Web Page Clustering by Combining Dense Units

Morteza Haghir Chehreghani[1], Hassan Abolhassani[1] and Mostafa Haghir Chehreghani[2]
[1]Department of CE, Sharif University of Technology, Tehran, IRAN
{haghir, abolhassani}@ce.sharif.edu
[2]Department of ECE, University of Tehran, Tehran, Iran, m.haghir@ece.ut.ac.ir

***Abstract* — One of the most important approaches of extracting knowledge from the web is to cluster the web data. In this paper a novel method for clustering the web pages is presented which at first finds the dense units using K-Means method and then joins these units for constructing final suitable clusters. The method also is extended for hierarchical clustering. The experimental results show the high quality of both flat and hierarchical clusters.**

***Index Terms* — Web Mining, Web clustering, hierarchical clusters, K-Means.**

I. Introduction

Data clustering is an important pre-processing task in information mining that can provide useful results. Clustering involves dividing a set of data objects into non-specified groups. The data objects within each group should exhibit a large degree of similarity while the similarity among different clusters should be minimized. Clustering takes an important role on the web; it can be used for enhancing search results, enhancing web crawling, and organizing and presenting domain knowledge. Up to now various methods have been introduced for clustering the web, more of them use techniques such as link analysis [3], [7,9], content mining [12] and combination of them [15,22]. In an overall categorization, web data clustering algorithms can be divided into two categories: Hierarchal and Partitioning algorithms. Hierarchical algorithms cluster the web pages in a multi level and tree-like structure, while partitioning methods cluster the web data in a single and flat level [4,8]. In general only algorithms such as graph-based [6,14] and tree-based [24] methods and methods that find a set of centroids (K-Means) [21], are applied for clustering the web data [23]. Although hierarchical methods are often said to have better results from quality respective, but usually they do not provide the repetition and the reassignment of pages, which may have been poorly clustered in the early steps [8]. Moreover, the time complexity of the hierarchical methods is often quadratic [20].

Partitioning methods divide a collection of documents into a set of groups, so as to maximize a pre-defined fitness measure. It seems that the recent partitioning clustering methods are well suited for clustering a large document dataset due to their relatively low computational requirements [20]. The best known partitioning algorithm is *K-Means* [13] that in a simple form selects K points as cluster centers and assigns each data point to the nearest center. The updating and reassigning process can be kept until a convergence criterion is met. This algorithm can be performed on a large data set almost in linear time complexity. K-Means has some problems that make it inappropriate for many applications. In section 2 we will explain some of its drawbacks.

In this paper we propose a new method which at first finds dense units of the entire data set and then combines them for creating final clusters. The combination process is done in either flat or hierarchical modes and so can generate flat or hierarchical clusters. While creating dense units, a center of gravity is determined for each of them. In the combination step, these centers are used as representatives of dense units in average distance combination. The remaining of the paper is organized as follows: In section 2, some primary definitions and concepts are introduced. Section 3 contains a complete description of the proposed method and its time complexity analysis. The experimental results of the method are evaluated in section 4 and finally a conclusion is given in section 5.

II. Primary Definitions and Concepts

A. Document Representation and Similarity Measures

In the most data mining algorithms, each data point (such as a document) is represented as a vector $V = \{v_1, v_2, \ldots, v_n\}$, where v_i is the weight of dimension i in vector space. About text documents, this is known as Vector Space Model [2] that each weight v_i represents the weight of associated term of the document. The most common term weighting approach is the combination of Term Frequency with Inverse Document Frequency (*TF-IDF*) [8]. In this approach the weight of term i in document j is defined as (1).

$$w_{ji} = tf_{ji} \times idf_{ji} = tf_{ji} \times \log_2(n / df_{ji}) \quad (1)$$

