INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Harmonizing User Search Data with Efficient Adaptive Clustering

P.SOUJANYA NAIDU[1], K.RAVINDRA[2], Y.RAMESH[3], K.PAVAN RAJU[4]

[1]Final year M.tech Student, [2] Associate Professor, [3,4]Asst.professor
[1, 2, 3] Avanthi Institute of Engineering and Technology (JNTUK), Cherukupally, Vizianagaram Dist, Andhra Pradesh, India
[4]Shri Vishnu Engineering College for Women, Bhimavaram, Andhra Pradesh, India

Abstract-*Web based applications are at full stretch in today world. For a small piece of info we can easily find it on web with the help of search engine to us on web.Arrangments of the travel oriented phenomena, financial management, online purchases respectively we are dependent on web. To properly guide the users for their information quests on the web, search engines keep track of their queries and clicks while searching online. In this paper we are proposing a dynamic clustering algorithm will help us to group related queries together in such a way that the user can have faster access to their required links and the computational time is lesser.*

1. INTRODUCTION

As the web is growing very rapidly, a user interacts very often and carries out many complex-task oriented operations over the net. The burst in the size and the richness of web is directly proportional to the variety and the complexity of task performed by user. Hence, the behaviour of a user is unpredictable and untraceable as in a user may perform many different search terms over small period of time or may perform many similar searches at different times. Query log generated by any user are hence no longer related to issuing simple navigational queries. One important step towards enabling services and features that can help users during their complex search quests online is the capability to identify and group related queries together. The main way of accessing the information over the internet is through keywords and queries using a search engine [1, 2]. A search engine has become a very important component of internet and they are broadly used for accessing any information over the net. However, a user decomposes the complex task-oriented operation into number of smaller and simplified queries, such as purchasing decision can be broken down into number of co-dependent steps over a period of time. For instance, a user may first search on possible choices of mobile phones depending upon budget, manufacturing company, features, comparison among few of them, etc. After deciding which mobile phone is to be purchased, the user may search for from where to buy to get better price and post purchase services, etc. Each step requires one or more queries, and each query results in one or more clicks on relevant pages. During their complex search online, one of the important step towards providing services and features that can help users is the capability to identify and group related queries together. This can be traced by using a new feature provided by any search engine which gives a user about their post navigational and task-oriented clicks and queries generally termed as "search histories"[3,4,5]. Query grouping can also assist other users by promoting task-level collaborative search. For instance, given a set of query groups created by expert users, we can select the ones that are highly relevant to the current user's query activity and recommend them to her. Explicit collaborative search can also be performed by allowing users in a trusted community to find share and merge relevant query groups to perform larger, long-term tasks on the Web. In this paper, we study the problem of organizing a user's search history into a set of *query groups* in an automated and dynamic fashion. Each query group is a collection of queries by the same user that are relevant to each other around a common informational need. We will achieve the above said by means by clustering algorithm.

2. AIM

Our main goal is to organize the user search histories into query groups [6], each containing one or more

related queries and their corresponding clicks. The main objective is to analyse the query log generated by the user and then use them for further operations like, generating query group, extracting semantics relations from query log, clustering them, query expansion, etc.

2.1. QUERY LOG

As user performs the search procedure over a period of time, a query log [7] is been generated and contains very important features. A query log contains a wealth of valuable knowledge about how web users interact with search engines as well as information about the interests and the preferences of those users. Extracting behavioural patterns from query log [8] is a key step towards improving the service provided by search engines and towards developing innovative web search paradigms.

2.2. QUERY GROUP AND DYNAMIC QUERY GROUP

A query group is an ordered list of queries, q*i*, together with the corresponding set of clicked URLs, clk*i* of q*i*. Each query group corresponds to an atomic information need that may require a small number of queries and clicks related to the same search goal. The process of identifying the query group is to first consider every query as a singleton query group, and then merge these singleton query groups in a iterative manner (in a k-means or algometric way[8]).

