Modification to K-Medoids and CLARA for Effective Document Clustering

Phuong T. Nguyen[1], Kai Eckert[2], Azzurra Ragone[3], Tommaso Di Noia[4]

[1] Duy Tan University, 182 Nguyen Van Linh, Da Nang, Vietnam
phuong.nguyen@duytan.edu.vn
[2] Stuttgart Media University, Nobelstr. 10 – D-70569 Stuttgart, Germany
eckert@hdm-stuttgart.de
[3] University of Milano-Bicocca, Piazza dell Ateneo Nuovo 1 – 20126 Milano, Italy
azzurra.ragone@unimib.it
[4] SisInf Lab, Polytechnic University of Bari, Via Orabona 4 – 70125 Bari, Italy
tommaso.dinoia@poliba.it

Abstract. Document clustering plays an important role in several applications. K-Medoids and CLARA are among the most notable algorithms for clustering. These algorithms together with their relatives have been employed widely in clustering problems. In this paper we present a solution to improve the original K-Medoids and CLARA by making change in the way they assign objects to clusters. Experimental results on various document datasets using three distance measures have shown that the approach helps enhance the clustering outcomes substantially as demonstrated by three quality metrics, i.e. *Entropy*, *Purity* and *F-Measure*.

1 Introduction

Document analysis accounts for a crucial part in many research fields such as Data Mining and Information Retrieval. The boom of social networks in the recent years has made document analysis become even more important. Among others, document clustering algorithms have come under the spotlight. Clustering can be used to assist document browsing and produce document summary [1]. For intelligent systems and social networks, clustering is utilized in generating forecasts and recommendations [4]. The K-Means algorithm has been applied widely in document clustering due to its simplicity. However the algorithm is susceptible to noise and outliers [7],[10]. K-Medoids was proposed and helps overcome the weakness of K-Means. An improved version of K-Medoids, CLARA has been derived to solve the problem of clustering big datasets. In the assignment step, both K-Medoids and CLARA attach an object to the cluster whose the medoid is closest to the object. However, we see that this may not help optimize the *compactness* of the objects within a cluster since the distance of the new object to the others is not considered. In this paper we present an approach to improve K-Medoids and CLARA. By making change in the way objects are assigned to cluster we are able to increase the overall effectiveness of the clustering. For comparison, we used some *de facto* standard document sets

as the input data. Through the use of *Entropy*, *Purity* and *F-Measure* as quality metrics, we saw that the amendment facilitates better clustering solutions. The main contributions of this paper are: (1) Proposing a modification to enhance the effectiveness of K-Medoids and CLARA; and (2) Evaluating the performance of some clustering algorithms on standard document sets.

2 Modified K-Medoids and Modified CLARA

2.1 K-Medoids and CLARA

K-Medoids. The K-Medoids algorithm groups a set of n data objects to a predefined number of clusters κ. Medoids are the reference point for the assignment of objects. First, a set of initial medoids is generated randomly, then a medoid is selected as the object in the cluster that has minimum average distance to all objects in the cluster. Objects are assigned to the cluster with the closest medoid. K-Medoids is explained by means of the following greedy strategy:

- Step 1: Populate initial medoids by randomly selecting κ objects.
- Step 2: Assign each of the remaining objects to the cluster with the nearest medoid.
- Step 3: Calculate a new set of medoids. For each cluster, promote the object having the smallest average distance to the other objects in the cluster to the new medoid. If there are no changes in the set of medoids, stop the execution and return the resulting clusters in Step 2. Otherwise go back to Step 2.

