Document Categorization with MAJORCLUST

Benno Stein, Sven Meyer zu Eissen
Department of Computer Science
Paderborn University, Germany
stein@upb.de, smze@upb.de

Abstract This paper investigates the text categorization capabilities of two special clustering algorithms: Fuzzy k-Medoid and MAJORCLUST. Aside from quantifying the categorization performance of the mentioned algorithms, our experimental setting will also help to answer special questions related to clustering problems such as cluster number determination or cluster quality evaluation.

Keywords Information Retrieval, Clustering, Document Categorization, Classification, LSI

1 Document Categorization and Clustering

Text categorization is a key concept to cope with huge collections of text documents such as the Internet, news repositories, or digital libraries. Text categorization means grouping together, say, clustering documents with similar topics; it can serve different purposes:

(1) Increasing the performance of the retrieval process. The respective technology is often called cluster-based search and can improve both the efficiency and the effectiveness of full search [6].

(2) Improving the user interface for browsing document sources. In particular, clustering can help to navigate, inspect, and organize document collections.

(3) Automatic text generation. A high-quality document clustering can form the basis for a further processing like summarization, or as a knowledge base for reasoning and documentation systems.

Note that the mentioned text categorization applications may be characterized by different prerequisites: The existence of a predefined classification scheme, the existence of a collection of classified documents, a classification process that is supervised by humans, etc.

1.1 Contributions of the Paper

Our focus lies on the development of a smart interface for Web-based search, where document clustering plays a key role. This implies that the category formation process is unsupervised: Except for experimental evaluation purposes no predefined classification scheme is given from which classification knowledge can be acquainted. In this connection Steinbach et al. have recently investigated several clustering algorithms [25, 29, 5]. Their evaluation of the quality of the found clusters is based on both external and internal quality measures. Measures of the former type compare the found clusters to the predefined categories (classes) as they are given in the analyzed document collections; measures of the latter type use no external reference knowledge. Even though Steinbach et al. provide answers and interpretations of their results, several questions are left open—amongst others the following:

(1) Should document clustering be performed in the original feature space?

(2) In which way can the “best” number of categories (clusters), k, be determined?

(3) How meaningful are external quality measures for cluster evaluation purposes?

At its heart, point (1) relates to the definition of the document similarity measure. If, for instance, human classification and category-finding relies on few latent semantic concepts, the term-document matrix should be transformed by a dimension reduction procedure like LSI. However, if the human classification process is oriented at a small number of terms (the core-vocabulary) as assumed in [25], the cosine-measure as normally applied will fail to find useful clusters. Point (2) relates to a classical clustering problem: Representative-based cluster algorithms like k-Means need the expected cluster number k as input; similarly, hierarchical algorithms require a threshold that determines the appropriate aggregation or division level in the cluster hierarchy. Note that in the document classification situation the parameter k may vary within a wide range, and that there is less a-priory knowledge on k. Point (3) picks up the observation made in [25]:

A large percentage of the documents in the investigated collections does not lie in the same class as their nearest neighbor. I. e., either the employed document similarity measure is too weak (see Point (1)), or the human performance in document clustering varies too much to be used as an external reference [16].

This paper contributes some experimental analyses to these questions. We investigate two clustering algorithms with respect to their capability in resembling the class structure of a document collection: Fuzzy-k-Medoid and MAJORCLUST [24]. The former is interesting because of its multiple cluster membership computation, which may be an advantage when clustering documents. The latter algorithm has—along its efficiency—the salient property that it automatically determines the number of clusters. However, with respect to interpretation and comparison purposes the experiments were also performed with the standard k-Means algorithm. For our experiments the term-document-vectors have been used in a raw form, but were also transposed to differently reduced feature spaces by means of Latent Semantic Indexing (LSI) [2]. A (supervised) classification of these reduced vectors shall give insights with respect to (1) the classification performance, say learnability, which can be regarded as an upper bound when evaluating the cluster quality, and (2) the robustness, say, simplicity of the human classification process.

