ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Semi supervised clustering for Text Clustering

N.Saranya[1]

Assistant Professor, Department of Computer Science and Engineering, Sri Eshwar College of Engineering, Coimbatore[1]

ABSTRACT: Based on clustering algorithm Affinity Propagation (AP) I present this paper a semisupervised text clustering algorithm, called Seeds Affinity Propagation (SAP). There are two main contributions in my approach: 1) a similarity metric that captures the structural information of texts, and 2) seed construction method to improve the semisupervised clustering process. To study the performance and efficiency of the new algorithm, I applied it to the benchmark data and compared it to two state-of-the-art clustering algorithms, namely, k-means algorithm and the original AP algorithm. Furthermore, I have analyzed the individual impact of the two proposed contributions. Results show that the proposed similarity metric is more effective in text clustering and the proposed semisupervised strategy achieves both better clustering results and faster convergence. The complete SAP algorithm obtains higher F-measure and lower entropy, improves significantly clustering execution time (25 times faster) in respect that k-means, and provides enhanced robustness compared with all other methods.

KEYWORDS: Affinity propagation, text clustering, cofeature set, unilateral feature set, significant cofeature set.

I. INTRODUCTION

Clustering digital objects (e.g., text documents) by identifying a subset of representative examples plays an important role in recent text mining and information retrieval research. In fact, organizing a large amount of objects into meaningful clusters (clustering) is often used to browse a collection of objects and organize the results returned by a search engine. After the clustering process, the obtained clusters are represented with examples, which can include all or part of the features that appear in the cluster members. During cluster-based query processing, only those clusters that contain examples similar to the query are considered for further comparisons with cluster members, e.g., documents. This strategy, sometimes called Cluster-Based Retrieval, is intended to improve both efficiency and effectiveness of the document retrieval systems. My work focuses on the proposal and detailed analysis of a new effective and fast clustering algorithm that can be used in cluster-based retrieval tasks. By using AP to preprocess texts, developed an incremental method for text clustering. However, they used AP only as an unsupervised algorithm and did not consider any structural information derived from the specific documents. Semisupervised learning is a machine learning paradigm in which the model is constructed using both labeled and unlabeled data for training—typically a small amount of labeled data and a large amount of unlabeled data.

In this paper, I present a new clustering algorithm by extending Affinity Propagation with 1) a novel asymmetric similarity measurement that captures the structural information of texts, and 2) a semisupervised learning approach, where we exploit the knowledge from a few labeled objects versus a large number of unlabeled ones.

In information retrieval, there are several commonly used measurements of similarities. The simplest of all similarity measures, namely, simple matching coefficient, is counting the number of shared terms in two sets (e.g., documents). In this paper, I propose an asymmetric similarity measurement for two different documents, which is different from the conventional symmetric measurements. Embracing some ideas of positive and negative association rules proposed, I define three feature sets containing structural information. An asymmetric similarity measurement—called Tri-Set method—is thus proposed based on these three feature sets. Finally, I present and analyze the definition of specific initial values for the clustering algorithm, that we named Seeds, to bootstrap the initial phases of the new clustering algorithm. I thus propose a novel semisupervised clustering algorithm: Seeds Affinity Propagation (SAP). This model aims to address the complexity

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

problem in text clustering which results from the high dimension and sparse matrix computations. To examine the effectiveness of the proposed method, I have applied it to the benchmark data set Reuters-21578. In order to analyze the behavior of the new algorithm (and also the impact of the two individual proposed contributions), I have performed a detail comparison with four clustering methods on the same data set, namely,

1. k-means approach;

2. The original Affinity Propagation algorithm with conventional similarity measurement (AP(CC));

3. A modified Affinity Propagation method, which combines AP with the new similarity measurement (AP(Tri-Set)); and

4. A modified Affinity Propagation method which combines AP with the new seed construction semi supervised method (SAP(CC)).

In my experiments k-means is selected as the baseline state-of-the-art clustering algorithm. Many algorithms are derived from k-means or compete with it. In particular, several limitations about k-means in detail, such as its sensitivity to initialization, to the presence of outliers. My experimental results show that SAP offers better speed (i.e., about 20 times faster than k-means) and overall precision than the other four clustering algorithms (i.e., F-measures increase up to 44 percent compared with k-means).