That tf_{ji} is the number of occurrences of term i in the document j; df_{ij} is the total term frequency in data set and n is the number of documents. The vector space model gives us a good opportunity for defining various metrics of similarity between two documents. The most common similarity metrics are Minkowski distances [1] and cosine measure [18]. Minkowski distances

computes the distance of documents m_p and m_j by (2) (for $n=2$ it is converted to Euclidean distance).

$$D_n(m_p, m_j) = \left(\sum_{i=1}^{d_m} | m_{i,p} - m_{i,j} |^n \right)^{1/n} \quad (2)$$

Cosine measure is defined by (3), where $m_p^t m_j$ is the inner product of two vectors. Both metrics are widely used in text clustering literatures. However it seems that in the cases which the number of dimensions of two vectors differs considerably, the cosine is more useful.

$$\cos(m_p, m_j) = \frac{m_p^t m_j}{| m_p || m_j |} \quad (3)$$

B. Description of K-Means Algorithm

K-Means chooses K points as centroids and assigns each web page to the nearest one. Then the centroids are updated. The assigning and updating process is kept until a convergence criterion is met. Since clustering is a NP-Complete problem, so this algorithm (like other algorithms) is a heuristic solution that finds the nearest local optimum of the search space. The local optimum and therefore the quality of the results are dependent to the initial selection of the centers. Different steps of K-Means are shown in Figure 1.

1. Initial Step :
Select *K* initial points by random;
2. Repeat
for each data point *dp* **in** data set:
find the nearest centroid to *dp*;
assign *dp* to the nearest centroid;
for each cluster *c* **in** cluster set:
update the centroid *c*;
3. until (convergence is confirmed**)**

Fig. 1. The K-Means algorithm.

In the algorithm of Figure 1, the criterion used for distinguishing the convergence and so finalizing the process is defined as (4). This measure is known as minimizing Sum of Squared Errors.

$$E(A, C) = \sum_{i=1}^{K} \sum_{j=1}^{n} (a_{ij}) D_{ij}^2 (c_i, x_j) \quad (4)$$

In (4), A is the assignment matrix (shows which web page has been assigned to which centroid) and $C = \{c_i\}$ is the set of K centroids. Assignment matrix includes values of 0 and 1 (in crisp clustering) and is obtained by assigning each data point to the corresponding centroid. In the algorithm of Figure 1, updating the centroids is done according to (5).

$$c_i = \frac{\sum_{j=1}^{n} (a_{ij}) x_j}{\sum_{j=1}^{n} a_{ij}}, \qquad 1 \le i \le K \quad (5)$$

K-Means has several drawbacks. The main drawback is the sensitivity of result clusters to the initial centroids; so that it may convergence to the local optima [19] that is not acceptable. In fact, K-means is an optimization solution that finds the local optimum placed in the vicinity of the initial solution. But the same initial cluster centroids in a dataset will always generate the same cluster results and so repeating the algorithm for achieving better results must be done with different initial centroids.

III. Proposed Algorithm

In this section, we introduce our algorithm and analyze its time complexity. The algorithm includes two steps: constructing dense units and combining them for creating final clusters. The combination process can be done either in flat or in hierarchical mode. In following we describe each step with enough details.

A. Constructing Dense Units

The algorithm at first selects a constant value for centroids denoted by C. It is a large value that is related to the number of all data points. If n shows the number of data points, then C is selected as $\frac{n}{MinPts}$. Where *MinPts* shows the minimum required data points that must reside in a small region so that the region can be considered as dense. In the next step, each data point is assigned to the centroids and after assignment; the centroids are updated according to (5). The process is repeated until the convergence criterion is met (similar to typical K-Means). The convergence is met when the position of centroids do not change more after. Therefore at the end of this step we will have numerous spherical dense units which each of them has a center point representing the distribution of density. The process has been depicted in Figure 2.

1. Initial Step :
Select *MinPts*; Select *C* as $\frac{n}{MinPts}$;
2. Repeat
for each data point *dp* **in** data set:
assign *dp* to the nearest center;
for each unit *u*:
update the center of *u*;
3. until (convergence is confirmed**)**

Fig. 2. Finding dense units.

