A Scalable Hierarchical Fuzzy Clustering Algorithm for Text Mining

M.E.S. Mendes Rodrigues and L. Sacks
Department of Electronic and Electrical Engineering
University College London
Torrington Place, London, WC1E 7JE, United Kingdom
e-mail: {mmendes, lsacks}@ee.ucl.ac.uk

Abstract*: Clustering techniques are generally applied for finding unobvious relations and structures in data sets. In this paper, we propose a novel scalable hierarchical fuzzy clustering algorithm to discover relationships between information resources based on their textual content, as well as to represent knowledge through the association of topics covered by those resources. The algorithm addresses the important problem of defining a suitable number of clusters for appropriately capturing all the topics of the knowledge domain. In particular, the sought granularity level defines the number of clusters. Furthermore, the algorithm exploits the concept of asymmetric similarity to link clusters hierarchically and to form a topic hierarchy.*

Keywords*: Hierarchical fuzzy clustering, hyper-spherical fuzzy c-means, asymmetric similarity, topic hierarchy, text mining, knowledge representation.*

1. Introduction

Document clustering has been widely applied in the field of information retrieval for improving search and retrieval efficiency [15]. Furthermore, document clustering has also been applied as a tool for browsing large document collections [3, 6] and as a post-retrieval tool for organizing Web search results into meaningful groups [16, 18]. We have recently applied document clustering to dynamically discover content relationships in e-Learning material based on document metadata descriptions [9]. Although our motivation for applying clustering techniques is related with enhancing the navigation of e-Learning material, our main focus is on the discovery and representation of unobvious or unfamiliar knowledge about a domain rather than on facilitating the access to specific information resources through a set of document clusters. In this paper, we propose a novel scalable hierarchical fuzzy clustering algorithm for document clustering that was motivated by the e-Learning context, but that has wider applicability as a generic text mining tool.

The subsequent sections of this paper are organized as follows. In section 2, the argument for using fuzzy clustering techniques is presented and the issue of finding the optimum number of clusters is addressed. In section 3, a detailed description of the new Hierarchical Hyper-spherical Fuzzy c-Means algorithm is presented. In section 4, the experimental work is described and the results are analysed. Finally, section 5 contains the conclusions.

2. Fuzzy clustering of text documents

Topics that characterise a given knowledge domain are somehow associated with each other. Those topics may also be related to topics of other domains. Hence, documents may contain information that is relevant to different domains to some degree. With fuzzy clustering methods documents are attributed to several clusters simultaneously and thus, useful relationships between domains may be uncovered, which would otherwise be neglected by hard clustering methods.

We have recently modified the Fuzzy c-Means (FCM) algorithm for clustering text documents based on the cosine similarity coefficient rather than on the Euclidean distance [11]. The modified algorithm works with normalised k-dimensional data vectors that lie in hyper-sphere of unit radius and hence has been named Hyper-spherical Fuzzy c-Means (H-FCM). Our experiments with the H-FCM algorithm for document clustering have shown that it outperforms the original FCM algorithm as well as the hard k-Means algorithm [10, 11].

The objective function the H-FCM minimises is similar to the FCM one [2], the difference being the replacement of the squared norm by a dissimilarity function $D_{i\alpha}$:

$$J_m(U,V)=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^{m}D_{i\alpha}=\sum_{i=1}^{N}\sum_{\alpha=1}^{c}u_{\alpha i}{}^{m}(1-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j})\cdot \tag{1}$$

The cosine coefficient [1] ranges in the unit interval and when data vectors are normalised to unit-length it is equivalent to the inner product. The dissimilarity function $D_{i\alpha}$ in equation (1) consists of a simple transformation of the cosine similarity coefficient, *i.e.* $D_{i\alpha} = 1- S_{i\alpha}$.