3. EXTRACTING SEMANTIC RELATION FROM QUERY LOGS

Most of the work on query similarity is related to query expansion or query clustering. One early technique proposed by [9] attempts to measure query similarity using the differences in the ordering of documents retrieved in the answers, which is not feasible in the current Web. Later, Defays [10], measured query similarity using the normalized set intersection of the top 200 documents in the answers for the queries. Again, this is not meaningful in the Web as the intersection for semantically similar queries that use different synonyms can and will be very small. R Sibson et al [11] proposed to cluster similar queries to recommend URLs to frequently asked queries of a search engine [7] they used four notions of query distance based on: (1) keywords or phrases of the query; (2) string matching of keywords; (3) common clicked URL's; and (4) the distance of the clicked documents in some pre-defined hierarchy. L. Kaufman and P. Rousseau [12] also proposed a query clustering technique based on distance notion (3). As the average number of words in queries is small (about two) and the number of clicks in the answer pages is also small, notions (1) an (2) generate very sparse distance matrices. Notion (4) needs concept taxonomy and the clicked documents to be classified into the taxonomy, which cannot be done in a large scale. Also (3) is sparse, but this sparsity can be diminished using large query logs. The query log is viewed as a set of transactions, with each transaction representing a session in which a single user submits a sequence of related queries in a time interval. The method shows good results, but two problems arise: it is difficult to determine sessions of queries belonging to the same search process; moreover the most interesting related queries, those submitted by different users, cannot be discovered, since the support of a rule increases only if its queries appear in the same query session (i.e. they are submitted by the same user.). [13-16] used the content of clicked Web pages to define a term-weight vector model for a query. They consider terms in the URLs clicked after a query. Each term is weighted according to the number of occurrences of the query and the number of clicks of the documents in which the term appears. Then the similarity of two queries is equivalent to the similarity of their vector representations, like the cosine distance function.

This notion of query similarity has several advantages. First, it is simple and easy to compute. On the other hand, it allows relating queries that happen to be worded differently but stem from the same topic, hence capturing semantic relationships among queries. Recently, Sahami and Heilman used a query similarity based on the snippets of the answers to the queries. However, they do not consider the feedback of the users (i.e. clicked pages) [17, 18].

4. QUERY CLUSTERING

Query clustering is a process used to find frequently searched or popular topics on a search engine. This process is crucial for search engines due to the short lengths of queries; approaches based on keywords are not suitable for query clustering. A new query clustering method that makes use of user logs which allow us to identify the documents the users have selected for a query. The similarity between two queries may be deduced from the common documents the users selected for them. A combination of both keywords and user logs is better than using either method alone [20, 21, 22]. Although the need for query clustering is relatively new, there have been extensive studies on document clustering, which is similar to query clustering. In this section,

we give a review of some approaches related to query clustering.

4.1 ADAPTIVE CLUSTERING ALGORITHM

The clustering problem can be described as a blind search on a collection of unlabelled data, where elements with similar features are grouped together in sets. K-means is a clustering algorithm that uses a fixed number (K) of clusters and looks for the best division of the dataset (through a predefined metric or distance) in this number of groups. Several clustering algorithms, such as K-means, have been improved using genetic algorithms [23].

A genetic algorithm is inspired by biological evolution [24]: the possible problem solutions are represented as individuals belonging to a population. The individuals are encoded using a set of chromosomes (called the genotype of the genome). Later these individuals are evolved, during a number of generations, following a survival/selection model where a fitness function is used to select the best individuals from each generation. Once the fittest individuals have been selected, the algorithm reproduces crosses and mutates them trying to obtain new individuals (chromosomes) with better features than their parents. The new offspring and, depending on the algorithm definition, their parents, will pass to the following generation. This kind of algorithms have been usually employed in optimization problems [10], where the fitness function tries to find the best solution among a population of possible solutions which are evolving. In other approaches, such as clustering, the encoding and optimization algorithm are used to look for the best set of groups that optimizes a particular feature of the data. In this new approach each chromosome is used to define a set of K clusters which represents a solution to the clustering problem. Clustering techniques can also be applied to different kinds of representations of the data collection like strings, numbers, records; text, images and semantic or categorical data [11-12].The proposed algorithms define a new definition for the density of data points throughout the data set. This new definition mentions the drawback of (2) which is based on kn-nearest neighbors density estimation [22-23].

