ISSN: 2321-7782 (Online)
Volume 2, Issue 11, November 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Categorizing the Actors in Twitter based on Consolidated Management

Ragini Veeramachaneni[1]
Student
Department of Computer Science and Technology
VR Siddhartha Engineering collage
India

Sandeep Yelisetti[2]
Assistant Professor
Department of Information and Technology
VR Siddhartha Engineering collage
India

Abstract: ***Twitter is an online social micro blogging service that enables users to send and read tweets. Registered users can read and post tweets, but unregistered users can only read them. As twitter contains an immense amount of data it is difficult to observe and predict the collective behavior. To address the scalability issue, various clustering schemes are used to extract sparse social dimensions. The approaches are used to handle how fairly the tweet rate is identified considering each aspect for actors, while demonstrating a comparable prediction performance to other non-scalable methods now posed to analyze the performance of hierarchical clustering technique. The recognition of user tweet rate becomes a boon to increase the rating of a pronominal aspect. An inimitable advantage of this model is that it easily scales to handle contacts with millions of actors while the earlier models fail. The scalable approach offers a viable solution to effective learning of online consolidated management on a large scale.***

Keywords: Twitter, Hierarchical clustering, Agglomerative hierarchical clustering, Divisive collective behaviour.

I. INTRODUCTION

Analysis of micro blogging sites during crises situation has seen a rising interest as discussed in [1] [2]. Content oriented analysis by applying traditional natural language techniques using syntactic and semantic model is difficult due to reasons described in [2]. These can be summarized as:

• Tweets are very short in length with the message length being about 140 characters. Such a short piece of text provides very few contextual clues for applying machine learning techniques.

• Tweets are written in informal style and often consist of simple phrases, sentence fragments and or ungrammatical text. They contain abbreviations, internet slang and misspelled words.

• Tweets may contain implied references to locations as described in [3]. Hence, named entity recognition using off the shelf named entity recognizers yield poor results.

We believe that clustering of tweets will help to easily categorize them based on their content. Using such clusters we would be able to identify the topic or particular event about which the tweet is. In this work, we would thus like to define a process that will classify an incoming tweet to one of the clusters existing in the system using topic model approach. Currently, we have analyzed the clusters generated using differently trained topic models [4]. These topic models vary in size of training data, training data itself and number of topics.

1. We determine the topic model configuration that is optimal to cluster tweets. Typically, in machine learning a model is trained using data that belongs to the same domain as the test data. For example, a named entity recognition system for biology related data is trained on biological data. But as mentioned above the short nature and esoteric form of tweets makes it

necessary to explore if a topic model trained on twitter can yield better performance compared to a topic model trained on new wire text which has more contextual information.

2. We then evaluate the decisions made in point 1 by clustering a new set of tweets and also estimate the accuracy of the results. We compare the accuracy obtained with a baseline approach to show the merit of topic model based approach.

3. We show that the use of topic model to cluster Twitter users based on their status updates. We show the merit of topic model based approach to cluster Twitter users.

II. Related Work

Mining plays a crucial role for any data extraction. Data mining commonly involves four classes of tasks such as Classification, Clustering, Regression, Association rule learning [5]. Generally we use many kinds of tools and languages to collect data from social environment. In this application the data set is retrieved through Python language [6]. By retrieving the authentication keys from Twitter API, we gather data by executing python code using unique keys.

Python is a broadly useful, abnormal state programming dialect whose outline rationality stresses code meaningfulness. Python cases to consolidate "wonderful force with clear sentence structure" and its standard library are huge and exhaustive. Python helps various programming standards, basically however not constrained to protest situated, basic and, to a lesser degree, useful programming styles. It offers a completely dynamic sort framework and programmed memory administration, like that of Scheme, Ruby, and Perl.

- Python is free and open source programming and has a group based improvement model.
- Python is powerful.
- Python is object oriented.