$$u_{\alpha i}=\left[\sum_{\beta=1}^{c}\left(\frac{D_{i\alpha}}{D_{i\beta}}\right)^{\frac{1}{(m-1)}}\right]^{-1}=\left[\sum_{\beta=1}^{c}\left(\frac{1-\sum_{j=1}^{k}x_{ij}\cdot v_{\alpha j}}{1-\sum_{j=1}^{k}x_{ij}\cdot v_{\beta j}}\right)^{\frac{1}{(m-1)}}\right]^{-1} \tag{2}$$

$$v_{\alpha}=\sum_{i=1}^{N}u_{\alpha i}{}^{m}x_i\cdot\left[\sum_{j=1}^{k}\left(\sum_{i=1}^{N}u_{\alpha i}{}^{m}x_{ij}\right)^{2}\right]^{-1/2} \tag{3}$$

The update expression for the membership of data element x_i in cluster α, denoted as $u_{\alpha i}$ and shown in equation (2), is also similar to the original FCM expression since the calculation of $D_{i\alpha}$ does not depend explicitly on $u_{\alpha i}$. However, a new update expression for the cluster centroid v_{α}, shown in equation (3), had to be developed. Like the original algorithm, H-FCM runs iteratively until a local minimum of the objective function is found or the maximum number of iterations is reached.

2.1 Finding the optimum number of clusters

The H-FCM algorithm requires the selection of the number of clusters c. However, in most clustering applications the optimum c is not known *a priori*. A typical approach to find the best c is to run the clustering algorithm for a range of c values and then apply validity measures to determine which c leads to the best partition of the data set [5]. The validity of individual clusters is usually evaluated based on their compactness and density.

In low-dimensional spaces it is acceptable to assume that valid clusters are compact, dense and well separated from each other. However, text documents are typically represented as high-dimensional sparse vectors. In such problem space, the similarity between documents and cluster centroids is generally low and hence, compact clusters are not expected. Therefore, the approach mentioned above for finding the optimum c is inappropriate.

A question that arises is how the H-FCM algorithm is able to discover meaningful document clusters considering such low similarity patterns. As observed for the hard k-Means algorithm [4], the good performance of the H-FCM is justified by the fact that documents within a given cluster are always more similar to the corresponding centroid than documents outside that cluster, regardless of the number of clusters that has been selected. We believe that in the high-dimensional document space the issue of finding the optimum number of clusters is not so relevant. The choice of c should rather address the desired granularity level, since the higher the number of clusters the more specific will be the topics covered by the documents in those clusters.

3. Hierarchical Hyper-spherical Fuzzy c-Means algorithm (H^2-FCM)

In view of the previous analysis regarding the selection of c, we consider applying the H-FCM algorithm for an over-specified number of clusters and creating a hierarchical organisation of those clusters based on parent-child type relationships between cluster centroid vectors. As a result, we have developed the Hierarchical Hyper-spherical Fuzzy c-Means algorithm (H^2-FCM). Scalability issues as well as the fact that a hierarchical clustering structure is far more intuitive to browse than a non-hierarchical one, have provided the motivation for the new algorithm.

3.1 Asymmetric similarity measure

The new algorithm explores the concept of asymmetry to define parent-child type relationships between H-FCM cluster centroid vectors with the purpose of building a cluster hierarchy. By definition a parent-child relationship embraces the concept of inheritance: the child inherits all the attributes from its parents, while adding some new attributes.

In the present case, cluster centroids are high-dimensional vectors of unit-length containing k term weights. Thus, a child vector should contain all the terms from its parent vector and some additional terms. However, it is likely that in such a high-dimensional space a candidate parent contains some terms with low weights, which are not present in its candidate child. Nevertheless, following Tversky's model of similarity [17] the notion of inheritance can be relaxed considering that a child cluster should be less similar to its parent than the opposite. We apply the asymmetric similarity measure defined in equation (4) to link cluster centroids hierarchically. If v_α is more a child of v_β than the opposite, then $S(v_\alpha, v_\beta) < S(v_\beta, v_\alpha)$.

$$S(v_\alpha, v_\beta) = \sum_{j=1}^{k} \min(v_{\alpha j}, v_{\beta j}) \Big/ \sum_{j=1}^{k} v_{\alpha j} \tag{4}$$

3.2 Description of the H^2-FCM algorithm

The new H^2-FCM algorithm consists of three main stages. It starts by invoking the H-FCM algorithm for obtaining a sufficiently high number of document clusters. Then, it takes each pair of clusters and calculates their asymmetric similarity and finally, it links the cluster centroids hierarchically using a top-to-bottom approach to obtain a cluster hierarchy. The algorithm is summarised as follows:

Step 1. Given N documents indexed with k terms, apply the H-FCM algorithm to obtain a fuzzy partition of the document set into an over-specified number of clusters c. To avoid the inclusion in the hierarchy of clusters with very few documents define the minimum size of the cluster t_{ND}. While there are K>0 clusters with less than t_{ND} documents (for a given membership threshold α), re-apply H-FCM for $c = c\text{-}K$ clusters. Return the cluster centroids V $(c\times k)$ and the partition matrix U $(c\times N)$.