2 Representing and Clustering Documents

Clustering algorithms work on abstract descriptions of the interesting objects. Typically, each such description is a vector $\mathbf{d}$ of numbers comprising values of essential object features. A common representation model for documents is the vector space model, where each document is represented in the term space, which (roughly) corresponds to the union of the n words that occur in a document collection [20, 11]. In this term space, the common words are filtered out by means of a stop word list, words that are unique in the collection are omitted, and stemming is applied to reduce words to a canonical form. A document d can then be described by a term vector $\mathbf{d} = (f_{t_1}, f_{t_2}, \ldots, f_{t_n})$, where f_{t_i} designates the frequency of term t_i in d. A widely accepted variant of this model includes an additional weighting of each term based on its inverse document frequency.

Clustering exploits knowledge about the similarity among the objects to be clustered. The similarity φ of two documents, d_1, d_2, is computed as a function of the distance between the corresponding term vectors $\mathbf{d}_1$ and $\mathbf{d}_2$. Various measures exist for similarity computation, from which the cosine-measure proved to be the most successful for document comparison. It is defined as follows.

$$\varphi(d_1, d_2) = \frac{\langle \mathbf{d}_1, \mathbf{d}_2 \rangle}{||\mathbf{d}_1|| \cdot ||\mathbf{d}_2||},$$

where $\langle \mathbf{d}_1, \mathbf{d}_2 \rangle = \mathbf{d}_1^T \mathbf{d}_2$ denotes the scalar product, and $||\mathbf{d}||$ the Euclidean length. Given these abstractions, a document collection can be considered as a weighted graph $G = \langle V, E, \varphi \rangle$ with node set V, edge set E, and weight function $\varphi : E \rightarrow [0, 1]$ where V represents the documents, and φ defines the similarities between two adjacent documents. This weighted graph forms the base for the subsequent clustering.

Definition 1 (Clustering) *Let V be a set of objects. A clustering $\mathcal{C} = \{C \mid C \subseteq V\}$ of V is a division of V into sets for which the following conditions hold: $\bigcup_{C_i \in \mathcal{C}} C_i = V$, and $\forall C_i, C_j \in \mathcal{C} : C_i \cap C_{j \neq i} = \emptyset$. Likewise, if V represents the nodes of a graph $G = \langle V, E, \varphi \rangle$, then $\mathcal{C}$ is called a clustering of G. Both the sets C_i and the induced subgraphs $G(C_i)$ are called clusters.*

2.1 Clustering Approaches

Clustering data given as graphs has been a focus of research for years. The existing approaches can be classified as follows.

Hierarchical Algorithms. Hierarchical algorithms create a tree of node subsets by successively subdividing or merging the graph's nodes. In order to obtain a unique clustering, a second step is necessary that prunes this tree at adequate places. Agglomerative hierarchical algorithms start with each vertex being its own cluster and union clusters iteratively. For divisive algorithms on the other hand, the entire graph initially forms one single cluster which is successively subdivided. Examples for agglomerative algorithms are k-nearest-neighbor, linkage, or Ward methods [3, 23, 7]. Typical divisive algorithms are Min-cut-clustering [13, 27] or dissimilarity-based algorithms [17].

Iterative Algorithms. Iterative algorithms strive for a successive improvement of an existing clustering. These methods can be further classified into exemplar-based and commutation-based approaches. The former assume for each cluster a representative, e. g. a centroid (for interval-scaled features) or a medoid (for arbitrary similarity measures) [8], to which the objects become assigned according to their similarity. The most well-known amongst these algorithms is k-Means [15], which resembles a self-organizing Kohonen network whose neighborhood function is set to size 1 [10].

Meta-Search Algorithms. Meta-Search algorithms treat a clustering task as an optimization problem where a given goal criterion is to be minimized or maximized [1, 21, 22, 21]. Though this approach offers maximum flexibility, only less can be stated respecting its runtime: Meta-search driven cluster detection may be realized by genetic algorithms [19, 4], simulated annealing [9], or a two-phase greedy strategy [29].

Exclusive versus Non-exclusive Algorithms. Exclusive clustering algorithms assign every node to exactly one cluster, while non-exclusive algorithms assign a membership value to a node with respect to each cluster. A well-known class of non-exclusive algorithms are Fuzzy clustering algorithms [28].

