uCLUST - A NEW ALGORITHM FOR CLUSTERING UNSTRUCTURED DATA

D. Venkatavara Prasad, Sathya Madhusudanan and Suresh Jaganathan
Department of Computer Science and Engineering, SSN College of Engineering, Chennai, India
E-Mail: dvvprasad@ssn.edu.in

ABSTRACT

Data that resides in a fixed field within a record or file is called structured data and have a defined schema. Unstructured Data refers to information that either does not have a pre-defined data model and does not fit well into relational tables. Clustering gains importance in the fields of Libraries (book ordering), Insurance (identifying groups and identifying frauds), WWW (document classification and clustering weblog data). Available clustering algorithms work only with structured data and use medoids as parameter for clustering. Clustering big data is not feasible, as they are mostly unstructured. It is not possible to label large collection of objects and identifying subspace clusters in unstructured data is a difficult task because of time complexity. In this paper, we proposed and designed a new algorithm called uCLUST, which identifies clusters in unstructured data as traditional distance functions cannot capture the pattern similarity among the objects. The proposed algorithm is applied in 6 different datasets and results are tabulated.

Keywords: big data, clustering, JSON, redis, unstructured data.

INTRODUCTION

Structured data [12, 13] depends on creating a data model, a type of business model recorded and how they are stored, processed and accessed. The data model includes defining what fields of data are available, how they are stored, what type of data and any restrictions on the data input. Structured data has the advantage of being quickly entered, stored, queried and analyzed. Because of the high cost and performance limitations of storage, memory and processing, relational databases and spreadsheets using structured data were the only way to manage data effectively. Structured data is often managed using Structured Query Language (SQL), a programming language created for managing and querying data in relational database management systems [4].

Unstructured data [12, 14] refers to information that either does not have a schema or not organized in a pre-defined manner. Unstructured information is typically text-heavy but may contain data such as dates, numbers, and facts as well. This result in irregularities and ambiguities that make it difficult to understand using traditional computer programs as compared to data stored in fielded form in databases. Data with some kind of structure may still be characterized as unstructured if its structure is not helpful for the processing task at hand. Unstructured information might have some structure (semi-structured) or even be highly structured, but in ways that are unanticipated or unannounced.

Clustering [15] could be defined as the process of organizing objects into groups whose members are similar in some way. Clustering on structured data is done using the distance measure between data points. If the components of the data instance vectors are all in the same physical units, then the simple Euclidean distance metric is sufficient to group similar data instances successfully. Clustering algorithms may be classified as i) Exclusive Clustering, ii) Overlapping Clustering, iii) Hierarchical Clustering, iv) Probabilistic Clustering. In Exclusive Clustering, data are grouped in an exclusive way, so that if an individual datum belongs to a definite cluster then it could not be included in another cluster (e.g. K-means [6]). Overlapping clustering uses fuzzy sets to cluster data, so that each point may belong to two or more clusters associated with an appropriate membership value (e.g. Fuzzy C-means [7]). A hierarchical clustering algorithm is based on the union between the two nearest clusters (e.g. HAC [3]). Probabilistic clustering uses a completely probabilistic approach (e.g. Mixture of Gaussian [2]).

Unstructured data files often include text and multimedia content. Examples include e-mail messages, word processing documents, videos, photos, audio files, presentations, web pages and many other kinds of business documents. Experts estimate that 80 to 90 percent of the data in any organization is unstructured. And the amount of unstructured data in enterprises is growing significantly often many times faster than structured databases are growing. Many organizations believe that their unstructured data stores include information that could help them make better business decisions. Unfortunately, its often very difficult to analyze unstructured data. Organizations have turned to a number of different software solutions designed to search unstructured data and extract valuable information. Because the volume of unstructured data is growing so rapidly, many enterprises have also turned to technological solutions to help them in managing and storing their unstructured data in a better way. These can include hardware or software solutions that enable them to make the most efficient use of their available storage space.

Clustering can be considered the most important unsupervised learning [1] problem. Like every other problem of this kind, it deals with finding a structure in the collection of unlabelled data. A cluster is, therefore, a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters.