B. Flat combination of Dense Units

In the next step we combine the dense units for constructing final clusters. Combining these dense units in an optimum way so that the final clusters would be the best combination is a NP-Complete problem. This is proofed in **Theorem 1**.

Theorem 1- Finding the best combination of dense units that yields best clusters is an NP-Complete problem.

Proof- We know that data clustering is a NP-Complete problem [17]. We can construct a hyper graph with nodes representing the dense units. Each node has links to two kinds of nodes: nodes that correspond to the data

points inside the dense unit, and conterminous nodes (w.r.t distance between the centers of dense units). We know that each hyper node (dense unit) has a limited number of data points. So the number of hyper nodes will be itself from *O(n)*. Therefore combination of these units can be simulated as a clustering problem and so finding the optimum solution is a NP-Complete problem. ▪

So, because of complexity considerations, we should take a heuristic method which can perform combination with a reasonable time complexity. For this we propose two alternatives:

1. Define a threshold and every two dense units which have the similarly more than this threshold, join together and construct a unique dense unit. So the resultant clusters will be dependent on the order of traversing the dense units. So the combination may lead to a local optimum while for achieving better results several runs may be needed.
2. With respect to the drawbacks of the first approach, we can follow a more heuristic method. We follow a process similar to the *average linkage* hierarchical clustering [10] but produce the clusters in flat format. The method repeatedly combines smaller units and produces bigger ones. The criterion used for combining the smaller groups is average distance between data inside two units. The created clusters have high quality, but finding the average distance between different dense units requires high time complexity. Our process has a considerable preference for applying this method: In the step of constructing spherical dense units, the center of each unit is gained while assigning and updating process. Therefore against of other linkage methods, here there is no need for finding average distance between base clusters; comparing the distance between units can be done only by comparing the distance between their centers.

Between these two methods, we select the second one. So after constructing the dense units, we follow the process of Figure 3. In the algorithm of Figure 3, finding new center is done according to (6).

1. Initialize
Select *K* as the desired number of clusters;
2. Repeat
Find the two nearest units;
Join them as a unique unit;
Find the center of new unit;
3. until (there are only *K* units**)**

Fig. 3. The Combination Step.

$$NewCenter = \frac{N_i}{N_i + N_j} \times C_i + \frac{N_j}{N_i + N_j} \times C_j \qquad (6)$$

That N_i and N_j shows the number of members of units *i* and *j*, respectively. So, the new center is weighted average of two centers. The joining process is kept until the desired clusters are obtained.

C. Availing Hierarchical Clustering

In this section we improve the method for hierarchical clustering. For this in each combination step, we must examine whether the combination yields a compound unit in the same unit or a new unit must be constructed in higher level.

Using a constant threshold as agglomeration level can not be useful. However small low level clusters should use smaller thresholds while big high level clusters should have greater thresholds.

Since the clusters are constructed from smaller ones to bigger ones and we want about the joining thresholds to act in this manner, so we establish a relationship between the thresholds and the joint units. We define a measure and try to optimize it during the combination process. The measure is defined in relation (7).

$$\frac{Max\,(Clustered\ Dis\tan ces\ of\ C_i\ and\ C_j)}{Dis\tan ce\ between\ two\ centers} ?\, M \qquad (7)$$

M is a constant value which is determined by user. Therefore, according to the result of equation (7) we have two situations (we name the left part of (7) as *L*):

1. If $L \geq M$, then these two dense units will join with each other and will construct a unique unit in the same level. The center of the new unit is updated by the weighted averaging of relation (6). Also, other parameters of the new unit are calculated.
2. If $L < M$, then a new unit will be constructed in a higher level and the two units will be its children. The new unit contains combined units with their updated parameters.

This method gives us a heuristic way for constructing the hierarchical clusters.

