DCCR: Document Clustering by Conceptual Relevance as a Factor of Unsupervised Learning

Annaluri Sreenivasa Rao Prof. S. Ramakrishna

Abstract: Present document clustering approaches cluster the documents by term frequency, which are not considering the concept and semantic relations during unsupervised or supervised learning. In this paper, a new unsupervised learning approach that estimates similarity between any two documents given by concept similarity measure is proposed. This novel method represents the concept as set of word sequences found in given documents. In regard to demonstrate the significance of the proposal we applied on set of benchmark document datasets.

Index Terms:- Document classification, document clustering, similarity measure, accuracy, classifiers, clustering algorithms.

1. INTRODUCTION

Clustering is unsupervised learning approach that partitions the data items into clusters. This unsupervised learning approach forms the clusters without using predefined class labels. Hence the number of clusters and documents in each cluster are dynamic. In contrast, the supervised learning approaches such as classification and prediction group the data by analyzing class-label data objects. The clustering analyzes given data and forms these labels dynamically. A similarity measure or similarity function is a real-valued function that enumerates the similarity between given any two elements.

2. RELATED WORK

In clustering algorithms and text classification the similarity measures are used to the maximum extent. For document clustering cosine similarity measure was adopted by the spherical k-means algorithm. [4] The label less documents clusters became very regular and available in this particular model. The text documents are projected as elevated dimensions and spares vectors by using words as features. In order to get unit Euclidean norm the algorithm outputs k disjoint clusters each with a conceptual vector which is the middle of the normalized cluster. For document clustering a cosine-based pair wise adaptive similarity measure was used by the Derrick Higgins. [5] The quality and pace of the neglected clustering are enhanced in pair wise adaptive similarity measure for elevated dimensional documents when compared with the original cosine similarity measure. By using the Kullback-leibler divergence a divisive data-theoretic feature clustering algorithm was proposed by Daphe koller [7] for text classification.

Annaluri Sreenivasa Rao is currently working as Associate Professor at Department of Computer Science, MRCE, Hyderabad, Telangana State, India Email: annaluri.rao@gmail.com
Prof. S. Ramakrishna is currently working as Professor at Department of Computer Science, Sri Venkateswara University, Tirupathi, Andhra Pradesh, India

In executing difficult learners like Support Vector Machines to the object of text classification we can prevent elevated dimensionality of text. In k-means like clustering algorithm squared Euclidean distance and relative entropy were combined by Kullback. [8] Recently emerging k means algorithm is developed to manage unit length document vectors specifically. Craven [9] performed document clustering, which is based on the Suffix Tree Document (STD) model that computes the pair-wise similarities of documents.

3. DOCUMENT CLUSTERING BY CONCEPTUAL RELEVANCE

Here in our proposed model, the given input documents will be clustered into minimum k clusters. Since the proposed model is an unsupervised learning approach, the class labels should be generated dynamically by using features extracted from the given input corpus.

3.1 The algorithmic approach of the DCCR:

- Let *TDC* be the text document corpus given as input for DCCR.
- Initially data preprocessing step will be applied that results processed documents word vector *pdwv* , which
  - Extracts words from each document and forms a row in a word vector *dwv*
  - For each row in word vector *dwv*
    - Remove stop words
    - Remove noise (non English terms and special symbols)
    - Apply Stemming on words of the each row

  After the preprocessing the word vector *dwv* turned to be the resultant documents word vector *pdwv*
- Next the word sequences of size *wst* will be considered as feature attributes from each row of the *pdwv* and forms

set of feature attributes *fas* with no duplicate elements. A word sequence is set of words with size *wst* appear in any row of *pdwv* in sequence.

- Then co-occurrence feature sets *cofs* such that each feature of feature set $\{fs \forall fs \in cofs\}$ is belong to *fas* will be formed and size of each set $\{fs \forall fs \in cofs\}$ can vary. The co-occurrence feature sets will be formed as follows
  - Initially one size feature sets will be formed and moved to *cofs*
  - And then two to max possible size co-occurrence feature sets will formed and moved to *cofs*
  - Then prunes co-occurrence feature sets as follows
    - If $\{fs_i \forall fs_i \in cofs\}$, $fs_i \subseteq \{fs_j \forall fs_j \in cofs\}$ and co-occurrence frequency of fs_i is identically equals to co-occurrence frequency of fs_j then fs_i will be pruned from *cofs*
- Then the co-occurrence feature sets of *cofs* will be ordered in descending order of their length and orders similar length co-occurrence feature sets in descending order of their co-occurrence frequency
- Then selects top k features sets as centroids. These set of top k centroids will be referred as *kcs* in further discussions.
- Further it performs clustering of the documents as follows:
  - For each document $\{d \forall d \in TDC\}$
    - Choose row wv_d from the *pdwv* that represents document *d*
    - For each centroid $\{c \forall c \in kcs\}$
      - Find similarity score $ss(wv_d, c)$ between wv_d and each centroid c.
      - Move the document *d* to the cluster that represented by a centroid c_i, such that similarity score between $ss(wv_d, c_i)$ is maximal when compared to the similarity score of document *d* with other centroids and $ss(vw_d, c_i)$ is greater than the given minimal similarity threshold *mst*.
      - If similarity score between document d and all centroids are less than *mst* then move document *d* to non cluster group *ncg*.
    - Continue the above process for all documents in *TDC*
  - If non cluster group is not empty then find new clusters as follows
    - Remove all from *kcs*
    - Consider left over feature sets with size greater than minimum centroid length threshold *mclt* of *cofs* as centroids and move to *kcs*
    - For each document $\{d \forall d \in ncg\}$
      - Choose row wv_d from the *pdwv* that represents document *d*
      - For each centroid $\{c \forall c \in kcs\}$
        - Find similarity score $ss(wv_d, c)$ between wv_d and each centroid c.
        - Move the document *d* to the cluster that represented by a centroid c_i, such that similarity score between $ss(wv_d, c_i)$ is maximal when compared to the similarity score of document *d* with other centroids.
          - Continue the above process for all documents in *ncg*

- Finally project documents of all clusters with their cluster id.