Step 2. Compute the asymmetric similarity between each pair of cluster centroids using equation (4) and specify the parent-child similarity threshold t_{PCS}.

Step 3. Define V_H and V_F as the set of cluster centroids already assigned and not yet assigned to the hierarchy, respectively. Initially, $V_H = \varnothing$ and $|V_F| = c$.

Step 4. While $V_F \neq \varnothing$ repeat the following steps:

Step 4.1 Select a candidate vector $v_\alpha \in V_F$ such that $\exists v_\beta \in V_F : S(v_\alpha, v_\beta) = \max[S(v_\iota, v_\varphi)]$, $\forall v_\iota, v_\varphi \in V_F$

Step 4.2 If there is more than one candidate, temporarily set $S(v_\alpha, v_\beta) = 0$ and repeat the selection process according to step 4.1.

Step 4.3 If $V_H = \varnothing$ make v_α a root cluster, else find the set of vectors $V_P \subseteq V_H$ such that $S(v_\alpha, v_\gamma) \geq t_{PCS}, \forall v_\gamma \in V_P$, and make v_α a child of v_γ. If $V_P = \varnothing$ make v_α a root cluster.

Step 4.4 Remove v_α from V_F and add it to V_H.

Step 5. Return the cluster hierarchy, partition matrix U and cluster centroids V.

The algorithm takes a heuristic approach for selecting a candidate cluster to be inserted into the hierarchy and for finding a parent cluster in the hierarchy for the current candidate. The candidate cluster is considered to be best parent to one of the remaining non-assigned clusters, excluding itself (step 4.1). Such selection process ensures the right ordering for a descending cluster insertion into the hierarchy. Once the candidate cluster has been selected, it is assigned to the hierarchy either under one of the existing clusters or at the root of the hierarchy. The parent selection criterion is based on a user defined threshold t_{PCS} for the asymmetric similarity (step 4.3). In case no suitable parent is found, the candidate cluster starts a new hierarchy branch. The higher the threshold value, the more clusters will appear at the root. Thus, the t_{PCS} parameter enables to control the depth of the hierarchy. When a limit is set for the hierarchy depth or for the number of hierarchy branches, the value of this threshold can be adaptively found.

Regarding the complexity of the H^2-FCM, this algorithm generates a hierarchy of fuzzy clusters with low computational costs, O(Nc^2), presenting linear time complexity with the number of documents.

4. Experimental trials

4.1 Data sets description

The clustering experiments have been carried out with four test document collections: two subsets of the Reuters-21578 text categorization collection (*reuters1* and *reuters2*) [14], a subset of the Open Directory Project metadata files (*odp*) [12] and a set of scientific abstracts from the INSPEC database (*inspec*) [7]. Reuters-21578 consists of newswire articles classified into 135 topic categories. We selected two sets of articles from the "training" set on the following topics: "trade", "acq" or "earn" (*reuters1:* N=1708, c_{REF}=3), and "crude", "interest", "money-fx", "ship" and "trade" (*reuters2*: N=1374, c_{REF}=5). The *odp* test collection (N=556, c_{REF}=5) was created with the short textual descriptions of Web sites from the *Kids and Teens* topic hierarchy that were related to the following topics: "game", "lego", "math", "safety" and "sport". The *inspec* test collection (N=7473, c_{REF}=3) was created by downloading all the abstracts from the INSPEC scientific database published since 2000 that contained the following keywords: "back-propagation", "fuzzy control" and "pattern clustering".