CLARA. CLARA (Clustering LARge Applications) was designed to deal with large datasets [6]. It draws different samples of objects and applies K-Medoids on these samples to find the best set of medoids. The remaining objects are then assigned to the closest medoid. This helps save processing time by finding medoids from subsets of data objects. CLARA is briefly recalled as follows [6]:

Set $minD \leftarrow N$, N is a large enough number. Repeat the following steps for 5 times:

- Step 1: Choose randomly a set of $40 + 2\kappa$ objects. Apply K-Medoids on this set to find κ medoids.
- Step 2: Assign each of the remaining objects to the cluster with the nearest medoid.
- Step 3: Compute $avgD$ the average distance of the clustering solution in Step 2. If $avgD < minD$ then $minD \leftarrow avgD$ and select κ medoids in Step 1 as the current medoids. Go back to Step 1.

2.2 Modification to K-Medoids and CLARA

A clustering algorithm aims to minimize the intra-distance within a cluster and maximize the inter-distance to other clusters at the same time. With K-Medoids and CLARA, the assignment of an object to the closest medoid helps reduce

the distance to the medoid. However, the average distance of a newly joining object to the other objects of the cluster might be high since the assignment does not take this into account. Objects in a cluster can be close to the medoid but they may be well apart among themselves. Given the circumstance, the assignment cannot minimize the average distance from the new object to the objects of the cluster. In this sense, we see that there is room for improvement. We propose making an amendment to the original K-Medoids (**oKM**) and the original CLARA (**oCLARA**) as follows. An object is assigned to a cluster iff the average distance from the object to all objects of the cluster is minimal. We retain the steps of **oKM** and **oCLARA** except for Step 2:

– Step 2: Assign each of the remaining objects **to the cluster with which it has the smallest average distance to all objects of the cluster**.

The proposed amendment is applied to K-Medoids and CLARA with the aim of improving their effectiveness. Two algorithms are derived, namely modified K-Medoids (**mKM**) and modified CLARA (**mCLARA**). In the succeeding sections, we are going to investigate whether the modification is beneficial by comparing the performance of **oKM**, **mKM**, **oCLARA** and **mCLARA**.

3 Document Clustering

We consider the problem of document clustering where a set of n documents needs to be grouped into different clusters. Based on the relationship among the input documents, a clustering algorithm distributes them to independent groups in a way that both the similarity among the members of a group as well as the dissimilarity among groups can be maximized.

3.1 Extraction of Document Features

To serve as the input for the clustering process, it is necessary to calculate the distance between each pair of documents. In the first place, a document needs to be represented in a mathematically computable form. We adopted the vector representation [2],[5]. There, a document is modeled as a feature vector where each element corresponds to the weight of a term in the document [1],[11]. If we consider a set of documents D and a set of terms $t = (t_1, t_2, .., t_r)$ then the representation of a document $d \in D$ is vector $\boldsymbol{\delta} = (w_1^d, w_2^d, .., w_r^d)$ where w_l^d is the weight of term l in d and computed using the *term frequency-inverse document frequency* function with f_l^d being the frequency of t_l in d [8]:

$$w_l^d = tf \cdot idf(l, d, D) = f_l^d \cdot log \frac{n}{|\{d \in D : t_l \in d\}|} \tag{1}$$

3.2 Distance Measures

Considering two documents d and e represented by feature vectors $\boldsymbol{\delta} = \{\delta_l\} \mid_{l=1,..,r}$ and $\boldsymbol{\epsilon} = \{\epsilon_l\} \mid_{l=1,..,r}$, the following metrics are utilized to calculate distance:

Cosine Similarity. A set of terms $t = (t_1, t_2, .., t_r)$ forms an r-dimension space and for each pair of two vectors $\boldsymbol{\delta}$ and $\boldsymbol{\epsilon}$ there is an angle between them. Intuitively, the cosine similarity metric measures the similarity as the cosine of the corresponding angle between the two vectors. And the distance between them is equivalent to the dissimilarity:

$$D_C(d, e) = 1 - SIM_C(d, e) = 1 - \frac{\sum_{l=1}^{r} \delta_l \cdot \epsilon_l}{\sqrt{\sum_{l=1}^{r} (\delta_l)^2} \cdot \sqrt{\sum_{l=1}^{r} (\epsilon_l)^2}}$$