In the following we outlined the investigated clustering algorithms. The starting point is a document collection, given as a weighted graph $G = \langle V, E, \varphi \rangle$, where each $v \in V$ is uniquely associated with a document's term vector $\mathbf{d}$.

2.2 The Fuzzy k-Medoid Algorithm

Fuzzy k-Medoid is the non-exclusive version of k-Medoid, which in turn is the variant of k-Means that is able to operate on arbitrary distance measures. These algorithms belong to the class of iterative, exemplar-based clustering algorithms and follow the same algorithmic scheme:

(a) Choose $r_1, \ldots, r_k$ representative documents from V.

(b) $\forall v \in V$: Assign v to cluster C_i iff $\varphi(v, r_i), i = 1, \ldots, k$ is minimum.

(c) $\forall C_i, i = 1, \ldots, k$: Compute a new representative r_i. Goto (b) until a convergence criterion is satisfied.

Within the standard k-Means algorithm, a representative r_i is computed according to Equation (1). Within the k-Medoid algorithm, a representative r_i of a cluster C_i is an element in V that minimizes Sum (2). Within the Fuzzy-k-Medoid algorithm, each node v in V belongs to every cluster C_i up to a certain degree $\mu_{v,i}$. A representative r_i of a cluster C_i is an element in V that minimizes Sum (3).

$$r_i = \frac{1}{|C_i|} \sum_{v \in C_i} \mathbf{d}_v \qquad (1) \qquad\qquad \sum_{v \in C_i} \varphi(v, r_i) \qquad (2) \qquad\qquad \sum_{v \in C_i} \mu_{v,i}^m \cdot \varphi(v, r_i) \qquad (3)$$

The exponent m in Sum (3) is called fuzzifier and is normally set to 2. With the additional constraint $1 = \sum_{i=1}^k \mu_{v,i}$ the membership function μ can be implicitly defined by the minimum of the following error function e:

$$e = \sum_{i=1}^{k} \sum_{v \in C_i} \mu_{v,i}^m \cdot \varphi(v, r_i), \quad \Rightarrow \quad \mu_{v,i} = \frac{\frac{1}{\varphi(v,r_i)}}{\sum_{j=1}^k \frac{1}{\varphi(v,r_j)}}$$

2.3 The MAJORCLUST Algorithm

MAJORCLUST is a new clustering algorithm presented in [24]. It strives at a maximization of a graph's *weighted partial connectivity* Λ, which is defined as follows.

Definition 2 (Λ) *Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering of a graph $G = \langle V, E, \varphi \rangle$.*

$$\Lambda(\mathcal{C}) := \sum_{i=1}^{k} |C_i| \cdot \lambda_i,$$

where λ_i designates the weighted edge connectivity of $G(C_i)$. The weighted edge connectivity, λ, of a graph $G = \langle V, E, \varphi \rangle$ is defined as $\min \sum_{\{u,v\} \in E'} \varphi(u, v)$ where $E' \subset E$ and $G' = \langle V, E \setminus E' \rangle$ is not connected. λ is also designated as the capacity of a minimum cut of G.

Initially, MAJORCLUST assigns each node of a graph its own cluster. Within the following re-clustering steps, a node adopts the same cluster as the majority of its weighted neighbors. If several such clusters exist, one of them is chosen randomly. If re-clustering comes to an end, the algorithm terminates.

MAJORCLUST.

Input. A graph $G = \langle V, E, \varphi \rangle$.