4.2 Document representation

Each text document was automatically indexed for term frequency extraction. Stop words (*i.e.* insignificant words like 'a', 'and', 'where', 'or') were eliminated and stemming (*i.e.* removing word affixes such as 'ing', 'ion', 's') was performed using Porter's stemming algorithm [13]. Documents were represented as TF (Term Frequency) vectors according to the Vector Space model of IR [1] and a pre-processing filter was applied to discard terms that appeared in a small percentage of documents, leading to significant dimensionality reduction without loss of clustering performance [11].

4.3 Performance evaluation measures

Precision and recall are two typical measures for evaluating the performance of information retrieval systems [1, 15]. Precision and recall have also been applied for evaluating text classification systems [8]. Likewise, these measures can be applied to evaluate the performance of clustering algorithms in cases where clustering benchmarks exist. In the clustering context, given a discovered cluster γ and the associated reference cluster Γ, precision ($P_{\gamma\Gamma}$) and recall ($R_{\gamma\Gamma}$) are defined as in equations (5) and (6), respectively. In these expressions $n_{\gamma\Gamma}$ is the number of documents from reference cluster Γ assigned to cluster γ, N_γ is the total number of documents in cluster γ and N_Γ is the total number of documents in reference cluster Γ. To obtain overall performance measures, a weighted average of the individual $P_{\gamma\Gamma}$ and $R_{\gamma\Gamma}$ is applied, leading to the expressions in (7) and (8).

$$P_{\gamma\Gamma} = \frac{n_{\gamma\Gamma}}{N_\gamma} \quad (5) \qquad R_{\gamma\Gamma} = \frac{n_{\gamma\Gamma}}{N_\Gamma} \quad (6) \qquad P = \sum_{\Gamma=1}^{c} N_\Gamma P_{\gamma\Gamma} \bigg/ \sum_{\Gamma=1}^{c} N_\Gamma \quad (7) \qquad R = \sum_{\Gamma=1}^{c} N_\Gamma R_{\gamma\Gamma} \bigg/ \sum_{\Gamma=1}^{c} N_\Gamma \quad (8)$$

In the fuzzy clustering case, documents may have membership in multiple clusters and it is even possible that all documents belong to all clusters to some degree. Consequently, the precision measure can result in very low values. Hence, fuzzy clusters are hardened according to the maximum membership criterion to calculate precision and recall.

4.4 Experiments and results

The main goal of these experiments is to establish whether having more clusters does indeed mean more granularity regarding the topics represented by each of them and whether the hierarchical linking heuristic of the H^2-FCM algorithm produces meaningful associations between the clusters.

Initially, we applied the H-FCM algorithm (with m=1.1) to the test document collections for a number of clusters that matched the number of reference classes c_{REF} and calculated the clustering precision and recall in each case. Then, we applied the H^2-FCM algorithm for a range of c values. In this case, the H^2-FCM generated a number of smaller clusters, c>c_{REF}, and linked them hierarchically. Thus, for assessing the performance of the algorithm not only do we have to analyse the actual quality of each individual cluster but also to determine whether sub-clusters of the same reference class are linked in the hierarchy.

The approach followed in the present experiments was to adaptively set the threshold t_{PCS} in order to obtain as many hierarchy branches as the number of reference classes in each document collection. Precision and recall were then calculated by comparing the contents of all the clusters from a given branch with the contents of the corresponding reference class. The documents membership in a given branch were taken as their maximum membership in any of the branch clusters.

Figure 4.1 presents the average clustering precision and recall of the H^2-FCM algorithm as a function of the number of clusters. From the plots it can be observed that the performance of the algorithm generally does not degrade as the number of hierarchy clusters increases. The average clustering precision and recall do not vary significantly for any of the test document collections. From these results we can conclude that as *c* increases, documents from the same reference class remain grouped together, but these documents are now divided into a higher number of smaller clusters. Furthermore, we can also conclude that the hierarchical linking procedure succeeds at placing in the same hierarchy branch clusters corresponding to the same reference topic.