Tanimoto Coefficient. The similarity defined by Tanimoto coefficient:

$$SIM_T(d, e) = \frac{\sum_{l=1}^{r} \delta_l \cdot \epsilon_l}{\sum_{l=1}^{r} (\delta_l)^2 + \sum_{l=1}^{r} (\epsilon_l)^2 - \sum_{l=1}^{r} \delta_l \cdot \epsilon_l}$$

And the distance between d and e is:

$$D_T(d, e) = 1 - SIM_T(d, e) \tag{2}$$

Euclidean Distance. Euclidean distance computes the geometric distance between d and e in the r-dimension space as given below [5] .

$$D_E(d, e) = (\sum_{l=1}^{r} \left| w_l^d - w_l^e \right|^2)^{\frac{1}{2}} \tag{3}$$

4 Evaluation

Once the distance between every pair of documents has been identified, we are able to cluster a set of documents. In order to validate the proposed hypotheses, we compared the clustering performance of **mKM** and **mCLARA** with that of **oKM** and **oCLARA** with reference to a set of *de facto* standard document sets. We examined if the modification helps improve the effectiveness of clustering. We used Latent Dirichlet Allocation (**LDA**) as baseline for our evaluation. **LDA** deals with the modeling of topics, however it can also be utilized in clustering documents [3]. Due to space limitation, the algorithm is not recalled in this paper, interested readers are referred to the original paper for further detail [7].

Experiments on various datasets have been performed by using the distance measures in Section 3.2. We chose some of the datasets available at the website of Karypis Lab[5] for the experiments. There, pre-processing stages had been conducted to extract terms from documents and term frequencies were then calculated and saved into text files [11]. Furthermore, category for each document, e.g. *Sport*, *Financial*, *Foreign*, has been identified and can be read from the provided files [12]. A summary of the datasets is given in Table 1.

[5] `http://glaros.dtc.umn.edu/gkhome/fetch/sw/cluto/datasets.tar.gz`

	la1	**la2**	**re0**	**re1**	**tr31**	**tr41**	**tr45**	**wap**
# of Docs (n)	3204	3075	1504	1657	927	878	690	1560
# of Terms (r)	31472	31472	2886	3758	10128	7454	8261	8460
# of Classes (κ)	6	6	13	25	7	10	10	20
# of Weights	484024	455383	77808	87328	248903	171509	193605	220482

Table 1. Datasets used for evaluation.

4.1 Evaluation Metrics

Every document in the datasets has already been classified into a pre-defined category. In the following we call the categories in a dataset $C = (C_1, C_2, .., C_k)$ as *classes*, being C_i the set of documents whose category is i, and the resulting groups of the clustering process $\hat{C} = (\hat{C}_1, \hat{C}_2, .., \hat{C}_\kappa)$ as *clusters*. We performed *external evaluation* to measure the extent to which the produced clusters match the classes [9]. Given a clustering solution $\hat{C}$, the task is to compare the relatedness of the clusters in $\hat{C}$ to the classes in C. Three evaluation metrics *Entropy*, *Purity* and *F-Measure* were chosen to analyze the clustering solutions. The rationale for this selection is that the metrics have been widely utilized in evaluating clustering and appear to be effective [5],[9],[11],[12]. The metrics are briefly recalled as follows.

Entropy. This metric gauges the relatedness of a cluster to the classes by measuring the presence of the classes in a cluster. If p_{ij} is the probability that a member of class i is found in cluster j then Entropy for cluster j is computed according to the probability of the existence of all classes in j:

$$E_j = -\sum_{i=1}^{\kappa} p_{ij} \cdot log(p_{ij}) = -\sum_{i=1}^{\kappa} \frac{\left|C_i \cap \hat{C}_j\right|}{\left|\hat{C}_j\right|} \cdot log(\frac{\left|C_i \cap \hat{C}_j\right|}{\left|\hat{C}_j\right|})$$