EI-Kmeans calculates the density of each data point in the given data set, and then it sorts the data points in descending order according to their densities. The first densest point is selected as the first prototype. Then the list of sorted data points is investigated to get the next candidate prototypes. However, we shall take care that the selected prototype does not have a direct connectivity with the previously selected prototypes. We define two versions of EI-Kmeans algorithm. The first version takes *kn* the number of nearest neighbors, and the number of clusters *k*in the data set as input parameters. Then the sorted list of data points is investigated until the number of obtained prototypes reached the specified *k* then the algorithm is aborted. The second version takes *kn* the number of nearest neighbors as the only input parameter and obtains both the number of clusters and the prototypes in parallel. We also propose another version which is an improvement of our second version. However, this third version considers as a new clustering algorithm. This new algorithm is referred to as Efficient Data Clustering.

4.2 THE ALGORITHM

The second version of EI-Kmeans takes only one input parameter which is the *kn* number of nearest neighbors. Figure 1 presents the EI-Kmeans algorithm. If we mention our algorithm, then we mean the second version. Otherwise, it is stated explicitly. The Algorithm with adaptive K value works as follows:

1. A random population of community groups is generated.
2. The population evolves using a standard GA. The steps of the process are the same as was previously described in the previous section for the K-fixed algorithm.
3. The chromosome that has the best fitness function value is selected as final solution.

Although the genetic algorithm has not been changed, the new codification has modified how the genetic operators (crossover and mutation) are applied. This limitation is about the type of given data sets. In which the Kmeans algorithm has a problem of discovering clusters of different non-convex shapes, different sizes and densities. Thus we develop a new clustering algorithm to cope this limitation. This new algorithm is referred to as Efficient Data Clustering Algorithm (EDCA). EDCA is able to find clusters with different non-convex shapes, different sizes and densities. It also has a definition of noise and outliers. We benefit from our new definition of data points' densities to propose the EDCA.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

```
1   Begin initialize k_n, C={ }, G={ }, n,
    Lq = -1 ∀q ∈ {1,...,n}
2     for i = 1 to n
3       PD_i ← density(x_i)
4     end_for
5     G ← sort(PD)
6     C_1 ← G_1
7     L_G1 ← 1
8     for each data point in data set D
9       Np ← compute the number of points with
        radius ε
10      if Np < k_n
11        Mark this point as Noise
12      end_if
13    end_for
14    k = 1
15    j = 1
16    do j ← j + 1
17      if Gj is not marked as noise
18        for m= 1 to k
19          If there is a path between Gj and C_m
20            L_Gj ← m
21            break
22          end_if
23        end_for
24        if L_Gj == -1
25          Append Gj to C
26          k ← k + 1
27          L_Gj ← k
28        end_if
29      end_if
30    Until j = n
31  end
```

Figure 1: EDCA k-means Algorithm

Since we base our proposed algorithm on our new definition of density, it is strongly recommend for testing it in a situation where the data set contains clusters with different densities. In Figure 2(a) we generate a data set with 400 data points and it has three clusters with different densities and sizes. When we apply our algorithm, it finds out the true number of clusters which is three, thus our algorithm automatically generates the number of clusters in this data set. About the initial prototypes, the proposed algorithm identifies the three prototypes from the first run into satisfactory positions as shown in Figure 2(b) which yield a true clustering result of three clusters which is shown in Figure 2(c) However, when we apply the Kmeans algorithms and inject it with the number of clusters. The clustering result has only two clusters and the third one is an empty cluster as shown in Figure 2(e). Thus the effect of the bad initialized prototypes makes the clustering behaves very poorly.

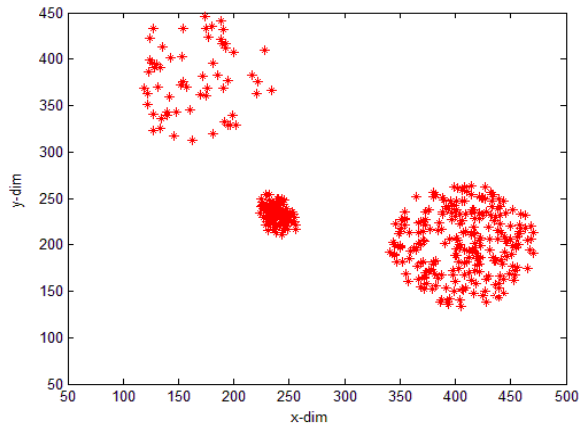

Figure: 2(a)