II. RELATED WORK

AP was proposed as a new and powerful technique for exemplar learning. In brief, the user has to provide as initial input to the algorithm a complete matrix of similarities (for the selected metric(s)) among the input data points. At first, all data points are viewed as potential exemplars. Then, after a large number of real-valued information messages are transmitted along the edges of the network (each data point is viewed as a node), a relevant set of exemplars and corresponding clusters is identified. Many detailed analyses of the AP approach have been carried out for various data sets with different scales. These studies show that for small data sets, there are only minor differences between traditional strategies and Affinity Propagation clustering for both precision and CPU execution time. Nevertheless, for large data sets, AP offers obvious advantages over existing methods. In particular, in their work, I showed that an improvement in execution time of roughly 100 times is achieved on data sets of more than 10,000 objects and 500 clusters. Moreover, it has been identified that the similarity measurement has a great influence on AP clustering.

III. SEEDS AFFINITY PROPAGATION

Based on AP method, we propose a novel method called "Seeds Affinity Propagation." The main new features of the new algorithm are: Tri-Set computation, similarity computation, seeds construction, and messages transmission. I start the presentation of the algorithm by explaining the basic similarity measurement used in my approach, i.e., three new feature sets, named by Cofeature Set (CFS), Unilateral Feature Set (UFS), and Significant Cofeature Set (SCS). The structural information of the text documents is included into the new similarity measurement. Then, I present how to extend the original AP approach with semisupervised learning strategy. The whole process of SAP is listed in Section 3.3

3.1 Similarity Measurement

In order to give specific and effective similarity measurement for our particular domain, i.e., text document, to introduce the following feature sets: the Cofeature Set, the Unilateral Feature Set, and the Significant Cofeature Set. To define these sets, first detail the computations of the new features. In my approach, each term in text is still deemed as a feature and each document is still deemed as a vector. However, all the features and vectors are not computed simultaneously, but one at a

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

time. I believe that this extended similarity measure can reveal both the difference and the asymmetric nature of similarities between documents. Moreover, to think that it is more effective in the application of Affinity Propagation clustering for text documents, image processing, gene detecting, and so on, since it is capable to deal with asymmetric problems. So I named the combination of this new similarity with conventional Affinity Propagation the Tri-Set Affinity Propagation (AP(Tri-Set)) clustering algorithm.

3.2 Seeds Construction

In semisupervised clustering, the main goal is to efficiently cluster a large number of unlabeled objects starting from a relatively small number of initial labeled objects. Given a few initial labeled objects, I would like to use them to construct efficient initial "seeds" for our Affinity Propagation clustering algorithm. I named the combination of semi supervised strategy with classical similarity measurement and conventional Affinity Propagation as Seeds Affinity Propagation with Cosine coefficient (SAP(CC)) clustering algorithm. By introducing both the seed construction method and the new similarity measurement into conventional AP, I arrive at the definition of the complete "Seeds Affinity Propagation algorithm," which will be detailed in the next section.

3.3 Seeds Affinity Propagation Algorithm

Based on the definitions of UFS, SCS, and the described seeds' construction method, the SAP algorithm is developed, following this sequence of steps:

1. Initialization:

2. Seeds construction: Constructing seeds from a few labeled objects according to Mean Features Selection

3. Tri-Set computation: Computing the (CFS), (UFS), and (SCS) between objects

4. Similarity computation: Computing the similarities among objects .

5. Self-Similarity computation: Computing the self-similarities for each object

6. Initialize messages: Initializing the matrixes of messages

7. Message matrix computation: Computing the matrixes of messages using (4) and (5).

8. Exemplar selection: Adding the two message matrixes and searching the exemplar for each object

9. Updating the messages using (6).

10. Iterating steps 6, 7, and 8 until the exemplar selection outcome stays constant for a number of iterations.

IV. EXPERIMENTS AND DISCUSSION

To examine the behavior and the performance of SAP algorithm, I have experimented on a widely used bench-mark text data. In order to compare the proposed SAP algorithm, I have performed the same clustering operation with two state-of-the-art clustering algorithms, namely, 1) k-means and 2) the original Affinity Propagation. Moreover, to further investigate the impact of the individual newly proposed contributions, I have also run Affinity Propagation algorithm using only the new Tri-Set Similarity metric (AP (Tri-Set)) and only seed construction semi supervised approach with the original

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

similarity measure (SAP (CC)). For the comparison of the obtained results, I have adopted three standard quality measurement parameters, namely, F-measure, entropy, and CPU execution time. In addition, I have also investigated the robustness of the proposed algorithm on different data distribution.