The Entropy value for a clustering solution is weighted across all clusters:

$$E = \sum_{j}^{\kappa} \frac{\left|\hat{C}_j\right|}{n} \cdot E_j \tag{4}$$

Purity. It is used to evaluate how well a cluster matches a single class in C:

$$P_j = \frac{1}{\left|\hat{C}_j\right|} \cdot \max_i \left\{\left|C_i \cap \hat{C}_j\right|\right\} \qquad \text{and} \qquad P = \sum_{j}^{\kappa} \frac{\left|\hat{C}_j\right|}{n} \cdot P_j \tag{5}$$

F-Measure. By this metric *Precision* and *Recall* are utilized as follows:

$$F_{ij} = \frac{2 \cdot precision_{ij} \cdot recall_{ij}}{precision_{ij} + recall_{ij}} \qquad \text{and} \qquad F = \sum_{i}^{\kappa} \frac{|C_i|}{n} \cdot \max_j \{Fij\} \tag{6}$$

Precision is the fraction of documents of class i in cluster j, whereas Recall is the fraction of documents of cluster j in class i:

$$precision_{ij} = \frac{\left|C_i \cap \hat{C}_j\right|}{\left|\hat{C}_j\right|} \qquad \text{and} \qquad recall_{ij} = \frac{\left|C_i \cap \hat{C}_j\right|}{|C_i|}$$

If an ideal clustering solution is found, i.e. $\hat{C}_i \equiv C_i \mid_{i=1,..,\kappa}$ then Entropy is equal to 0, whilst both Purity and F-Measure are 1.0. If the clusters are completely different to the classes then Purity and F-Measure are equal to 0. This implies that a good clustering solution has low Entropy but high Purity and F-Measure. It is expected that the modification helps increase Purity, F-Measure but decrease Entropy at the same time.

4.2 Results

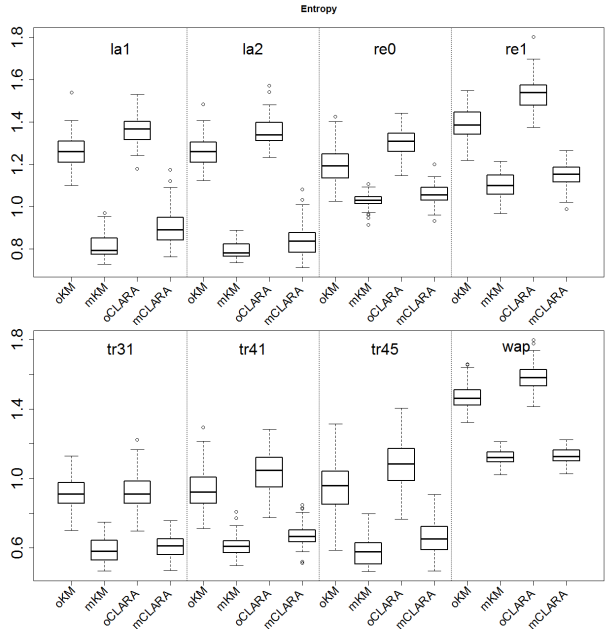

Fig. 1. Entropy of the clustering solutions using Cosine Similarity on the datasets

Using the datasets described in Table 1, we performed 24 independent experiments, each corresponds to applying one distance measure to a document set. From the distance scores, **oKM**, **mKM**, **oCLARA**, and **mCLARA** were applied to produce clusters. To aim for randomness, every clustering experiment was run in 100 trials. For the experiments with Cosine Similarity, the outcomes were visualized by sketching out Entropy, Purity and F-Measure. The boxplot diagram in Fig. 1 shows the Entropy scores for the datasets using Cosine Similarity. By all datasets, **mKM** and **mCLARA** help obtain a much better Entropy than **oKM** and **oCLARA** do. Similarly, as seen in Fig. 2, the Purity values for the clustering with **mKM** and **mCLARA** are much better. **mKM** produces a