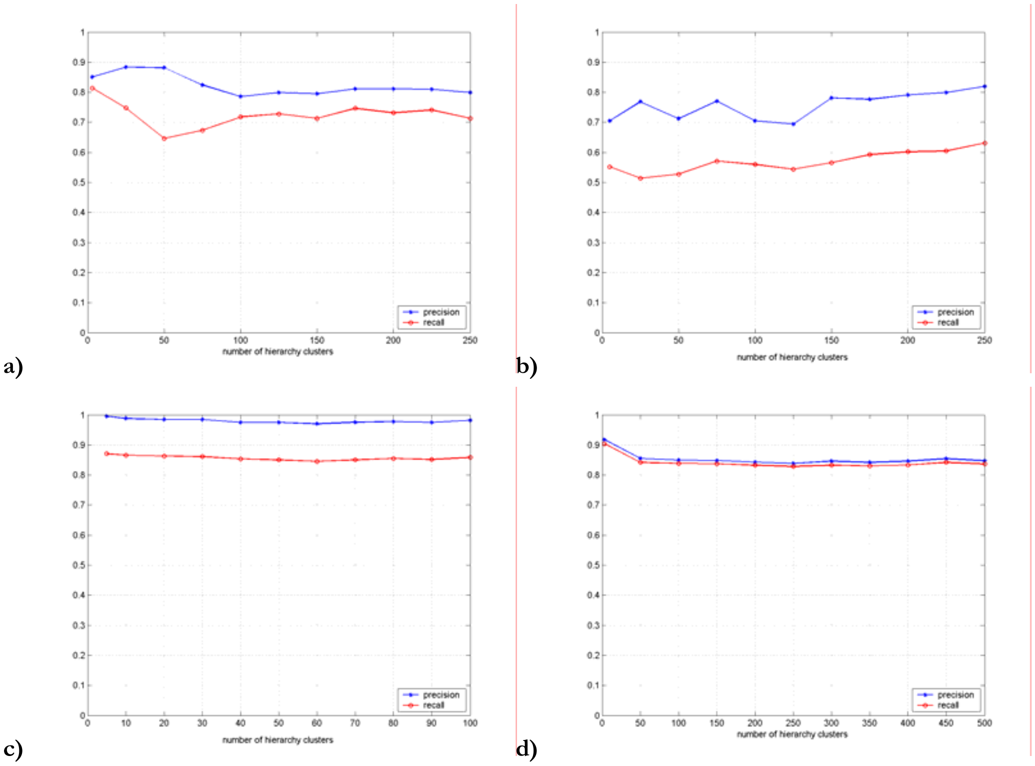

Figure 4.1: *Precision* and *recall* of the H^2-FCM for: a) *reuters1*, b) *reuters2*, c) *odp* and d) *inspec* collections.

In Figure 4.2, we present a graph visualisation of the H^2-FCM cluster hierarchy for the *odp* test collection (*c*=40). Clusters of various sizes were obtained. It can be verified that clusters which are more specific, *i.e.* that are located deeper in the hierarchy, generally contain less documents.

The algorithm generates a topic hierarchy through the set of terms that compose the cluster centroids. To simplify the graph only the first two weighted terms of the root centroids are shown. These terms summarise the topics covered by the documents in each branch. It can be observed that there is a direct mapping to the reference topics of the *odp* collection: “game”, “lego”, “math”, “safety” and “sport”.

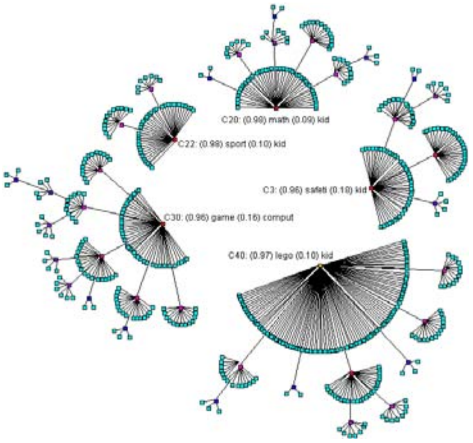
C20: (0.99) math (0.09) kid
C22: (0.99) sport (0.10) kid
C3: (0.96) safeti (0.18) kid
C30: (0.96) game (0.16) comput
C40: (0.97) lego (0.10) kid

Figure 4.2: The *odp* cluster hierarchy (*c*=40).