We can show this with a simple graphical example (refer Figure-1).

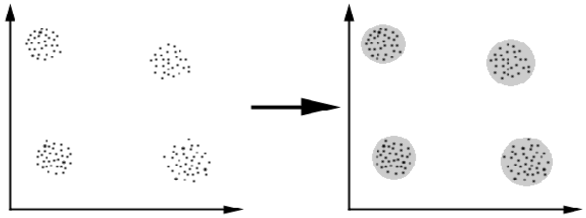


Figure-1. Example of clustering.

In this case, we easily identify the 4 clusters into which the data is divided and the similarity criterion used here is the geometrical distance (distance-based clustering). Another kind of clustering is conceptual clustering here two or more objects belong to the same cluster if this one defines a concept common to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

A cluster is a group of related documents, and clustering, also called unsupervised learning operated for the grouping of data on the basis of some similarity measure, automatically without having to pre-specify categories. We do not have any training data to create a classifier that has learned to group data. Without any prior knowledge of the number of groups, group size, and the type of data i.e. unstructured data, the problem of clustering appears challenging. In-use algorithms suffer from these problems, i) supports only small datasets, ii) time-consuming, iii) not dynamic and iv) supports only structured data.

Our contributions

The goal of clustering is to determine the intrinsic grouping in a set of unlabelled data. It is not feasible to label large collection of objects as there is no prior knowledge of the number and nature of groups in data as clusters may evolve over time. The problems involved in clustering are the following:

- current clustering techniques do not address all the requirements adequately and concurrently.
- dealing with a large number of dimensions and a large number of data items can be problematic because of time complexity.
- a new similarity model is needed for unstructured data as traditional distance functions cannot capture the pattern similarity among the objects.
- the result of the clustering algorithm is arbitrary in many cases.

All these problems made us to develop a tool comprising of an algorithm to cluster the unstructured data. Most of the big data[11] are unstructured data; clustering these unstructured data becomes a challenging task. So, we designed and developed a new algorithm for clustering the unstructured data. We tested our algorithm with 6 different datasets, they are: i) Aadhaar, ii) Census, iii) NYC social media, iv) King County social media, v) Seattle online inventory, vi) State of Oregon social media. Aadhaar and Census datasets are created by us and are stored using Redis.

RELATED WORKS

NoSQL [8] is a set of concepts that allows the rapid and efficient processing of data sets with a focus on performance, reliability, and agility. They are schemaless databases and they offer high scalability. NoSQL systems are designed to run on clusters of processors and machines, and fit better for big data scenarios. NoSQL follows CAP theorem which stands for Consistency, Availability and Partition Tolerance out of which only two can be satisfied at a time. Key-value store (KVS) is one of the types of NoSQL databases. Each value in Key-value store can be saved with a unique key. The key-value concept is equivalent to the notion of hash addressing. Redis [8] is one of the KVS databases which do not support complex queries or indexing.

The implementation of k-Means clustering algorithm [10] for clustering unstructured text documents is proposed. The document that is to be clustered is represented in the form of vector, such that the words represent dimensions of the vector and frequency of the word in the document is the magnitude of the vector. Before preparing vector for a document, the following techniques are applied on the input text: i) The stop words are excluded ii) stemming is performed in order to treat different forms of a word as a single feature. Additional transformations have been implemented to the vector representation by using ***tf**-idf*(term frequency - inverse document frequency) formulation.