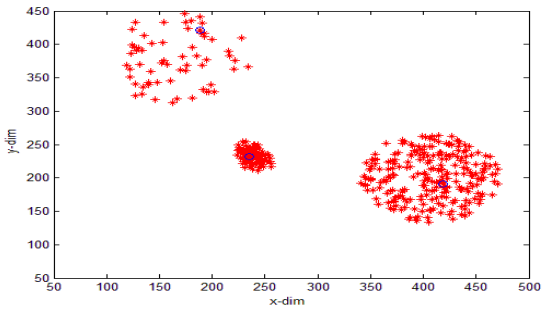

Figure: 2(b)

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

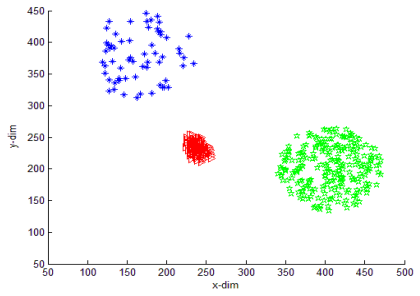

Figure: 2(c)

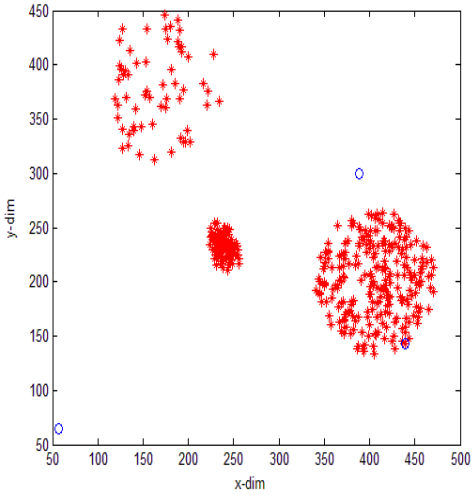

Figure: 2(d)

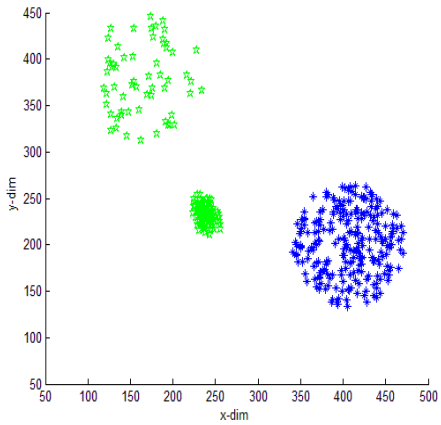

Figure: 2(e)

Fig. 2: (a) 400 data points with three clusters having different densities and sizes. (b) Blue 'o's are the initialized prototypes by EI-Kmeans. (c) Clustering result depends on the initialized prototypes in (b). (d) Blue 'o's are the best initialized prototypes by Kmeans after 5 times running of Kmeans. (e) Best clustering result depends on the initialized prototypes in (d).

5. CONCLUSION

We described the process of generation of query log by a user using a web search engine accessing any information over period of time. This query log can be grouped and then this query logs are used to extract semantics relations. Here we studied the basic concepts about organizing a user search histories for better performance. We showed how adaptive clustering algorithm works depending on the density of the queries requested for faster access of the user searched data.

REFERENCES

[1] M. Eirinaki and M. Vazirgiannis, 2003. Web Mining for Web Personalization, in ACM Transactions on Internet Technology (TOIT), vol. 3, no. 1 pp: 1-27. DOI: 10.1145/643477.643478

[2] B.Bahmani Firouzi, T. Niknam, and M. Nayeripour, Dec 2008. A New Evolutionary Algorithm for Cluster Analysis, Proc. of world Academy of Science, Engineering and Technology, vol.36.

[3] http://www.waset.org/journals/waset/v46/v46-100.pdf

[4] M. Al- Zoubi, A. Hudaib, A. Huneiti and B. Hammo, 2008. New Efficient Strategy to Accelerate k-Means Clustering Algorithm, American Journal of Applied Science, vol. 5, no. 9 pp: 1247-1250. DOI: 10.3844/ajassp.2008.1247.1250

[5] M. Celebi, 2009. Effective Initialization of K-means for Color Quantization, Proc. of the IEEE International Conference on Image Processing, pp: 1649-1652. DOI: 10.1.1.151.5281

[6] M. Borodovsky and J. McIninch, 1993. Recognition of genes in DNA sequence with ambiguities, Biosystems, vol. 30, issues 1-3, pp: 161-171. DOI: 10.1016/0303-2647(93)90068-N

[7] J. Bezdek and N. Pal, 1992. Fuzzy Models for Pattern Recognition, IEEE press (New York, NY, USA). http://www.amazon.com/Fuzzy-Models-Pattern-Recognition-Structures/dp/0780304225

[8] J. Bezdek, 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press (New York, NY, USA). http://www.Amazon.com/Recognition- Objective-Function Algorithms-Applications/dp/0306406713.