5. Conclusions

In this paper, we have proposed a novel fuzzy clustering algorithm for text mining – the Hierarchical Hyper-spherical Fuzzy c-Means (H^2-FCM). The algorithm exploits the notion of asymmetric similarity to link fuzzy clusters hierarchically and to form a meaningful topic hierarchy based on the clusters centroids. The time complexity of the H^2-FCM is linear with the number of documents $O(Nc^2)$, thus the algorithm is scalable to large document sets.

Precision and recall have been applied as objective quantitative measures of the clusters quality to evaluate the performance of the H^2-FCM algorithm. Our results have demonstrated that as the number of clusters increases, the H-FCM generates clusters with a higher level of granularity and that the resulting cluster hierarchy successfully links clusters of the same topic.

6. Acknowledgement

This work was supported by the Portuguese Foundation for Science and Technology through the PRAXIS XXI doctoral scholarship programme (grant ref. PRAXIS XXI/BD/21768/99).

References

[1] R. Baeza-Yates and B. Ribeiro-Neto (1999). Modern Information Retrieval. New York: Addison Wesley, ACM Press, 1999.

[2] J.C. Bezdek (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. New York: Plenum Press, 1981.

[3] D.R. Cutting, D.R. Karger, J.O. Pederson and J.W. Tukey (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In: *Proceedings of the* 15th *Annual International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'92, pp. 318-329, Copenhagen, Denmark, June 1992.

[4] I.S. Dhillon and D.S. Modha (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, vol. 42, no. 1-2, pp. 143-175, January-February 2001.

[5] I. Gath and A. Geva (1989).Unsupervised optimal fuzzy clustering. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 773-781, July 1989.

[6] M.A. Hearst, D.R. Karger and J.O. Pedersen (1995). Scatter/gather as a tool for the navigation of retrieval results. In: *Papers from the* AAAI *Fall Symposium* AI *Applications in Knowledge Navigation and Retrieval*, Technical Report FS-95-03, pp. 65-71, Cambridge, USA, November 1995.

[7] INSPEC database: http://www.iee.org/Publish/INSPEC/.

[8] D.D. Lewis (1991). Evaluating text categorization. In: *Proceedings of the* 1991 *Speech and Natural Language Workshop*, pp. 312-318, February 1991.

[9] M.E.S. Mendes, W. Jarrett, O. Prnjat and L. Sacks (2003). Flexible searching and browsing for telecoms learning material. In: *Proceedings of the* 2003 *International Symposium on Telecommunications*, IST'2003, Isfahan, Iran, August 2003.

[10] M.E.S. Mendes and L. Sacks (2003). Evaluating fuzzy clustering for relevance-based information access. In: *Proceedings of the* 12th IEEE *International Conference on Fuzzy Systems*, FUZZ-IEEE 2003, pp. 648-653, St. Louis, Missouri, USA, May 2003.

[11] M.E.S. Mendes and L. Sacks (2004). Dynamic Knowledge Representation for e-Learning Applications. In: M. Nikravesh, L. A. Zadeh, B. Azvin and R. Yager (editors). *Enhancing the Power of the Internet* - Studies in Fuzziness and Soft Computing, Springer, vol. 139, pp. 255-278, January 2004.

[12] Open Directory Project (ODP) : http://dmoz.org/.

[13] M. Porter (1980). An algorithm for suffix stripping. *Program*, vol. 14, no. 3, pp. 130-137, July 1980.

[14] Reuters-21578 test collection: http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[15] C.J. van Rijsbergen (1979). Information Retrieval. 2nd Edition. London: Butterworth, 1979.

[16] A. Schenker, M. Last and A. Kandel (2001). A term-based algorithm for hierarchical clustering of Web documents. In: *Proceedings of the Joint* 9th IFSA *World Congress and* 20th NAFIPS *International Conference*, vol.5, pp. 3076-3081, Vancouver, Canada, July 2001.

[17] A. Tversky (1977). Features of similarity. *Psychological Review*, vol. 84, no. 4, pp. 327-352, 1977.

[18] O. Zamir and O. Etzioni (1998). Web document clustering: a feasibility demonstration. In: *Proceedings of the* 21th *Annual International* ACM SIGIR *Conference on Research and Development in Information Retrieval*, SIGIR'98, pp. 46-54, Melbourne, Australia, August 1998.