The weight $w(j)$ assigned to word j in a document is dependent on the *tf- idf* formulation as in equation (1):

$$w(j) = tf(j) * \log_2(N / df(j)) \tag{1}$$

where $\log_2(N / df(j)) = idf(j)$

The *tf-idf* measure can be normalized to a unit length of a document D as in equation (2):

$$norm(D) = \sqrt{\sum w(j)^2} \tag{2}$$

The most important factor in the clustering algorithm is the similarity measure. In order to find the similarity between two vectors, Cosine similarity[9] is used as in equation (3):

$$\text{cosine}(d1, d2) = \sum (w_{d1}(j) * w_{d2}(j) / norm(d1) * norm(d2)) \tag{3}$$

where $d1, d2$ represents the documents to be clustered. Algorithm for clustering the document is detailed below:

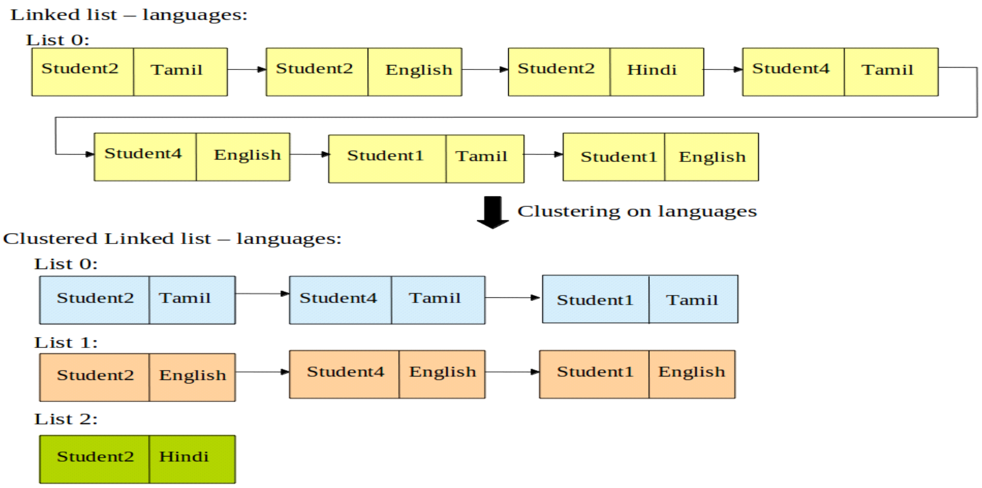
1. Distribute all documents among k bins:
2. A bin is an initial set of documents that is used before the algorithm starts. It can also be considered as the initial cluster.

The mean vector of the vectors of all documents is computed and is referred to as global vector.

- The similarity of each document with the global vector is computed.
- The documents are sorted on the basis of similarity computed in the previous step.
- The documents are evenly distributed to k bins.

3. Compute mean vector for each bin.
4. Compare the vector of each document to the bin means and note the mean vector that is most similar.
5. Move all documents to their most similar bins.
6. If no document has been moved to a new bin, then stop; else go to step 2.

This proposed work takes only frequency of words into account to calculate the similarity measure. Language Semantics and context of terms are not taken into account for clustering the documents. A system and method for analysing structured and unstructured data is proposed in paper[5]. This system is used for the analysis of both structured and unstructured data. Data extracted from a variety of unstructured data are well analysed to extract individual pieces of information and understand the relationships that exists between the extracted data. These transformed data and relationships may then be converted to a structured schema by passing them through an extraction/transform/load (ETL) layer. These structured schemas are then analysed using the structured data analysis tools.



uCLUST tool

Figure-2 shows the architecture design of *uCLUST*. Two Web Portals are designed using PHP to collect unstructured data (Aadhaar, Census) from the user and stored in Redis database as a key-value pair. In paper[8], the persistence of objects in Redis is explained. Data structures in Redis are called as Redis Data types. These include strings, lists, sets, sorted sets and hashes. The whole dataset is kept in-memory and therefore cannot exceed the amount of physical RAM. Redis server writes entire dataset to disk at configurable intervals. The unstructured datasets stored in the Redis database are processed and converted to JSON documents. *uCLUST* collects these JSON documents and does the clustering process by using singly-linked list. *uCLUST* then outputs the clustered results for the unstructured data based on the input key.