A. Creating Application in Twitter

For gathering obliged information from twitter somebody ought to begin with making an application in twitter site. New applications can be made at twitter engineer's page [7]. At whatever point another application is made twitter API produces some mystery keys which ought to be utilized by engineer as a part of the methodology of validating his/her application. Applications can be made at the page.

Oauth is an open standard for approval. It permits clients to impart their private assets put away on one site with an alternate site without needing to pass out their certifications, commonly by authenticating username and password we acquire tokens while we create an application Twitter API [8]. The names of keys are:

- Consumer key
- Consumer Secret
- Access key
- Access Secrete

These keys permits a client to concede an outsider site access to their data put away with an alternate administration supplier, without imparting their right to gain entrance consents or the full degree of their information.

B. Collecting the tweets from Twitter:

Twitter has helpfully selected to give you administer access to your own particular access token and access token mystery; with the goal that you can sidestep the Oauth move for a specific application you've made under your record. You can discover

a "My Access Token" connection to these qualities under your application's points of interest [9]. For our application TWEETPOOL1 all the keys created are recorded.

III. PROPOSED WORK

Generative models have been popular for document analysis. A generative model is a model for randomly generating observable data, typically given some hidden parameters. It specifies a joint probability distribution over observation and label sequences. Often these generative models talk about a special type called topic model. There has been some work around analysis of Twitter data using topic models. In this chapter, we will explain few background concepts that are necessary to understand this thesis work. We will also review some recent research about analyzing online social media using topic models. Topic models are generative models and a popular method for modeling term frequency occurrences for documents in a given corpus. The basic idea is to describe a document as mixture of different topics. A topic is simply a collection of words that occur frequently with each other.

A. Clustering Implementation:

Clustering is an unsupervised learning techniques that takes a collection of objects such as tweets and organizes them into groups based on their similarity. The groups that are formed are known as clusters.

B. Hierarchical Clustering:

Agglomerative (bottom-up): Agglomerative algorithms begin with each individual document as a separate cluster, each of size one. At each level the smaller clusters are merged to form a larger cluster. It proceeds this way until all the clusters are merged into a single cluster that contains all the documents.

Divisive (top-down): Divisive algorithms begin with the entire set and then the splits are performed to generate successive smaller clusters. It precedes recursively until individual documents are reached. The agglomerative algorithms are more frequently used in information retrieval than the divisive algorithms. The splits and merge are generally done using a greedy algorithm. A greedy algorithm is an algorithmic approach that makes the locally optimal choice at each stage of its run with the hope of finding the global optimum.

Birch approach: It is neighborhood in that each one grouping choice is made without filtering all information focuses and as of now existing groups. It abuses the perception that information space is not generally consistently possessed and not every information point is just as essential. It makes full utilization of accessible memory to infer the finest conceivable sub-groups while minimizing I/O costs. It is likewise an incremental technique that does not require the entire dataset ahead of time.

C. Clustering algorithms

Agglomerative Hierarchical Clustering:

1. Start with N clusters, each containing a single entity, N × N symmetric matrix of distances. Let d_{ij} = distance between item i and item j.
2. Examine distance matrix for the nearest pair clusters and denote the distance between these most similar clusters U and V by d U V.
3. Merge clusters U and V into a new cluster, labeled T. Update the entries in the distance matrix by a. Deleting the rows and columns corresponding to clusters U and V, as b. Adding a row and column giving the distances between the new cluster T and all the remaining clusters.
4. Repeat above two steps total of N-1 times.

5. Consider distance matrix as clustering criteria. The approach not requires k clusters as an input, but needs a termination condition.