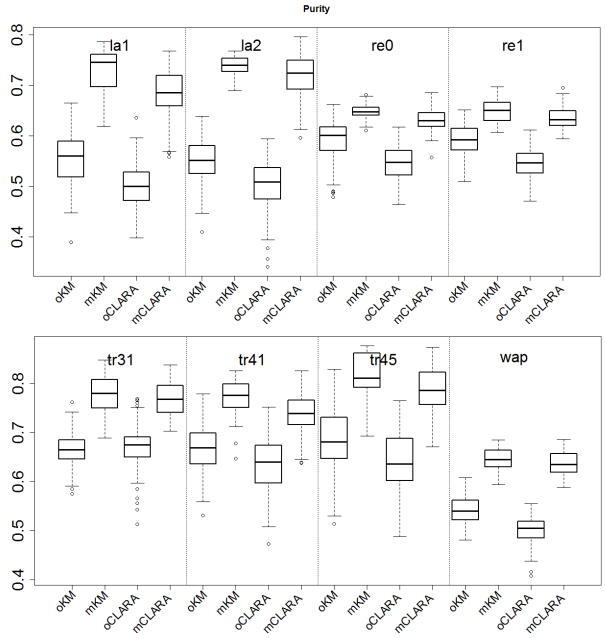

Fig. 2. Purity of the clustering solutions using Cosine Similarity on the datasets

clearly better Purity for **la1**. Fig. 3 shows the F-Measure scores obtained from performing the algorithms on all datasets. Equation 6 suggests that a good clustering solution has high F-Measure score. Back to the figures, we see that by all datasets, the F-Measure scores produced by **mKM** and **mCLARA** are of higher quality. These demonstrate that using Cosine Similarity on the datasets, the modified versions of the algorithms produce better results.

For other clustering experiments with Tanimoto and Euclidean, we also performed 100 trials each. Furthermore, **LDA** was used as a baseline for comparison. By **LDA**, a document can be assigned to each cluster with a specific probability and we attach the document to the cluster with the highest probability. A series of Entropy, Purity and F-Measure values have been derived from the clustering solutions. Due to space limitation the results for all trials are averaged out and the means are shown in Table 2, Table 3 and Table 4. In Table 2, we see that the Entropy scores using Tanimoto are similar to those of Cosine Similarity. However, with Euclidean distance the results obtained from **mKM** are the best among others whereas those between **oCLARA**, **mCLARA** and **LDA** are comparable. With Purity by Tanimoto, we witness the same pattern for these results as with Entropy, i.e. clustering with **mKM** and **mCLARA** yields superior Purity scores in all experiments. Again, with Euclidean distance **mKM** brings the best Purity while **oCLARA**, **mCLARA**, **LDA** possess similar F-Measure.

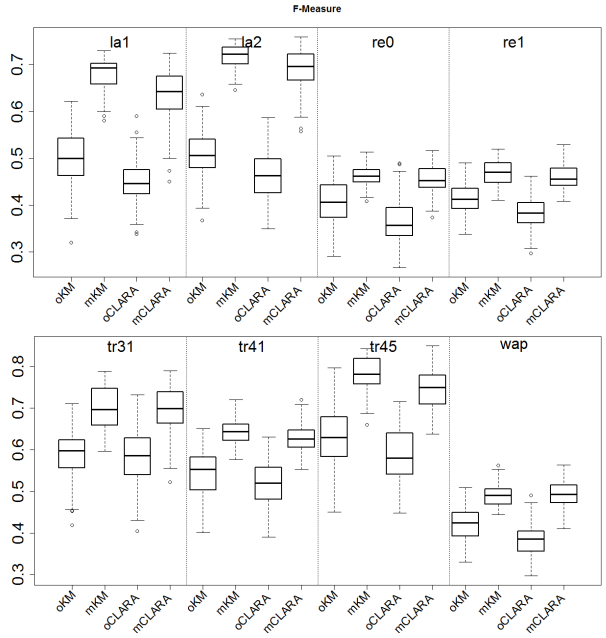

Fig. 3. F-Measure of the clustering solutions using Cosine Similarity on the datasets