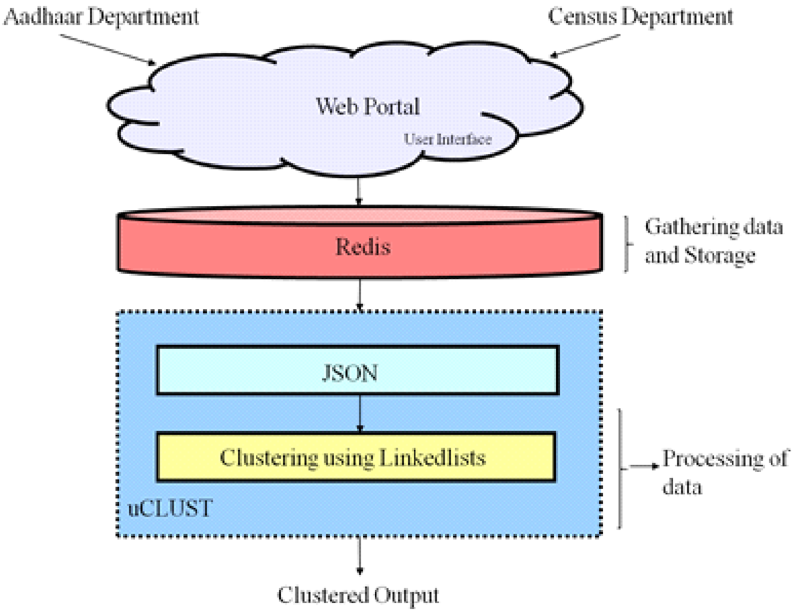


Figure-2. Architecture design of uCLUST.

Algorithm of uCLUST

Data: JSON document

```
Result: Clustered Output
read clustering attribute;
while not end of JSON document do read each record;
    if each record contains the clustering attribute then
        if the clustering attribute is single−valued then
            create a node containing the attribute value and its id;
        else
            create a node for each attribute value along with its id.;
        end
    end
    Insert the nodes created for each record in the linked list.;
end
while not end of the linked list do
    Retain the first distinct node value in the linked list itself;
    if a next distinct value is found then
        if there exists a linked list for the distinct value then
            Delete the node from the current linked list and insert the node into the linked list, which has the same node value;
        else
            Create a new linked list for the distinct value
        end
    end
end
Output all the linked lists created;
```

uCLUST explained

Figure-3 shows the working of uCLUST. The Aadhaar and Census datasets are stored in Redis as key-value pair. Each user is identified using a key. All the attributes of a user are mapped to the key using hash. The unstructured datasets stored in the Redis database are processed and converted to JSON documents. This conversion is done using rdbtools which is available in Redis. The JSON documents are then passed to the uCLUST for clustering. The clustering attribute is got as input from the user. The clustering process is done using the linked lists. For each new distinct value found in the JSON document, a new linked list is created. These linked lists created represent the clusters or groups.

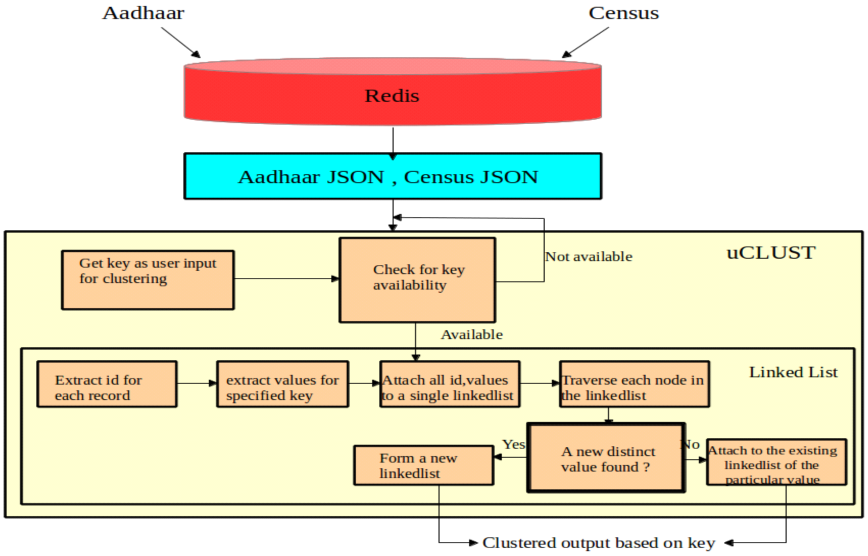

Aadhaar
Census
Redis
Aadhaar JSON , Census JSON
uCLUST
Get key as user input for clustering
Check for key availability
Not available
Available
Linked List
Extract id for each record
extract values for specified key
Attach all id,values to a single linkedlist
Traverse each node in the linkedlist
Form a new linkedlist
Yes
A new distinct value found ?
No
Attach to the existing linkedlist of the particular value
Clustered output based on key

Figure-3. Working of uCLUST.