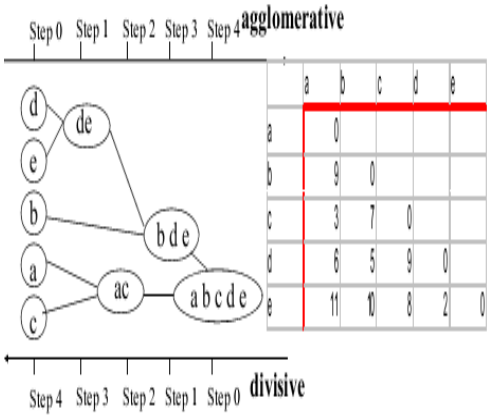


Fig.1: Comparison of Agglomerative and Divisive approaches.

Average Linkage Method:

The distance between clusters is the average distance between pairs of observations.

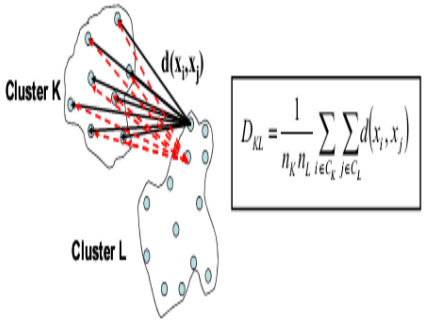


Fig. 2: Simple cluster form of Average Linkage Method

Average linkage has a tendency to join groups with little differences and it is somewhat one-sided to creating bunches with the same fluctuation[7]. Since it considers all parts in the bunch instead of simply a solitary point, however average linkage has a tendency to be less affected by amazing qualities than different strategies.

Centroid Linkage Method:

The distance between clusters is defined as the (squared) Euclidean distance between cluster centroid.

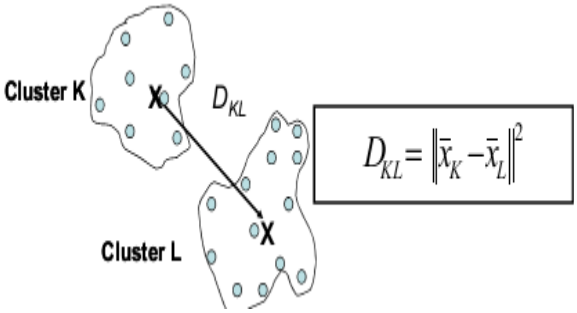


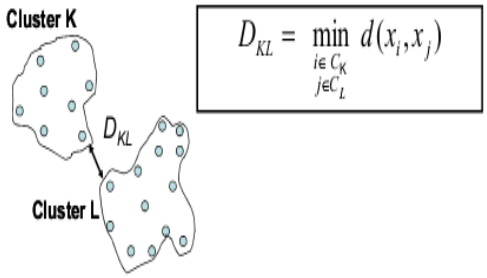
Fig. 3: Simple cluster form for Centroid Linkage Method

The centroid system thinks about group implies anomalies influence it short of what most different means, exceptions influence it short of what most other progressive grouping strategies. In different regards, then again, it may not execute and Ward's strategy or normal linkage (Milligan 1980).the bigger of two unequally estimated gatherings The bigger of two unequally measured gatherings fused utilizing centroid linkage has a tendency to command the consolidated cluster. Complete linkage is strongly biased toward producing compact clusters with roughly equal diameters, and it can be severely distorted by

moderate outliers. Complete linkage ensures that all items in a cluster are within some maximum clusters are within some maximum distance of one another.

Single Linkage Method:

The distance between two clusters is based on the points in each cluster that are nearest together Single Linkage Method in each cluster that are nearest together.



Fig, 4: Simple cluster form for Single lnkage method.

Single linkage clustering is otherwise called closest neighbor bunching. Single linkage has numerous attractive hypothetical properties however has fared ineffectively in Monte Carlo studies. By forcing no stipulations on the state of groups, single linkage offerings execution in the recuperation of minimal bunches as an exchange for the capacity to distinguish stretched and sporadic clusters. Likewise single linkage has a tendency to hack off the tails of Also single linkage has a tendency to slash off the tails of disseminations before differentiating the principle groups. The infamous fastening propensity of single linkage can be eased by indicating the Trim= choice.