[9] G. Gan, Ch. Ma, and J. Wu, 2007. Data

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Clustering: Theory, Algorithms, and Applications, ASA-SIAM series on statistics and Applied Probability, SIAM. DOI: 10.1111/j.1751-5823.2007.000392.x

[10] D. Defays, 1977. An Efficient Algorithm for A Complete Link Method, The Computer Journal, vol. 20, pp: 364-366. DOI: 10.1093/comjnl/20.4.364

[11] R. Sibson, 1973. SLINK: an Optimally Efficient Algorithm for the Single Link Cluster Method, The Computer Journal, vol. 16, No. 1, pp: 30-34. DOI: 10.1093/comjnl/16.1.30

[12] L. Kaufman, and P. Rousseau, 1990. Finding Groups in Data: An Introduction to Cluster Analysis, (John Wiley & Sons). DOI: 10.1002/9780470316801

[13] J. Macqueen, 1967. Some Methods for Classification and Analysis of Multivariate Observations, 5th Berkeley Symp. Math. Statist. Prob., vol. 1, pp: 281-297. http://digitalassets.lib.berkeley.edu/math/ucb/text/math_s5_v1_article-17.pdf

[14] J. Wen, J. Mie, and H. Zhang. Clustering user queries of a search engine. WWW'01.

[15] H. -J. Zeng, Q. -C. He, Z. Chen, W. -Y. Ma, and J. Ma.Learning To Cluster Search Results. SIGIR'04.

[16] Nino Boccara, *Modeling Complex Systems*, Springer, 1 edition, 2003.

[17] Daniel Fenn, Omer Suleman, Janet Efstathiou, and Neil Johnson, "How does europe make its mind up? connections, cliques, and compatibility between countries in the euro vision song contest," *Physical A: Statistical Mechanics and its Applications*, vol. 360, no. 2, pp. 576–598, February 2005.

[18] Marie Phillips., "It's time to make our minds up on Europe.," *The Observer*, , no. Friday 12, March 2004.

[19] EBU, "http://www.ebu.ch/," October 2010.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[21] G. Nathiya, S. C. Punitha, and M. Punithavalli, "An analytical study on behavior of clusters using k means, em and k* means algorithm," *CoRR*, vol. abs/1004.1743, 2010.

[22] Amir Ahmad and Lipika Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data and Knowledge Engineering*, vol.63, no. 2, pp. 503 – 527, 2007.

[23] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, 1996. A Density-based Algorithm for Discovering Clusters in Large Spatial Data sets with Noise, 2nd International Conference on Knowledge Discovery and Data Mining, pp: 226-231. DOI: 10.1.1.121.9220

[24] E. Forgy, 1965. Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classification, Biometrics, vol.21. http://www.citeulike.org/user/ dgrianna/ article/33824

About Authors

Mrs.Soujanya Naidu final year M.tech Student in Avanthi Institute of Engineering and Technology (JNTUK), Cherukupally, VizianagaramDist, Andhra Pradesh, India.

Mr.K.Ravindra working as an Asst.prof in the department of CSE at at Avanthi Institute of Engineering and Technology, Cherukupally, VizianagaramDist.

Y. Ramesh Kumar obtained his M.Sc (Computer Science) degree from Andhra University. Later he obtained his M.tech (CST) degree from Andhra University. Presently he is working as Associate Professor and Head of the Department (CSE) at Avanthi Institute of Engineering and Technology, Cherukupally, VizianagaramDist. He has guided more than 60 students of Bachelor degree, 40 Students of Master degree in Computer Science and Engineering in their major projects. His Research interest includes Ontology-based Information Extraction based on Web search and mining and ware housing.

Mr.PavanRaju working as Asst.prof in the department of CSE in Shri Vishnu Engineering College for Women, Bhimavaram, Andhra Pradesh, India.