Example

A sample JSON document is clustered based on the attributes languages and gender. Figure-4 shows the clustered output based on a single attribute. Figure-5 shows the clustered output based on two attributes.

Sample JSON file

[{
"student2":{"firstname":"sekar","gender":"M","languages":"[\"Tamil\",\"English\",\"Hindi\"]"},
"student4":{"firstname":"Rani","lastname":"shankar","gender":"F","languages":"[\"Tamil\",\"English\"]"},
"student1":{"languages":"[\"Tamil\",\"English\"]","gender":"M","firstname":"vijay","lastname":"sampath"}
}]

clustering key - languages , gender

Clustered output - single attribute

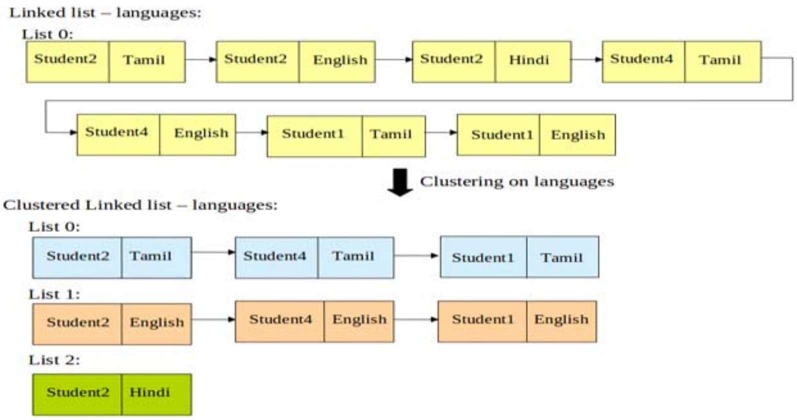


Figure-4. Clustered Linked list based on languages.

Clustered output - two attributes

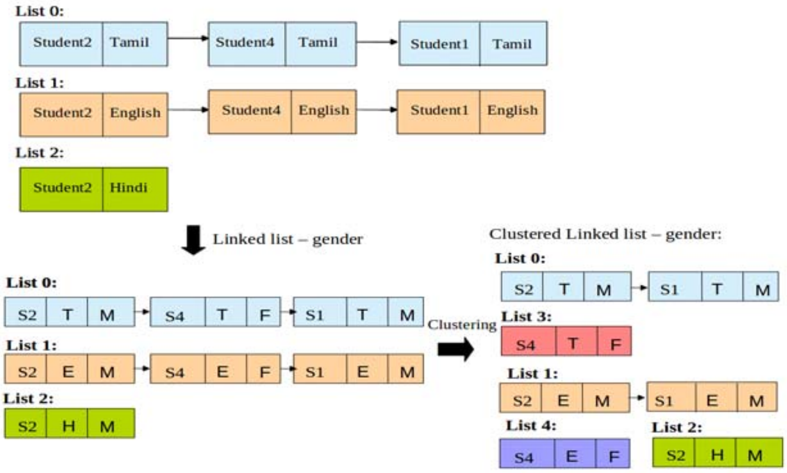


Figure-5. Clustered linked list based on languages, gender.

Strengths of uCLUST

In day-to-day life, most of the real-time data are unstructured in nature. There are no direct algorithms available to cluster these unstructured data. Unstructured data can be transformed to a well-defined schema and later the available clustering algorithms can be applied on these structured data. *uCLUST* aids in clustering the unstructured data directly. The following advantages are achieved by using uCLUST: i) preprocessing of unstructured data is avoided and ii) time complexity is reduced.