Birch clustering:

- The algorithm starts with single point clusters. (every point in a database is a cluster)
- Then it groups the closest points into separate clusters, and continues, until only one cluster remains.
- The computation of the clusters is done with a help of distance matrix (O (n^2) large) and O (n^2) time.

CF Tree: A height balanced tree with two parameters:

– branching factor B

– Threshold T

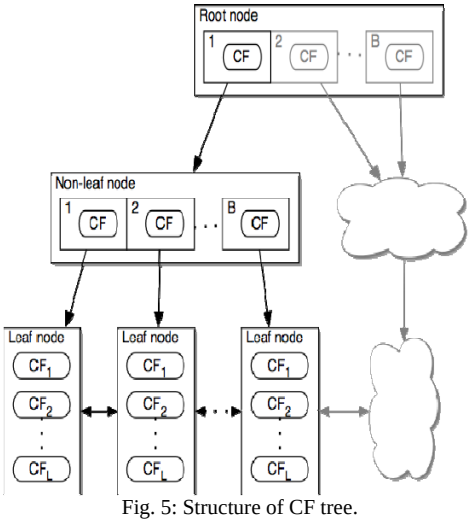


Fig. 5: Structure of CF tree.

Each non-leaf node contains at most B entries of the form [CF_i, $child_i$], where $child_i$ is a pointer to its i-th child node and CF_i is the CF of the subcluster represented by this child. So, a non-leaf node represents a cluster made up of all the sub clusters represented by its entries.

- A leaf node contains at most L entries, each of tem of the form [CF_i], where I =1, 2,... L.
- It also has two pointers, prev and next, which are used to chain all leaf nodes together for efficient scans.
- A leaf node also represents a cluster made up of all the sub clusters represented by its entries.
- But all entries in a leaf node must satisfy a threshold requirement, with respect to a threshold value T. The diameter has to be less than T.
- The tree size is a function of T (the larger the T is, the smaller the tree is).
- We require a node to fit in a page of size of P.
- B and L are determined by P (P can be varied for performance tuning).
- Very compact representation of the dataset because each entry in a leaf node is not a single data point but a subcluster.
- The leave contains actual clusters.
- The size of any cluster in a leaf is not larger than T.

Example of the CF Tree Insertion:

If the branching factor of a non-leaf node cannot exceed 3, then the root is split and the height of the CF Tree increases by one.

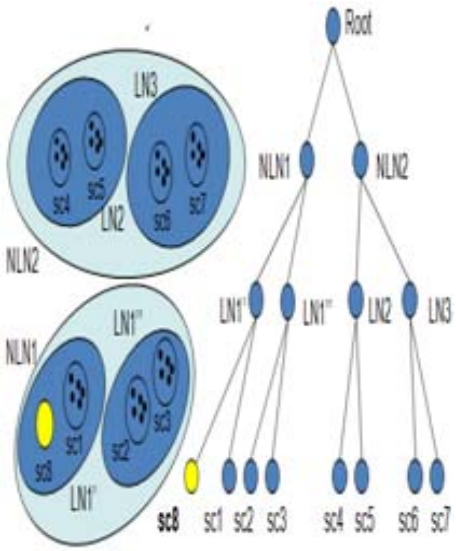


Fig. 6: Insertion in CF tree

Phase 1:

Scan all data and build an initial in-memory CF tree, using the given amount of memory and recycling space on disk. Starts with initial threshold, scans the data and inserts points into the tree. If it runs out of memory before it finishes scanning the data it increases the threshold value and rebuilds a new smaller CF tree by re-inserting the leaf entities from the older tree and then resuming the scanning of the data from point at which was interrupted.

Phase 2:

Condense into desirable length by building a smaller CF tree. Preparation for Phase 3.Potentially, there is a gap between the size of Phase 1 results and the input range of Phase 3.It scans the leaf entries in the initial CF tree to rebuild a smaller CF tree, while removing more outliners and grouping crowded sub clusters into larger ones.