D. Analysis of Time Complexity of the Method

K-Means requires *O(n)* time of complexity for finding a limited number of cluster centroids. But since the number of the centroids is itself from *O(n)*, so the total time complexity of the first step will be equal to $O(n^2)$. Second step (combination step) can be implemented with the time complexity of $O(n^2)$ whether in flat or in hierarchical mode. This process can be done as follows: at first we consider an array with the size of *2m-1* (*m* is the number of dense units). We store in each cell of array the nearest center to the center indexed by the array's id as well as the corresponding distance. In each combination the shortest distance is found with *O(m)*. After finding the shortest distance, the array is updated for the newly created unit which this can be done with *O(m)* time of complexity, too. Therefore the total time complexity for this part will be from $O(m^2)$ that it is itself equal to $O(n^2)$.

IV. EXPERIMENTAL RESULTS

For evaluation of proposed method we have used two data sets: One data set is collected from the set of *REUTERS* documents and another one is collected from *Politics* area containing 425 web pages that are selected randomly about the some topics of Politics domain. The properties of the data sets are shown in Table 1. For stemming we have used the Porter algorithm and after calculating the *TF/IDF* values of vectors, they were normalized.

TABLE 1. Examined Data Sets.

Data Set	Size	No.of Clusters	Selected *MinPts*
REUTERS	300	8	10
Politics	425	10	15

There are several methods for evaluation of clustering results most two well known methods among them are entropy-based methods ([16,20]) and F-measure [11]. Because of more generality of F-measure for evaluation of web searches, we use this measure for evaluation of our results. F-measure combines two measures *precision* and *recall* and evaluates whether the clustering can remove the noise pages and generate clusters with high quality. If P and R show *Precision* and *Recall* respectively, this measure is defined by (8) where precision and recall are obtained by (9). In these formulas n_j shows the size of cluster j, g_i shows the size of class i and $N(i,j)$ shows the number of pages of class i in cluster j.

$$F(i,j)=\frac{2(P(i,j)*R(i,j))}{(P(i,j)+R(i,j)}, \; F=\sum_i \frac{g_i}{n} \max_j \{F(i,j)\} \tag{8}$$

$$P(i,j)=N(i,j)/n_j, \; R(i,j)=N(i,j)/g_i \tag{9}$$

TABLE 2- Comparison of methods for the first data set.

	K-Means		Proposed Method	
CENTER ID	N	F	N	F
1	26	0.597333	36	0.714125
2	41	0.634015	32	0.747866
3	35	0.627460	28	0.757013
4	12	0.594843	43	0.794968
5	67	0.716301	51	0.796875
6	18	0.688428	23	0.778846
7	76	0.531215	60	0.699580
8	25	0.720827	27	0.798875

For more comparisons we have implemented the *K-Means* and have evaluated its results (since the results of the *K-Means* are dependent on the initial centroids, so we have run the *K-Means* five times with different initial centers and have selected the best result). The evaluation results are shown in Tables 2 and 3. About the first data set, the total F-Measure is *0.7570627* for the proposed algorithm while its value is *0.6313383* for *K-Means* method. The difference between these two values shows the quality of the proposed algorithm. About the second data set, the total F-Measure value of the *K-Means* and the proposed method are *0.6270834* and *0.7479225*, respectively. These values also show the efficiency of the proposed method.

TABLE 3- Comparison of methods for the second data set.

	K-Means		Proposed Method	
CENTERID	N	F	N	F
1	42	0.629365	46	0.71875
2	17	0.658666	31	0.762352
3	84	0.599508	56	0.708732
4	46	0.658646	52	0.794716
5	51	0.662205	61	0.782420
6	28	0.674705	42	0.749466
7	54	0.584957	29	0.699580
8	38	0.679852	38	0.798875
9	21	0.555789	47	0.713194
10	44	0.649384	23	0.729863

In the next step we examine the hierarchical clustering. For this we select M equal to *0.7* and construct the hierarchy. The resultant hierarchy is depicted in Figure 4. The numbers existing in each cluster depict their number of members and we assign a unique ID for each cluster in a left to right BFS order.