Performance overhead of uCLUST

uCLUST clusters unstructured data with a single attribute efficiently. But, when clustering the unstructured data with multiple attributes, more time is consumed while processing large datasets.

Performance evaluation

Datasets

We used six datasets, out of six, four datasets are already available and their details are tabulated in Table-1. The remaining two datasets, Aadhaar and Census are generated by using Web Portal designed using PHP.

Table-1. Details of the datasets used.

Dataset	No. of. records	Format	Link
NYC Social Media Usage	1,50,000	JSON	*https://data.cityofnewyork.us/api/views/5b3ars48/rows.json*
King County Social Media	1,50,000	JSON	*https://data.kingcounty.gov/api/views/rfcd-nnxe/rows.json*
Seattle Communities Online inventory	1,50,000	JSON	*https://data.seattle.gov/api/views/5ytf-wban/rows.json*
State of Oregon Social Media Sites	1,50,000	JSON	*https://data.oregon.gov/api/views/hqhe-shsc/rows.json*
Aadhaar	1,50,000	JSON	Manually created and Collected using Web Portal and stored in Redis
Census	1,50,000	JSON	Manually created and Collected using Web Portal and stored in Redis

Results

uCLUST algorithm for clustering single attribute was implemented and tested for the datasets specified above. Table-2 shows the running time for clustering a single attribute of the datasets.

Table-2. uCLUST running time for single-attribute clustering.

Datasets	Running time for single attribute					
	1000 records	5000 records	10000 records	50000 records	1,00,000 records	1,50,000 records
	seconds					
Aadhaar	4	6	9	12	17	25
Census	5	8	12	16	22	30
NYC Social Media	6	8	14	17	22	29
King County Social Media	4	7	9	13	16	22
Seattle Online inventory	5	9	13	17	21	25
State of Oregon Social Media	7	10	14	18	24	32

Figure-6 shows the running time for clustering a single attribute of the datasets.

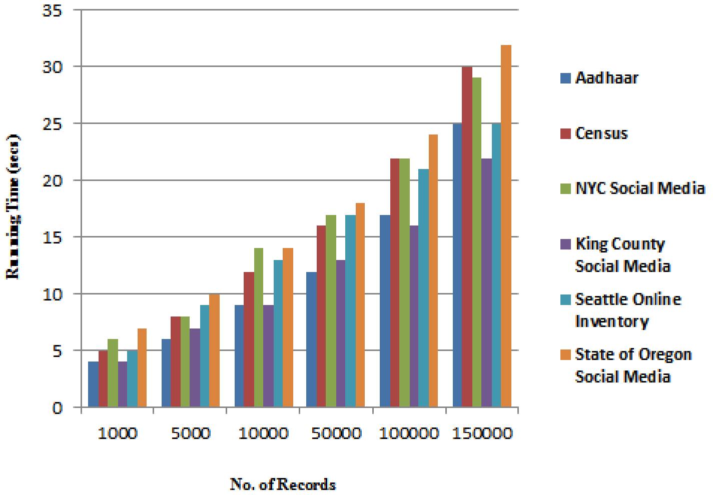


Figure-6. Running time for single attribute clustering.

Table-3 shows the running time for clustering two attributes of the specified datasets.

Table-3. uCLUST Running time for clustering two attributes.

Datasets	Running time for two attributes					
	1000 records	5000 records	10000 records	50000 records	1,00,000 records	1,50,000 records
	seconds					
Aadhaar	75	160	300	450	800	950
Census	84	170	325	478	864	988
NYC Social Media	89	175	338	486	870	992
King County Social Media	72	150	290	450	820	974
Seattle Online inventory	90	190	333	482	877	1002
State of Oregon Social Media	95	210	350	490	890	1050

Figure-7 shows the running time for clustering two attributes of the datasets.