Output. A function $c : V \to \mathbf{N}$, which assigns a cluster number to each node.

```
(1) $n = 0, t = \mathit{false}$
(2) $\forall v \in V$ **do** $n = n + 1, c(v) = n$ **end**
(3) **while** $t = \mathit{false}$ **do**
(4)     $t = \mathit{true}$
(5)     $\forall v \in V$ **do**
(6)         $c^* = i$ **if** $\left( \sum_{\substack{c(u)=i,\\ \{u,v\} \in E}} \varphi(u,v) \right)$ is max.
(7)         **if** $c(v) \neq c^*$ **then** $c(v) = c^*, t = \mathit{false}$
(8)     **end**
(9) **end**
```

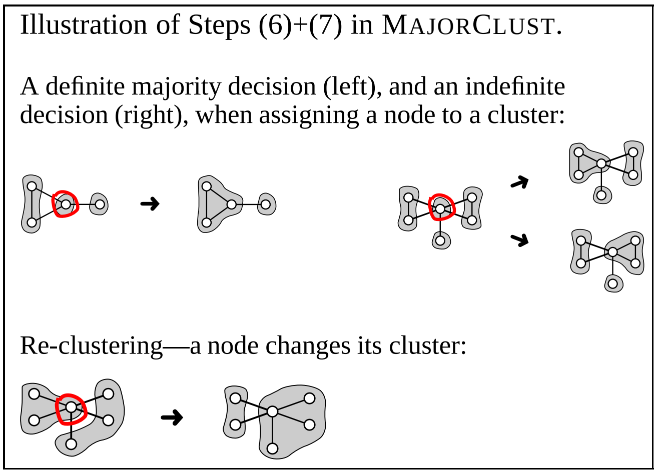

Illustration of Steps (6)+(7) in MAJORCLUST.

A definite majority decision (left), and an indefinite decision (right), when assigning a node to a cluster:

Re-clustering—a node changes its cluster:

The runtime complexity of MAJORCLUST is $\Theta(|E| \cdot |C_{max}|)$, where $C_{max} \subseteq V$ designates a maximum cluster. Note that choosing a node $v \in V$ in Step (5) and choosing between clusters with the same attraction in Step (6) must happen randomly. MAJORCLUST can be classified as non-hierarchical, exclusive cluster algorithm. MAJORCLUST finds a fast, but possibly suboptimal solution for the problem of Λ-maximization.

3 Experimental Analysis

For our experiments the Reuters-21578 text document database was used [14]. A considerable part of the documents are assigned to one or more categories. To uniquely measure the classification performance, only single-topic documents are considered, which make up 66 different categories of different sizes. The 3 biggest categories contain more than 70% of all single-topic documents, and the 10 biggest categories make up 86% of all single-topic documents. To account for these biased a-priory probabilities, the investigated test sets were constructed as uniform distributions with 10 classes. To render the representation of the documents more precisely, stop words were reduced from each document by means of Porter's stemming algorithm [18]. The algorithm reduced the number of different words by more than 30%.

We chose the popular F-Measure for validating and comparing different clusterings [12]. The F-Measure combines the precision and recall ideas from information retrieval: Let V represent the set of documents and let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a clustering of V. Moreover, let $\mathcal{C}^* = \{C_1^*, \ldots, C_l^*\}$ designate the "correct" clustering of V, i. e., the known classification developed by a human editor. Then the recall of cluster j with respect to class i, $rec(i,j)$, is defined as $|C_j \cap C_i^*|/|C_i^*|$. The precision of cluster j with respect to class i, $prec(i,j)$, is defined as $|C_j \cap C_i^*|/|C_j|$. The F-Measure combines both values according to Formula (4). Based on this formula, the F-Measure for the *overall* quality of a clustering $\mathcal{C}$ is defined by means of Equation (5).

$$F_{i,j} := \frac{2 \cdot prec(i,j) \cdot rec(i,j)}{prec(i,j) + rec(i,j)} \qquad (4)$$

$$F := \sum_{i=1}^{l} \frac{|C_i^*|}{|V|} \cdot \max_{j=1,\ldots,k} \{F(i,j)\} \qquad (5)$$

3.1 Classification Results (supervised)

Although our main objective is unsupervised text categorization, we performed classification experiments to get an idea of both the difficulty of the learning problem and "randomly" assigned categories. For this purpose a linear classifier in the form of a neural network was employed, which also forced us to substantially reduce the dimension of the feature space. In this connection, we applied the LSI-reduction with the target dimensions of 40, 20, and 10.