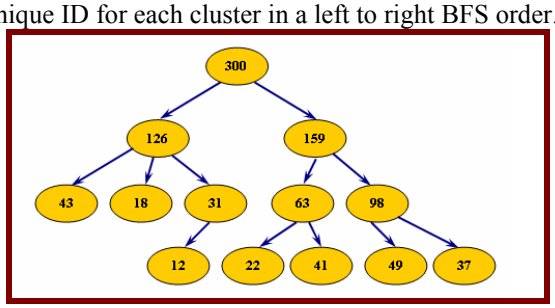

Fig. 4. The obtained hierarchy.

Evaluation methods such as F-measure and entropy are developed for evaluation of flat clusters (or lowest level of hierarchical clusters). So here we use an enhanced version for hierarchical clustering developed in [5]. According to [5] the F-measure values of different clusters follow from a button-up process: The precision of a parent cluster is calculated from the precision of its children according to (10). In (10) P_R shows the precision of the cluster members which do not belong to any other clusters. Since for computing the recall we must calculate the R_R (is defined in similar with P_R) by traversing all web pages, so a more simple way can be done by calculating recall without considering pre-calculated values for sub-clusters [5].

$$P_C=\sum_{\forall k\in C.children}\frac{n_k}{n}P_k+\frac{n-\sum_{\forall k\in C.children} n_k}{n}P_R \tag{10}$$

However using the adapted measure, the results are brought in Table 4. We have applied the hierarchical method only on the first data set. In this case the

obtained total F-Measure is *0.72934* which is acceptable for a hierarchical clustering.

TABLE 4- The results of hierarchical algorithm.

CENTER ID	N	P	R	F
1	293	0.64	0.92	0.7548
2	126	0.63	0.83	0.7163
3	159	0.65	0.91	0.7583
4	43	0.78	0.71	0.7433
5	18	0.83	0.63	0.7163
6	31	0.73	0.78	0.7541
7	63	0.64	0.81	0.7150
8	98	0.57	0.82	0.6725
9	12	0.81	0.6	0.6893
10	22	0.78	0.66	0.715
11	41	0.71	0.72	0.7149
12	49	0.63	0.73	0.6763
13	37	0.73	0.67	0.6987

V. CONCLUSION

In this paper, a new method for clustering the web documents is introduced. In fact, the proposed method tries to overcome some drawbacks of the well known *K-Means* algorithm which finds only the nearest local optimum. The method tries to improve the local optimum toward global optimum. This method, at first, finds dense units by bombarding the data space with too many centroids. Therefore at the end, we have dense units with a representative point (center) for each of them. Having these center points, we can apply average linkage clustering algorithm on the dense units and construct the final clusters by joining them. The method has been extended for hierarchical clustering. For this purpose a heuristic measure has been defined and optimized in each cluster. If a new joint cluster does not satisfy the measure, the cluster is created in higher level. The method has been evaluated by well known F-Measure and its improved version for hierarchical clusters. According to the evaluation results, both of the flat and hierarchical clusters have suitable quality.

REFERENCES

[1] Cios K., Pedrycs W., Swiniarski R., *"Data Mining-Methods for Knowledge Discovery"*, Kluwer Academic Publishers, 1998.

[2] Everitt, B., *"Cluster Analysis"*, 2nd Edition. Halsted Press, New York, 1980.

[3] Getoor, L., *"Link Mining: A New Data Mining Challenge"*, ACM SIGKDD Explorations Newsletter, Vol. 5, pp. 84–89, 2003.

[4] Grira, N., Crucianu, M., Boujemaa, N., *"Unsupervised and Semi-supervised Clustering: a Brief Survey"*, 7th ACM SIGMM international workshop on Multimedia information retrieval, pp. 9-16, 2005.