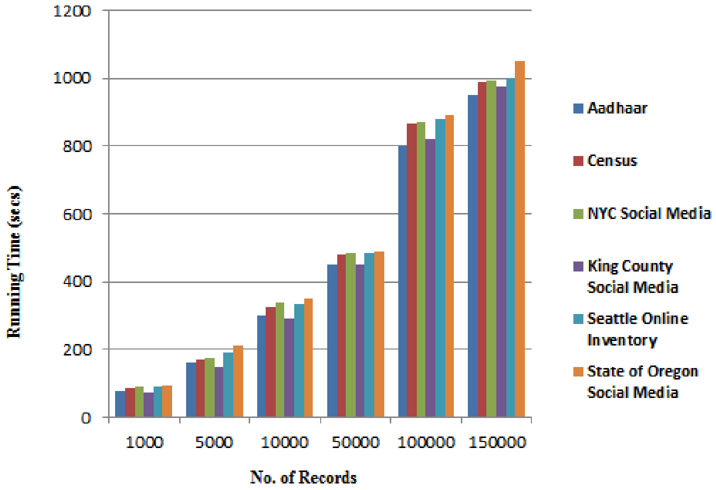


Figure-7. Running time for clustering two attributes.

Table-4 shows the running time for clustering five attributes of the specified datasets.

Table-4. uCLUST Running time for clustering five attributes.

Datasets	Running time for five attributes					
	1000 records	5000 records	10000 records	50000 records	1,00,000 records	1,50,000 records
	minutes					
Aadhaar	125	300	584	784	1367	1597
Census	137	311	570	800	1440	1617
NYC Social Media	150	317	614	823	1458	1660
King County Social Media	120	280	533	757	1362	1574
Seattle Online inventory	148	313	584	803	1449	1642
State of Oregon Social Media	157	333	634	887	1468	1672

Figure-8 shows the running time for clustering five attributes of the datasets.

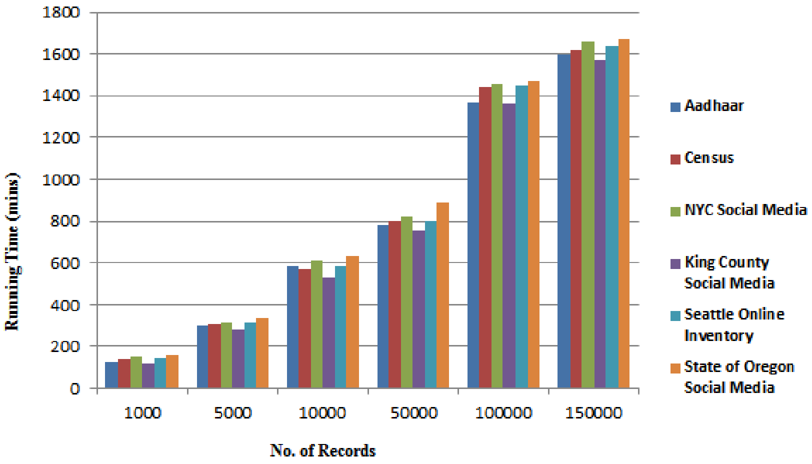


Figure-8. Running time for clustering five attributes.

Table-5 shows the comparison of results for clustering 1, 50, 000 records with single, two and five attributes.

Table-5. Comparison of results for clustering 1,50,000 records.

Datasets	**1, 50, 000 records**		
	Single attribute	**Two attributes**	**Five attributes**
	secs	**secs**	**mins**
Aadhaar	25	950	1597
Census	30	988	1617
NYC Social Media	29	992	1660
King County Social Media	22	974	1574
Seattle Online inventory	25	1002	1642
State of Oregon Social Media	32	1050	1672

Figure-9 shows the comparison of results for clustering 1, 50, 000 records with single, two and five attributes.