# Features	Feature reduction	# Samples train./test	Class.
40	LSI	400/100	0.92
40	random	400/100	0.16
40	LSI	800/200	0.86
40	random	800/200	0.23
20	LSI	800/200	0.83
20	random	800/200	0.11
10	LSI	800/200	0.75
10	random	800/200	0.11

		F-Measure		
# Docs	# Features	k-Means abs./norm.	Fuzzy k-Medoid abs./norm.	MajorClust abs./norm.
400	4616	0.36/ –	0.61/ –	0.41/ –
400	40	0.32/0.35	0.58/0.63	0.64/0.70
800	6277	0.36/ –	0.50/ –	0.43/ –
800	40	0.36/0.42	0.47/0.55	0.62/0.72
800	20	0.43/0.52	0.59/0.71	0.63/0.76
800	10	0.52/0.69	0.61/0,81	0.55/0.73

Table 1 (left): Classification performance of the linear classifier, depending on the number of features (1. column), the feature reduction method (2. column), and the number of samples (3. column). **Table 2** (right): Classification performance of the investigated clustering algorithms, quantified in F-Measure values (between 0 and 1, larger is better). Here, "abs." designates the values according to Equation (5), while "norm." designates the F-Measure values that have been normalized by the LSI-classification results of the left table.

The LSI-reduction did not include the test samples; in fact, they were projected into the reduced feature space. If LSI was performed on a matrix which contained both the test and the training samples, knowledge of the test samples would be compiled into the LSI-reduced training samples. This would significantly increase the classification performance—but not reflect the underlying application: In the unsupervised setting we neither know the contents of a document nor its class. Aside from the LSI-transformed features we also made experiments with randomly chosen indices of the document vectors. Table 1 comprises the results.

3.2 Clustering Results (unsupervised)

We used the training sets of the neural network for clustering, and performed several runs of k-Means and k-Medoid $(k = 2, 4, 8, 10, 16, 32)$ and calculated the F-Measure for all resulting clusterings. From these results we took the maximum F-Measure values (cf. Table 2, column 3 and 4). Note that MajorClust implicitly determines the cluster number k, and hence multiple runs are not necessary. The outermost column of the right table shows the respective F-Measure results for a clustered document similarity graph from which edges with a weight < 0.5 have been removed.

In particular, we suggest normalizing the F-Measure values of the clustering experiments with regard to a collection-specific maximum F^*: Obviously, the unsupervised classification performance cannot exceed the supervised classification performance, if an external quality assessment like the F-Measure is applied. In this connection the supervised classification result as estimations for F^* were used.

4 Summary

To efficiently search in large document repositories such as the World Wide Web, smart interface technology is required. In this respect it is advantageous to employ clustering algorithms in order to generate *query-dependent* categories—rather than using predefined categories as realized in the Yahoo! directory. However, the key question is whether clustering algorithms can compete with human classification abilities.

We evaluated two modern clustering algorithms that have not been applied for text categorization so far: Fuzzy k-Medoid and MajorClust. The results we obtained are encouraging. A comparison of the results to a standard k-Means implementation shows that these algorithms perform significantly better. Despite its algorithmic simplicity it turned out that MajorClust performed at a slightly higher level than Fuzzy k-Medoid, although MajorClust has not been informed of the number of desired categories.

From our point of view, the evaluation of the quality of a clustering can only be rated by an external measure, i. e., by a comparison to an existing classification. In this paper the F-Measure was used; it quantifies the precision and recall performance of a detected clustering. However, the application of external measures is crucial since the underlying human classification behavior may be biased or unreliable. To account for this situation we related the found clusterings with respect to a supervised classification, in order to obtain a guideline for the performance that is achievable.

The F-Measure values as reported in Table 2 are similar to those published in [25]. Nevertheless, we would like to note that our experimental setting is more pretentious with respect to the F-Measure computation. Steinbach et al. do not resort to a single clustering when computing an F-Measure value: The argument of the $\max$-operator in Equation (5), Page 7 in [25], comprises all possible clusters in the dendogram.

References

[1] Thomas Bailey and John Cowles. Cluster Definition by the Optimization of Simple Measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, September 1983.