Phase 3:

Global clustering: It uses a global or semi-global algorithm to cluster all leaf entries. Adapted agglomerative hierarchical clustering algorithm is applied directly to the sub clusters represented by their CF vectors.

Phase 4:

Cluster refining – this is optional, and requires more passes over the data to refine the results. Additional passes over the data to correct inaccuracies and refine the clusters further. It uses the centroid of the clusters produced by Phase 3 as seeds, and redistributes the data points to its closest seed to obtain a set of new clusters.

IV. Results and Observation

We conducted a thorough experimental evaluation of the proposed techniques on real twitter data set. Initially, we select data set and load it into database, then perform clustering approaches one after another. The sample dendrogram of agglomerative clustering is shown in below graph.

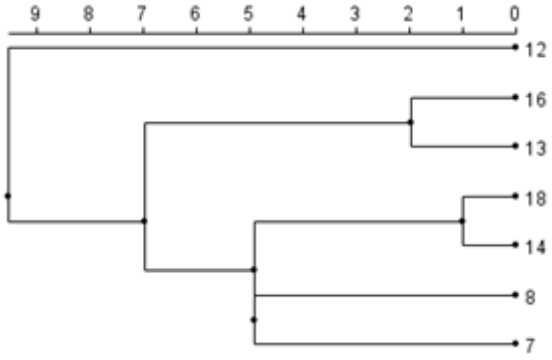


Fig. 7: Dendrogram for Agglomerative

By the keen observation of all the results, the time complexity of Birch is efficient over agglomerative and divisive clustering techniques.

V. Conclusion

In this work, we proposed a framework to perform clustering on data to rate the tweets related to the specific aspect. Social dimensions are extracted to represent the potential affiliations of actors before massive data sets. The hierarchical clustering techniques are performed to cluster the sparse datasets. By analyzing the performance of the sub techniques agglomerative clustering, divisive clustering and birch clustering, the time complexity of Birch clustering is low relatively.

References

1. Miller, Claire Cain (October 30, 2010). "Why Twitter's C.E.O. Demoted Himself". The New York Times. Retrieved October 31, 2010.
2. Lie Tang, Xufei Wang, and Huan Lix, “Scalable Learning of Collective behavior.” IEEE 2012 Transaction on knowledge and Data Engineering, Volume 24, Issue 6, . New York, NY, USA: ACM, 2009, pp. 1107–1116.
3. “Relational learning via latent social dimensions,” in KDD ’09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, 2009, pp. 817–826.
4. M. Newman, “Finding community structure in networks using the eigenvectors of matrices,” Physical Review E (Statistical, Nonlinear, and Soft Matter Physics), vol. 74, no. 3, 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.74.036104
5. P. Singla and M. Richardson, “Yes, there is a correlation: - from social networks to personal behavior on the web,” in WWW ’08: Proceeding of the 17th international conference on World Wide Web. New York, NY, USA: ACM, 2008, pp. 655–664.
6. Tang and H. Liu, “Toward predicting collective behavior via social dimension extraction, ”IEEE Intelligent Systems, vol. 25, pp. 19–25, 2010.
7. A. T. Fiore and J. S. Donath, “Homophily in online dating: when do you like someone like yourself?” in CHI ’05: CHI ’05 extended abstracts on Human factors in computing systems. New York, NY, USA: ACM, 2005, pp. 1371–1374.
8. H. W. Lauw, J. C. Shafer, R. Agrawal, and A. Ntoulas, “Ho- mophily in the digital world: A Live Journal case study, ”IEEE Internet Computing, vol. 14, pp. 15–23, 2010.
9. X. Zhu, “Semi-supervised learning literature survey,” 2006. [Online]. Available: http://pages.cs.wisc.edu/~jerryzhu/ pub/sslsurvey 1292006.pdf