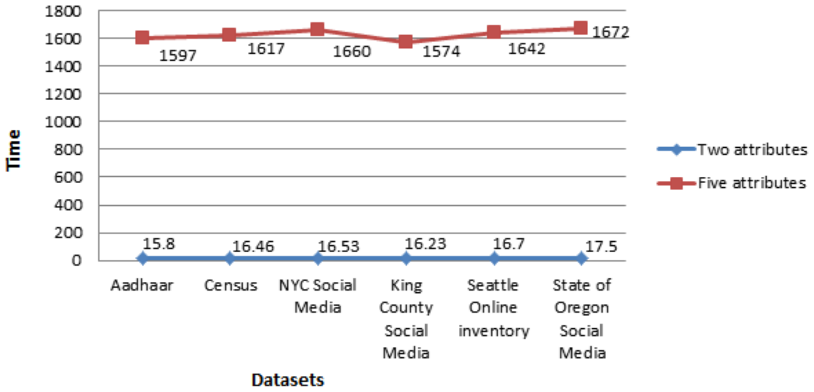


Figure-9. Comparison of results for clustering 1, 50, 000 records.

From Table-5 and Figure-9, it can be realized that, when clustering a large data the processing time increases. To reduce this, clustering process can be parallelized.

CONCLUSIONS

Clustering of unstructured data is a challenging task and in the Big Data era, most of the data are unstructured. In this paper we have proposed and designed a new clustering algorithm called uCLUST, which clusters the unstructured data. The proposed algorithm is tested using 6 different datasets. We manually created a two new dataset called Aadhaar and Census data and stored it in Redis database, which is key-value store, stored in a unstructured format. The results obtained are tabulated and shows that the designed clustering algorithm works well, as every work may lag in some point, our *uCLUST* algorithm lags while clustering more than 5 fields. The performance of the algorithm degrades in terms of compilation time; this problem can be solved by using the parallelization concepts, which is our future work.

REFERENCES

Barlow, Horace. “Unsupervised learning”. Neural Computing. 1.3(1989): 295-311.

Cathy Maugis, Gilles Celeux, Marie-Laure Martin-Magniette. “Variable Selection for Clustering with Gaussian Mixture Models”. Biometrics, Vol: 65(3), pages: 701-709, September 2009, DOI: 10.1111/j.1541-0420.2008.01160.x.

Hung Chim, Xiaotie Deng. “Efficient Phrase-Based Document Similarity for Clustering”. Knowledge and Data Engineering, IEEE Transactions on Vol.20 (9), pages: 1217-1229, March 2008, ISSN: 1041-4347.

Jun-Sung Kim, Kyu-Young Whang, Hyuk-Yoon Kwon, Il-Yeol Song. “PARADISE: Big data analytics using the DBMS tightly integrated with the distributed file system”. Springer Science+Business Media New York 2014, DOI 10.1007/s11280-014-03 12-2.

Justin Langseth, Nithi Vivatrat, and Gene Sohn. “Analysis and transformation tools for structured and unstructured data”. January 11, 2007, US20070011183 A1.

K. A. Abdul Nazeer, M. P. Sebastian. “Improving the Accuracy and Efficiency of the K-means Clustering Algorithm”. Proceedings of the World Congress on Engineering, July 1-3, 2009, ISSN: 978-988-17012-5-1.

Lin Zhu, Fu-lai Chung, Shitong Wang. “Generalized Fuzzy C-means Clustering Algorithm with Improved Fuzzy Partitions”. IEEE Transactions, January 2009, Vol: 39(3), pages: 578-591. ISSN: 1083-4419.

Matti Paksula. “Introduction to store data in Redis, a persistent and fast key-value database”. AMICT, pages 39-49, 2010-2011.

Richardo Bazes, Yates Berthier Ribeiro, Neto. A Book on “Modern Information Retrieval”, ACM Press, 1999.

Yasir Safeer, Mustafa Atika, and Anis Noor Ali. “Clustering Unstructured Data (Flat Files), an Implementation in Text Mining Tool”. International Journal of Computer Science and Information Security, 8(2), pp. 174-180, 2010, ISSN: 1947-5500.

A Report - “Planning Guide - Getting Started with Big Data”, Intel IT Center, February 2013.

http://smartdatacollective.com/michelenemschoff/206391/quick-guide-structured-and-unstructured-data

http://www.webopedia.com/TERM/S/structured_data.html.

http://www.webopedia.com/TERM/U/unstructured_data.html.

http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/.