[2] Scott C. Deerwester, Susan Dumais, Thomas Landauer, George Furnas, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[3] K. Florek, J. Lukaszewiez, J. Perkal, H. Steinhaus, and S. Zubrzchi. Sur la liason et la division des points d'un ensemble fini. *Colloquium Methematicum*, 2, 1951.

[4] D. Fogel and L. Fogel. Special Issue on Evolutionary Computation. *IEEE Trans. of Neural Networks*, 1994.

[5] E.-H. Han and G. Karypis. Centroid-Based Document Classification: Analysis and Experimental Results. Tech. Report 00-017, University of Minnesota, Dept of Computer Science / Army HPC Research Center, 2000.

[6] Makoto Iwayama and Takenobu Tokunaga. Cluster-based text categorization: a comparison of category search strategies. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 273–281, Seattle, USA, 1995. ACM Press, New York, US.

[7] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32, 1967.

[8] Leonard Kaufman and Peter J. Rousseuw. *Finding Groups in Data*. Wiley, 1990.

[9] R. W. Klein and R. C. Dubes. Experiments in Projection and Clustering by Simulated Annealing. *Pattern Recognition*, 22:213–220, 1989.

[10] T. Kohonen. *Self Organization and Assoziative Memory*. Springer, 1990.

[11] Gerald Kowalsky. *Information Retrieval Systems—Theory and Implementation*. Kluwer Academic, 1997.

[12] Bjornar Larsen and Chinatsu Aone. Fast and Effective Text Mining Using Linear-time Document Clustering. In *Proceedings of the KDD-99 Workshop San Diego USA*, San Diego, CA, USA, 1999.

[13] Thomas Lengauer. *Combinatorical algorithms for integrated circuit layout*. Applicable Theory in Computer Science. Teubner-Wiley, 1990.

[14] D. Lewis. Reuters-21578 Text Categorization Test Collection. `http://www.research.att.com/`, 1994.

[15] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[16] Sofus A. Macskassy, Arunava Banerjee, Brian D. Davison, and Haym Hirsh. Human Performance on Clustering Web Pages: A Preliminary Study. In *Knowledge Discovery and Data Mining (KDD-98)*, pages 264–268, New York City, USA, August 1998.

[17] M. B. Dale P. MacNaughton-Smith, W. T. Williams and L. G. Mockett. Dissimilarity analysis. *Nature*, 1964.

[18] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[19] V. Raghavan and K. Birchand. A Clustering Strategy Based on a Formalism of the Reproduction Process in a Natural System. In *Proc. of the Second Int. Conf. on Information Storage and Retrieval*, pages 10–22, 1979.

[20] C. J. van Rijsbergen. *Information Retrieval*. Buttersworth, London, 1979.

[21] Tom Roxborough and Arunabha. Graph Clustering using Multiway Ratio Cut. In Stephen North, editor, *Graph Drawing*, Lecture Notes in Computer Science, Springer Verlag, 1996.

[22] Reinhard Sablowski and Arne Frick. Automatic Graph Clustering. In Stephan North, editor, *Graph Drawing*, Lecture Notes in Computer Science, Springer Verlag, 1996.

[23] P.H.A. Sneath. The application of computers to taxonomy. *J. Gen. Microbiol.*, 17, 1957.

[24] Benno Stein and Oliver Niggemann. *25. Workshop on Graph Theory*, chapter On the Nature of Structure and its Identification. Lecture Notes on Computer Science, LNCS. Springer, Ascona, Italy, July 1999.

[25] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. Technical Report 00-034, Department of Computer Science and Egineering, University of Minnesota, 2000.

[26] Frank Wissbrock. Fuzzy Clustering in Document Classification. Diploma thesis, Paderborn University, Department of Computer Science, July 2002.

[27] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Nov. 1993.

[28] J. Yan and P. Hsiao. A fuzzy clustering algorithm for graph bisection. *Inf. Processing Letters*, 52, 1994.

[29] Ying Zaho and George Karypis. Criterion Functions for Document Clustering: Experiments and Analysis. Technical Report 01-40, Univercity of Minnesota, Department of Computer Science / Army HPC Research Center, Feb 2002.