[5] Haghir Chehreghani, Morteza, Abolhassani, H., and Haghir Chehreghani, Mostafa, *"Attaining higher quality for Density Based Algorithms"*, Web Reasoning and Rule Systems, LNCS (Springer), pp. 329-338, 2007.

[6] Hea, X., and Zhaa, H., Ding, C.H.Q., Simon, H.D., *"Web document clustering using hyperlink structures"*, Computational Statistics & Data Analysis, Vol. 41, pp.19-45, 2002.

[7] Henzinger, M., *"Hyperlink analysis on the world wide web"*, Sixteenth ACM conference on Hypertext and hypermedia HYPERTEXT '05, pp. 1–3, 2005.

[8] Jain, A. K., Murty, M. N., and Flynn, P. J., *"Data Clustering: A Review"*, ACM Computing Surveys (CSUR), Vol. 31, Issue 3, pp. 264–323, 1999.

[9] Kleinberg, M., *"Authoritative sources in a hyperlinked environment"*, Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, 1998.

[10] Koller, D., Sahami, M., *"Hierarchically classifying documents using very few words"*, 14th International Conference on Machine Learning, pp. 170–178, 1997.

[11] Larsen, B., and Aone, C., *"Fast and effective text mining using linear-time document clustering"*, SIGKDD'99, CA, pp. 16–22, 1999.

[12] Liu, B., Chang, K.C.-C., *"Editorial: Special Issue on Web Content Mining"*, ACM SIGKDD Explorations Newsletter, Vol. 6, Issue 2, pp. 1-4, 2004.

[13] McQueen, J., *"Some methods for classification and analysis of multivariate observations"*, Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281-297, 1967.

[14] Neville, J., Adler, M., Jensen, D., *"Clustering Relational Data Using Attribute and Link Information"*, 18th International Conference on Artificial Intelligence, 2003.

[15] Nurminen, M., Honkaranta, A., Kärkkäinen, T., *"ExtMiner: Combining Multiple Ranking and Clustering Algorithms for Structured Document Retrieval"*, Sixteenth Workshop on Database and Expert Systems Applications, pp. 1036-1040, 2005.

[16] Quinlan, R.J., *"C4.5: Programs for Machine Learning"*, Morgan Kaufmann, 1993.

[17] Schenker, A., Last, A., Bunke, H., Kandel, A., *"A Comparison of Two Novel Algorithms for Clustering Web Documents"*, 2nd International Workshop on Web Document Analysis, pp. 71-74, 2004.

[18] Salton, G., Buckley, C., *"Term-weighting approaches in automatic text retrieval"*. Information Processing and Management, 24 (5): pp. 513-523, 1988.

[19] Selim, S.Z., Ismail, M.A. *"K-means type algorithms: A generalized convergence theorem and characterization of local optimality"*, IEEE Trans. Pattern Anal. Mach. Intell. 6, pp. 81–87, 1984.

[20] Steinbach, M, Karypis, G., Kumar, V., *"A Comparison of Document Clustering techniques"*, KDD'2000, Technical report of University of Minnesota, 2000.

[21] Wang, Y., and Kitsuregawa, M., *"Enhancing Content-Link Coupled Web Page Clustering and It's Evaluation"*, DEWS' 2004, 2004.

[22] Weiss, R., Vélez, B., and Sheldon, M.A., *"HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering"*, 7th ACM conference on Hypertext, USA, pp. 180-193, 1996.

[23] Yao, Z., and Choi, B., *"Bidirectional Hierarchical Clustering for Web Mining"*, IEEE/WIC International Conference on Web Intelligence WI, pp. 620-624, 2003.

[24] Zamir,O., Etzioni, O., *"Web document clustering: A feasibility demonstration"*, 19th International ACM SIGIR Conference on Research and Development of Information Retrieval, pp.46-54, Australia, 1998.