4.1 Experimental Setup

The publicly available Reuters-21578 (Reuters) data set is preclassified manually. This classification information is eliminated before the clustering processes, and is used to evaluate the clustering accuracy of each clustering algorithm at the end of the execution. The original Reuters data consist of 22 files (for a total of 21,578 documents) and contain special tags such as "<TOPICS>" and "<DATE>" among others. The preprocessing phase on the data set cuts the files into single texts and strips the document from the special tags. Then, those documents which belong to at least one topic are selected. For text clustering problem, Cofeature Set can be viewed as a two-tuples set. The terms in the Unilateral Feature Set, on the other hand, consist of the words. Moreover, there are some words that exist in the title, abstract, or in the first sentence of each paragraph.

4.2 Evaluation Measures

To evaluate the performance of clustering, three kinds of measures, F-measure, entropy, and CPU execution time, are used to compare the generated clusters with the set of categories created manually in Reuters. The F-measure is a harmonic combination of the precision and recall values used in information retrieval. Due to the higher accuracy of the clusters mapping to the original classes, the larger the F-measure, the better the clustering performance. Entropy provides a measure of the uniformity or purity of a cluster. The last metric—the CPU execution time—provides us a measure of the efficiency and scalability of the algorithm when large data set is used.

4.3 General Comparison

The experiments use the top 10 classes ("acq," "corn," "crude," "earn," "grain," "interest," "money-fx," "ship," "trade," and "wheat") extracted from Reuters, which have been widely used in the information retrieval area.

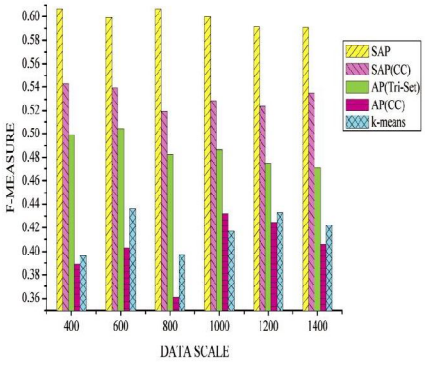


Fig.1. F-measure comparison.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

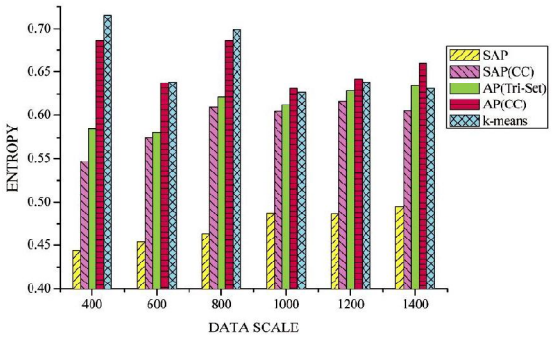


Fig.2. Entropy comparison.

For semisupervised learning strategy of SAP (CC) and SAP, four labeled documents are merged for a seed using the Mean Features Selection method and each class owns one seed. Figures show the comparisons for F-measure, Entropy, and CPU execution time for the five algorithms, respectively. In Fig. 2 and the summary results in Table 2, it can be seen that the average F-measure value of AP (CC) is close to that of k-means, while the average F-measures of AP (Tri-Set), SAP (CC), and SAP are 16.8, 27.6, and 43.9 percent higher than that of k-means, respectively.

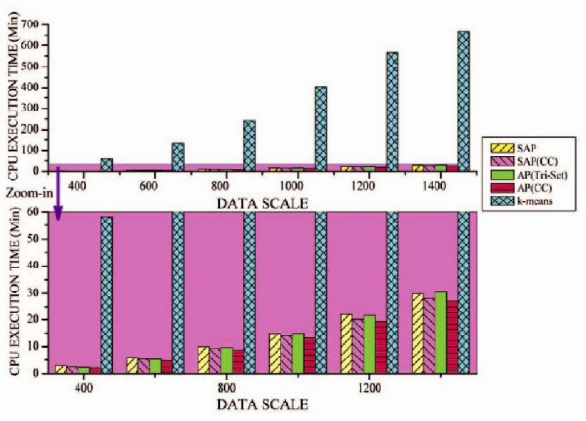


Fig.3 CPU execution time comparison.