		Dataset							
		la1	la2	re0	re1	tr31	tr41	tr45	wap
Tanimoto	oKM	1,33	1,33	1,26	1,43	0,94	1,02	1,05	1,46
	mKM	**1,03**	**0,95**	**1,11**	**1,14**	**0,61**	**0,69**	**0,65**	**1,10**
	CLARA	1,60	1,38	1,60	2,34	1,39	1,73	1,21	2,24
	mCLARA	1,06	1,00	1,13	1,16	0,62	0,74	0,78	1,11
Euclidean	oKM	1,26	**1,67**	1,19	1,39	0,91	0,93	1,05	1,47
	mKM	**0,81**	**1,67**	**1,02**	**1,10**	**0,58**	**0,61**	**0,63**	**1,12**
	CLARA	1,68	1,67	1,70	2,41	1,44	1,82	1,92	2,47
	mCLARA	1,68	1,67	1,77	2,42	1,46	1,85	2,00	2,40
LDA		1,68	1,67	1,74	2,39	1,51	1,85	2,01	2,40

Table 2. Entropy

The F-Measure scores for experiments with Tanimoto and Euclidean are shown in Table 4, together with F-Measure produced by **LDA**. With Tanimoto Coefficient, F-Measure scores by **mKM** and **mCLARA** are always superior to those of **oKM** and **oCLARA**. With Euclidean distance, the difference among all algorithms is marginal. F-Measure of **LDA** is inferior to that of the others.

To sum up, we see that applying the proposed modification to **oKM** and **oCLARA** in clustering appears to be effective as demonstrated by all evaluation metrics. It is evident that **mKM** and **mCLARA** are suitable for use in combination with Cosine Similarity and Tanimoto. Taken all metrics into consideration, i.e. Entropy, Purity and F-Measure, we see that **LDA** does not produce good outcomes compared with other algorithms. As already suggested

in [11], we are confident that **mKM** and **mCLARA** are the best algorithms in the given context. Under the circumstances, we come to the conclusion that the modification is highly beneficial to clustering on the observed datasets.

5 Related Work

In [5], an evaluation of the influence of similarity measures on clustering is presented. Using various distance measures, the effectiveness of various distance measure for document clustering is compared using Purity and Entropy. [12] presents a comprehensive study of partitional and agglomerative algorithms that use different criterion functions and merging schemes. The results demonstrate that partitional algorithms always lead to better solutions than agglomerative algorithms. This is also confirmed in [7]. The authors in [11] demonstrate a comparison for two document clustering techniques, namely agglomerative hierarchical clustering and K-Means. In this work, documents are modeled as feature vectors and Entropy, F-Measure and Overall Similarity are used as the metrics for evaluation. The work in [1] provides a comprehensive survey of text clustering. There, the key methods and their advantages of the clustering problem applied to the text domain are discussed. Furthermore, the potential of clustering for social networks and Linked Open Data is also mentioned.

		Dataset							
		la1	la2	re0	re1	tr31	tr41	tr45	wap
Tanimoto	oKM	0,52	0,50	0,56	0,58	0,69	0,64	0,66	0,54
	mKM	**0,61**	**0,64**	**0,61**	**0,63**	**0,76**	**0,73**	**0,79**	**0,65**
	CLARA	0,34	0,48	0,44	0,31	0,44	0,35	0,59	0,30
	mCLARA	**0,60**	**0,63**	**0,61**	**0,63**	**0,77**	**0,71**	**0,74**	**0,64**
Euclidean	oKM	0,55	**0,29**	0,59	0,59	0,66	0,66	0,66	0,54
	mKM	**0,73**	**0,29**	**0,64**	**0,64**	**0,77**	**0,77**	**0,80**	**0,64**
	CLARA	0,30	0,30	0,43	0,29	0,43	0,32	0,29	0,25
	mCLARA	0,29	0,30	0,42	0,28	0,42	0,30	0,26	0,25
LDA		0,29	0,29	0,40	0,24	0,37	0,28	0,23	0,22

