics. Generally, network operators have multi-million customers and for each customer, system will have multiple CDR generated on day to day basis.

To analyse such mammoth amount of CDR data, network operator needs a system which should be based upon distributed architecture. This paper proposes to choose Hadoop as big data stack to do the profiling of customer based on spatial and temporal data retrieved from CDR.

3. IMPLEMENTATION APPROACH

i. As this paper intent to do the profiling of the customers, it will be focusing on CDR data to identify the movement of the subscriber on day to day basis. CDR contains the detailed information about the origination/ ending of all calls done by the subscriber.

ii. By having the analysis of the customer location and repeated patterns, telephone operators can offer better services and offers to their respective users.

iii. Operator will capture the raw CDR which are initially collected on temporal and spatial basis. This raw CDR data can be collected by operator to have the further data enrichment after filtering personal information so that user identification cannot be compromised.

iv. By having the spatial and temporal data from CDR, location of different user can be easily identified.

v. Next step is to create different clusters based upon the usage pattern by using below approach:

a) Capture the raw CDR from the operator.

b) Apply subspace clustering algorithm (PROCLUS) to identify the repeated or common patterns of the customer representing the different state.

c) Generation of different data matrix for different states of the customer will be generated.

d) Transition states should be scattered in such a way, so that they can be reduced on minimum number of filters or states.

e) These transition states can be used for doing various type of queries or analysis to have better service/ offer to the customer.

f) Based upon the states of the user derived through CDR, service provider needs to identify the mainstay which can be primary and secondary location of the customer e.g. if customer goes to office or park or market then service provider can create primary mainstay like office and work and secondary mainstay like market based upon frequency of commuting.

g) Criteria can be chosen based upon the frequency of repeated behavior. Accordingly, service provider can create cluster based upon primary mainstay and secondary mainstay.

h) Considered data contain only two fields, which are the temporal and spatial information relative to each registered telephonic activity. However, it is possible to extract useful implicit information contained in the dataset, adding additional features. The resulting structure of an element in the dataset are ref id, day of week, day of work, connection time, period of day, previous call. Description for the fields is as follows:

- ref_id is preserved as in the initial dataset. Since the values are exclusive identifiers, operator consider the domain for this field to be separate.
- day_of_week this field are having weekdays values as Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday. It can be deduced from the time and date of call connection.
- day_of_work It is an identifier which clearly differentiate between working and non-working day. It can be deduced from the time and date of call connection. For simplicity purpose take working day as 1 and non-working day as 0.
- connection_time is the time, when call is made by the subscriber on the network. It can be represented in hh:mm format. This is a regular ring-shaped domain in [00:00 AM, 11:00 PM] with resolution of 1 hour.
- period_of_day is the period of the day when the call was issued. calls happened in the time interval [08:00, 12:00] are considered morning activities, the ones in [12:00, 18:00] are afternoon activities and the ones in [18:00, 08:00] are evening/night activities. It will have then a nominal domain with the values for morning, afternoon, night/evening.
- previous_call is the spent time from the previous call.
- PROCLUS clustering algorithm is basically the key to achieve the above-mentioned steps.

4. CLUSTERING ALGORITHM – PROCLUS

Clustering is basically a methodology which is responsible for finding groups of similar data points in attributes of datasets. PROCLUS algorithm is very efficient algorithm for creating subspace clusters (dataset partitioning) based upon datapoints. In this subspace algorithm each data point is designated a specific cluster. This property makes it as ideal choice for customer demarcation and orientation analysis where partition of data point is required. This algorithm can also find oddity.

Proclus works by sampling the data by using K-medoids group or set, it is faster than other subspace clustering algorithms like CLIQUE which is another subspace algorithm based upon bottom upon bottom up approach. while proclus is based upon top down.The algorithm works in three stages consisting of initialization, iteration and refinement of the cluster. It takes two constant values which are part of data set, average dimensionability l and number of clusters k as input.

All three steps are mentioned in the below Figure.