Fig. 3 show an extremely different trend of the Entropy values for the five methods. Those of k-means and AP (CC) are close and they show the highest entropies; AP (Tri-Set) entropy is about 7.3 percent lower than that of k-means, on average; SAP (CC) is about 10.0 percent lower than k-means; the lowest one is SAP: it is 28.3 percent lower than that of k-means, on average, and 28.2 percent lower than the original AP(CC) algorithm.

From Fig. 3, it is clear that the CPU execution time of all Affinity Propagation-based algorithms—SAP, SAP (CC), AP (Tri-Set), and AP (CC)—is far less than that of k-means, and the gaps enlarge exponentially when the data set scale increases. For example, k-means consumes about 18.1 times larger than SAP for a 400-document data set, while this number is increased to 21.3 for a 1,400-document data set. The most significant advantage of SAP is that it is better than k-means in the foregoing evaluations, while k-means runs 200 times (the best run is used to compare with SAP) and costs

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

about 20-fold of SAP in time. This result also confirms the one in. Furthermore, even after 10,000 runs of k-means—with a size of 400 documents (F-measure: 0.406; Entropy: 0.677), we can't get similar results as SAP. To get the totally best result of k-means, it needs to execute all possible solutions. That is, at least k-means runs need to be performed for 400 documents.

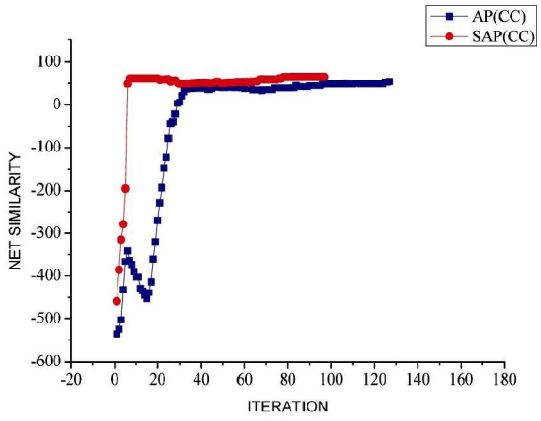
	Mean F-MEASURE	Mean Entropy	Mean CPU execution time
SAP	**0.599**	**0.472**	14.3
SAP (CC)	0.531	0.592	13.3
AP(Tri-Set)	0.486	0.610	14.1
AP(CC)	0.403	0.657	**12.6**
K-means	0.416	0.658	345.9

TABLE 1 :Mean Values over All Experiments

To exam the effectiveness of semisupervised strategy, we plot the net similarities curves of AP (CC) and SAP (CC). It takes 400 texts as an example and the net similarity of iteration each has been calculated. Net similarity is the objective function that AP tries to maximize. SAP (CC) uses less iterations and earns high net similarities. When they converge, SAP (CC) is 11.67 percent higher than. According to the discussion and figures above, it can safely draw the conclusion that SAP is superior to the other four algorithms. With the help of the new similarity measurement and the addition of the seeds, SAP greatly enhances the clustering performance. In addition, for a similar CPU execution time, SAP with both two contributions obtains higher F-measure than AP (CC), AP (Tri-Set), and SAP (CC). Finally, with the growth of data set, SAP has a steady advantage on both F-measure and Entropy.

4.4 Robustness Comparison

Compared with discrete uniform distribution, non uniform distribution of different categories is more familiar in the real world. Taking the Reuters as an example, the class



For all the three measurements and five different non uniform distribution data sets (focus on the "difficult" classes percentage), SAP performs better than k-means, AP (CC), AP (Tri-Set), and SAP (CC) although AP (CC) is the fastest one. From Table 5, the average F-measure value of AP (CC) and k-means is much similar (k-means is 5.2 percent higher than

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

AP (CC)). By analyzing the details of the clustering results, we could further conclude that SAP is more robust than the other four algorithms. Not only because it outperforms the other algorithms on evaluation measures, but also we found that SAP could catch the less represented topics. Based on all the experimental results above, an original analysis can be given: k-means is based on the objective function and searches for the minimum on coordinate descent.