Table 3. Purity

		Dataset							
		la1	la2	re0	re1	tr31	tr41	tr45	wap
Tanimoto	oKM	0,48	0,46	0,37	0,40	0,58	0,51	0,59	0,41
	mKM	**0,56**	**0,61**	**0,41**	**0,45**	**0,72**	**0,61**	**0,74**	**0,48**
	CLARA	0,31	0,44	0,32	0,21	0,43	0,31	0,53	0,21
	mCLARA	**0,55**	**0,60**	**0,41**	**0,45**	**0,72**	**0,60**	**0,69**	**0,49**
Euclidean	oKM	0,32	**0,33**	0,35	**0,23**	**0,40**	**0,31**	0,59	**0,19**
	mKM	**0,32**	**0,33**	**0,36**	0,22	**0,40**	**0,30**	**0,75**	**0,19**
	CLARA	**0,32**	**0,33**	**0,36**	**0,23**	0,39	**0,30**	0,26	**0,19**
	mCLARA	**0,32**	**0,33**	**0,36**	0,22	0,39	**0,30**	0,24	**0,19**
LDA		0,20	0,19	0,17	0,09	0,22	0,18	0,17	0,11

Table 4. F-Measure

6 Conclusion

Based on the observation that **oKM** and **oCLARA** may not minimize the average distance of a newly joining object to the other objects of the cluster, we proposed modified version of the algorithms and we applied it to the use case of document clustering. A document is assigned to the cluster with which it has the smallest average distance to all existing objects. Some *de facto* standard document datasets were utilized in the evaluation. Using *Entropy*, *Purity* and *F-Measure* as the quality metrics, we saw that the modified versions produce better outcome compared to the original ones. For future work, we plan to examine the performance of **mKM** and **mCLARA** for a variety of documents using more distance measures. Finally, we consider using other evaluation metrics to better study the performance of the proposed algorithms.

References

1. C. C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In C. C. Aggarwal and C. Zhai, editors, *Mining Text Data*, pages 77–128. Springer, 2012.
2. T. Basu and C. Murthy. A similarity assessment technique for effective grouping of documents. *Inf. Sci.*, 311(C):149–162, Aug. 2015.
3. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
4. T. DuBois, J. Golbeck, J. Kleint, and A. Srinivasan. Improving Recommendation Accuracy by Clustering Social Networks with Trust. New York, NY, USA, 2009.
5. A. Huang. Similarity measures for text document clustering. pages 49–56, 2008.
6. L. Kaufman and P. J. Rousseeuw. *Finding groups in data : an introduction to cluster analysis*. Wiley, New York, 1990.
7. R. T. Ng and J. Han. Clarans: A method for clustering objects for spatial data mining. *IEEE Trans. on Knowl. and Data Eng.*, 14(5):1003–1016, Sept. 2002.
8. J. W. Reed, Y. Jiao, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *Proceedings of the 5th International Conference on Machine Learning and Applications*, ICMLA '06, pages 258–263, Washington, DC, USA, 2006. IEEE Computer Society.
9. E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz. Internal versus external cluster validation indexes. *Int. Journal of Compt. and Comm.*, 5:27–34, March 2011.
10. L. Rokach and O. Maimon. *Clustering Methods*, pages 321–352. Springer US, Boston, MA, 2005.
11. M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *6th ACM SIGKDD, World Text Mining Conference*, 2000.
12. Y. Zhao, G. Karypis, and U. Fayyad. Hierarchical clustering algorithms for document datasets. *Data Min. Knowl. Discov.*, 10:141–168, March 2005.