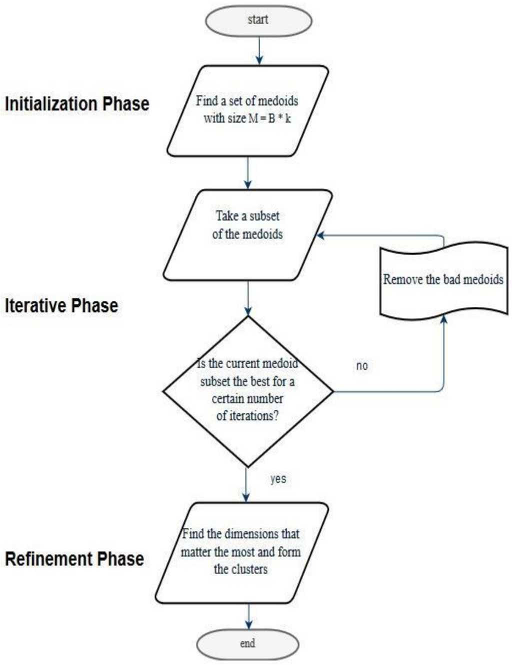
start
Initialization Phase
Find a set of medoids with size M = B * k
Take a subset of the medoids
Remove the bad medoids
Iterative Phase
Is the current medoid subset the best for a certain number of iterations?
no
yes
Refinement Phase
Find the dimensions that matter the most and form the clusters
end

Fig. 1 All three phases of Proclus, showing initialization, iteration and refinement [3].

5. HADOOP MAP REDUCE AND PROCLUS ALGORITHM

Proclus uses map and reduce at each stage of its execution to create the sub space cluster. Idea is to create the filtered medoids so that relevant data set can be created.

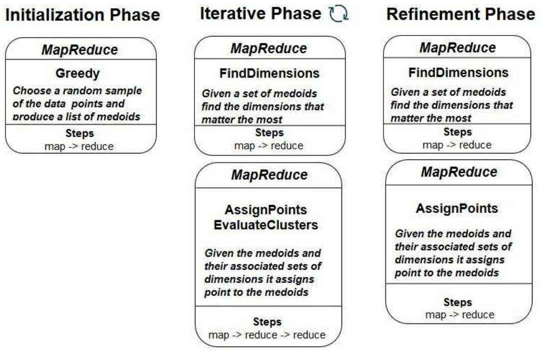
Initialization Phase
Iterative Phase
Refinement Phase
MapReduce
Greedy
Choose a random sample of the data points and produce a list of medoids
Steps
map -> reduce
MapReduce
FindDimensions
Given a set of medoids find the dimensions that matter the most
Steps
map -> reduce
MapReduce
FindDimensions
Given a set of medoids find the dimensions that matter the most
Steps
map -> reduce
MapReduce
AssignPoints
EvaluateClusters
Given the medoids and their associated sets of dimensions it assigns point to the medoids
Steps
map -> reduce -> reduce
MapReduce
AssignPoints
Given the medoids and their associated sets of dimensions it assigns point to the medoids
Steps
map -> reduce

Fig. 2 Proclus clustering algorithm showing how it is doing the mapping and aggregation in each stage [9].

6. BIG DATA STACK – HADOOP

Apache Hadoop is an open-source software framework which work on inexpensive hardware. It is used for distributed storage and distributed processing of very large data sets. It consists of computer clusters or node built from inexpensive hardware. Fault tolerance is one of the primary aspects of Hadoop. This paper will focus on using Hadoop as big data stack for our predictive analysis. In addition to Hadoop, this paper proposes to use Zookeeper, Kafka to leverage specific advantages of these tools.

Primarily Hadoop is divided into two parts i.e. Storage and Processing

i. Storage HDFS - It is responsible for fault tolerant storage. It maintains the replication or copies for each data at multiple nodes in cluster. It follows master (NameNode) slave (Data Node) architecture for all if its operations.

a) *NameNode* -It is the server or node which is responsible for managing the other nodes which contains the actual user data. It also keeps meta data for data nodes. Data nodes are basically slave nodes which contains the actual user data. NameNode has all the information like location, size, permissions about all the files which are stored in HDFS cluster.