	Case1	Case2	Case3	Case4	Case5	Mean
SAP	**0.325**	**0.463**	**0.484**	**0.492**	**0.487**	**0.450**
SAP (CC)	0.444	0.609	0.576	0.563	0.533	0.545
AP(Tri-Set)	0.486	0.621	0.626	0.609	0.594	0.587
AP(CC)	0.525	0.686	0.632	0.653	0.602	0.620
K-means	0.491	0.699	0.661	0.676	0.595	0.624

TABLE 2 :Entropy Comparison with Nonuniform Distribution

	Case1	Case2	Case3	Case4	Case5	Mean
SAP	8.7	9.9	11.1	10.0	9.8	9.9
SAP (CC)	8.3	9.3	10.1	9.0	8.9	9.1
AP(Tri-Set)	8.7	9.7	10.6	9.4	9.2	9.5
AP(CC)	**7.8**	**8.7**	**9.6**	**8.4**	**8.7**	**8.7**
K-means	226.3	242.4	218.7	184.1	178.2	209.9

TABLE 3: CPU Execution Time Comparison with Nonuniform Distribution (Min)

	Case1	Case2	Case3	Case4	Case5	Mean
SAP	**0.749**	**0.606**	**0.573**	**0.544**	**0.489**	**0.592**
SAP (CC)	0.662	0.519	0.511	0.450	0.385	0.505
AP(Tri-Set)	0.577	0.482	0.419	0.364	0.290	0.426
AP(CC)	0.450	0.361	0.392	0.314	0.225	0.348
K-means	0.518	0.397	0.368	0.280	0.269	0.366

TABLE 4:F-measure Comparison with Nonuniform Distribution

Due to the nature of greedy-descent algorithm, the search is led to the direction of energy reduction. In this case, k-means is easy to be trapped into a local minimum in which it could get stuck in a suboptimal solution or may not converge when the data contain many classes with different sizes. However, SAP calculates similarity using Tri-Set method which considers different feature sets, adopts the max-sum algorithm, adds dumped factor, and introduces seeds that can definitely lead the algorithm to converge more quickly to the correct direction. Therefore, SAP can more efficiently work out the solution and avoid suboptimal solutions.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

V. CONCLUSIONS

In this paper, I first proposed a similarity measurement which is extended from Cosine coefficient using structural information on the basis of Cofeature Set, Unilateral Feature Set, and Significant Cofeature Set. These three sets represent different features at different positions of texts. Their structural information improves the clustering results. The new similarity measurement can be used to calculate the asymmetric similarity directly, which is not limited to the symmetric space. Moreover, a new clustering algorithm which combines Affinity Propagation with semisupervised learning, namely, Seeds Affinity Propagation algorithm is proposed. SAP is applied to full text clustering which extends the application of Affinity Propagation. In the comparison with the classical clustering algorithm k-means, SAP not only reduces the computing complexity of text clustering and improves the accuracy, but also effectively avoids being random initialization and trapped in local minimum. SAP is also more robust and less sensitive to data distribution than k-means, conventional AP, AP (Tri-Set), and SAP (CC). In other words, it makes an important improvement in text clustering tasks. In addition, I believed that since SAP is based on a detailed similarity measurement and on a generic seeds construction strategy, it can be widely applied to other clustering problem domains. This is what I want to explore in our future work.

REFERENCES

1. Y.J. Li, C. Luo, and S.M. Chung, "Text Clustering with Feature Selection by Using Statistical Data," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 5, pp. 641-652, May 2008.

2. C. Buckley and A.F. Lewit, "Optimizations of Inverted Vector Searches," Proc. Ann. ACM SIGIR, pp. 97-110, 1985.

3. B.J. Frey and D. Dueck, "Clustering by Passing Messages between Data Points," Science, vol. 315, no. 5814, pp. 972-976, Feb. 2007.

4. G. Salton, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, 1989.

5. Z.H. Zhou and M. Li, "Semi-Supervised Regression with Co-Training Style Algorithms," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 11, pp. 1479-1493, Aug. 2007.

6. A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," Proc. 11th Ann. Conf. Computational Learning Theory, pp. 92-100, 1998.

7. T.Y. Jiang and A. Tuzhilin, "Dynamic Micro Targeting: Fitness-Based Approach to Predicting Individual Preferences," Proc. Seventh IEEE Int'l Conf. Data Mining (ICDM '07), pp. 173-182, Oct. 2007.

8. H.F. Ma, X.H. Fan, and J. Chen, "An Incremental Chinese Text Classification Algorithm Based on Quick Clustering," Proc. 2008 Int'l Symp. Information Processing (ISIP '08), pp. 308-312, May 2008.