Whenever there is a change in file metadata then NameNode captures the change for future processing of that requested data or file. Consider a scenario where a file has been removed from the data node, then Name Node will capture this metadata change in EditLog which basically a transaction log for HDFS. Another responsibility of NameNode is basically do the health check for all data nodes so that it can maintain the information about the availability for any requested or stored data. This also provides data locality service as it exactly knows the information about the data stored.

b) *DataNode* - Data nodes act as warehouse for actual user data. Data Nodes does serve the on demand read and write (create, update, delete) requests for the user. Replication and copies of specific data is done by data node but governed by the name node. It works as a slave node which is managed by the master node i.e. name node.

ii. Processing - Processing is done by map reduce along with PROCLUS sub space algorithm. please refer to section 4 and 5.

iii. Configuration and Synchronization – Zookeeper. Zookeeper is a unified service for maintaining naming and configuration data and responsible for providing adjustable and sturdy synchronization within distributed systems. It might possible that leader get down, then in that case Zookeeper helps us in selecting the new leader. For e.g. all read and write requests will be done by the leader for a topic of Kafka streaming service. The Zookeeper Atomic Broadcast (ZAB) protocol is the core of the system which ensures the atomicity of the operation. It can be viewed as atomic broadcast system for ordered updates.

Example: Leader selection for apache Kafka can be done by the zookeeper so that consistency of data can be maintained.

iv. Message Streaming – Kafka. As this paper focus on using big data solutions which is distributed in nature, so it needs an efficient, fault tolerant, low latency message streaming mechanism so that it encashes the big data stack in an efficient way.

Apache Kafka is streaming software which provides us the processing of messages between producer and consumer in a very fast, fault tolerant and scalable way. It runs as a cluster on multiple nodes. It allows many permanent or on demand consumers. It is fault tolerant and guarantees no data loss by having a replication of all messages received. It will be done by Kafka broker. Some of the advantages in compare to traditional messaging systems like ActiveMQ, RabbitMQ is as follows.

a) *High-throughput* – By having multiple consumers on topics whether it is permanent or ad-hoc.

b) *Low Latency* - Kafka has very low latency in compare other messaging systems like ActiveMQ and capable of keeping it in the range of milliseconds.

c) *Fault-Tolerant* - One of the best advantages is Fault Tolerance as it maintains the replication of all received data and leader selection in case of node failure for a topic partition.

d) *Durability* - It provides us the facility to do the replication for messages. It virtually guarantees that incoming message is never lost.

e) *Scalability* - It might possible that systems have various streams of data and has very high frequency.

V. SECURITY ASPECTS - HADOOP

Hadoop provides three types of security for data

a) *Encryption (DEK):* DEK encryption automatically applied to data in HDFS and in transit.

b) *Authentication*: Kerberos is integrated as authentication protocol in Hadoop. It provides better performance in compare to SSL.

c) *Access & Permissions (ACL):* Permissions can be set by individual, group, and role and set for specific data types and files. Basically, ACL based security can be done.

7. Conclusions

The proposed work has focused on detecting the user behaviour by analysing the CDR using big data stack with specific subspace clustering algorithm (PROCLUS). Hadoop big data stack along with subspace algorithm can do the data mining in a very short interval of time as it harnesses the capability of distributed systems in a very efficient and fault tolerant way. By following the approach mentioned in this paper, telecom operators can deduce the common, repeated and regular data points of their customers. These data points will be further filtered on demand to predict subscriber behaviour so that better offer and services can be provided to targeted customers.

References

[1] Calabrese, F., Colonna, M., Lovisolo, P., Parata, D., & Ratti, C., “Real-time urban monitoring using cell phones: A case study in Rome”, IEEE Transactions on Intelligent Transportation Systems, 12(1), 2010.

[2] Grindrod, Peter. "Infering behavior-based lifestyle categorizations based on mobile phone usage data." U.S. Patent Application 13/841,852 filed January 30, 2014.

[3] Jain, A. K., & Dubes, R. C., Algorithms for clustering data. Prentice-Hall, Inc, 1988

[4] Apache Foundation, Hadoop web page, http://hadoop.apache.org/.

[5] Apache Foundation, Zookeeper web page, https://zookeeper.apache.org/.

[6] Apache Foundation, Kafka web page, https://kafka.apache.org/.

[7] Apache Foundation, Kafka web page,https://kafka.apache.org/.

[8] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S., “Fast algorithms for projected clustering”, ACM SIGMoD Record, 28(2), 1999

[9] Nayagam, S. C. , “Comparative Study of Subspace Clustering Algorithms”, International Journal of Computer Science and Information Technologies, 6(5), 2015