3.2 Pseudo code for DCCR

i. Main Process

Inputs:

TDC (text document corpus to be clustered)
k (minimal number of clusters to be formed)
mst (minimal similarity threshold)
mclt (minimal centroid length threshold)
wst (word sequence length threshold)

1. Begin:
2. Form a word vector *dwv* such that each row of the vector represents words in each document of the *TDC*
3. $pdwv \leftarrow preprocess(dwv)$
4. $fas \leftarrow findFAS(wst, pdwv)$
   // *fas* represents feature attributes (set of words in sequence of size *wst*) set
5. $cofs \leftarrow findCOFS(fas, pdws)$
   // The method invoked in step 5 can use any frequent item set mining algorithm such as fpgrowth [14] or éclat [15]. Due to space constraint that algorithm not explored in this article
6. Order *cofs* in descending order by size of each $\{fs \forall fs \in cofs\}$ and order all feature sets with similar size in descending order of their co-occurrence frequency
7. Choose top *k* feature sets from *cofs* such that each feature set length must be greater than *mclt* as centroids and refer them as *kcs*
8. Set non cluster group $ncg \leftarrow \phi$
9. $kClusters \leftarrow findClusters(TDC, pdwv, kcs, mst, ncg)$
10. $if\ (ncg \neq \phi)$ Begin
11. Remove all from *kcs*
12. Add leftover feature sets from *cofs* to *kcs*
13. $k'Clusters \leftarrow find\_k'Clusters(ncg, pdwv, kcs)$
14. End

15. *if* $(k'Clusters \neq \phi)$ $kClusters \leftarrow kClusters \bigcup k'Clusters$
16. End

ii. Preprocessing

1. *preprocess*(*dwv*) Begin
2. Set $pdwv \leftarrow \phi$
3. For each row *dr* of *dwv* Begin
4. Set $pdr \leftarrow \phi$
5. Remove non English characters from *dr*
6. Trim leading and trailing spaces of each word of *dr*
7. For each word *w* of *dr* Begin
8. *if* $(w \in sws)$ *then remove w from dr* //here *sws* is stop words set.
9. *else* Begin
10. apply stemming process on *w* and add *w* to *pdr*
11. Add *pdr* to *pdwv*
12. End
13. End
14. End
15. Return *pdwv*
16. End

iii. Finding feature attribute sets

1. *findFAS*(*wst*, *pdwv*) Begin
2. Set $fas \leftarrow \phi$
3. For each row *dr* of the *pdwv* Begin
4. For each word *w* of *dr* Begin
5. *if* $((index\_of(w) + wst) < size\_of(dr))$ Begin
6. *fas* ← word sequence of size *wst* begins from $index\_of(w)$
7. End
8. End
9. End
10. Return *fas*
11. End

iv. *Finding min k Clusters*

$find\_k'Clusters(TDC, pdwv, kcs, mst, ncg)$
1. Begin
2. For each document *d* in *TDC* Begin
3. Select row *dr* from *pdwv* that represents *d*
4. Set $s \leftarrow \phi$
5. Set $i \leftarrow \phi$
6. For each centroid c in *kcs* Begin
7. $ss(dr, c) \leftarrow \frac{dr \bigcap c}{c}$
//finding similarity score of document *d* and centroid *c*
8. *if* $(ss(dr, c) > s)$ Begin
9. set $s \leftarrow ss(dr, c)$
10. Set $i \leftarrow index\_of(c)$
11. End
12. End
13. *if* $(s \geq mst)$ Begin
14. Move document *d* into $cluster_{kcs[i]}$
15. End
16. Else Move document *d* to *ncg*
17. End
18. End

v. *Finding clusters from left over documents*

$find\_k'Clusters(ncg, pdwv, kcs)$
19. Begin
20. For each document *d* in *ncg* Begin
21. Select row *dr* from *pdwv* that represents *d*
22. Set $s \leftarrow \phi$
23. Set $i \leftarrow \phi$
24. For each centroid c in *kcs* Begin
25. $ss(dr, c) \leftarrow \frac{dr \bigcap c}{c}$
//finding similarity score of document *d* and centroid *c*
26. *if* $(ss(dr, c) > s)$ Begin
27. set $s \leftarrow ss(dr, c)$
28. Set $i \leftarrow index\_of(c)$
29. End
30. End
31. Move document *d* into $cluster_{kcs[i]}$
32. End
33. End

4. EXPERIMENTAL RESULTS

In this section, the effectiveness of the proposed Text Document Clustering By Conceptual Relevance (DCCR) as similarity factor of unsupervised learning is investigated. The investigation is done by applying the DCCR measure on Reuter's data.

4.1 Data set Exploration:

Reuters-21578 is collection of thousands of documents belongs to Reuter's newswire. This dataset is rich in volume and categorization.

4.2 Performance Analysis

From the corpus of the input dataset, we removed their actual cluster identification, and then applied DCCR over those documents with divergent word sequence length threshold and minimum number of clusters count. The performance of the DCCR is explored by clustering the given documents under divergent word sequence length threshold and minimum number of clusters required. Fig 1 indicates the accuracy of clusters formation under different values for *wst* .

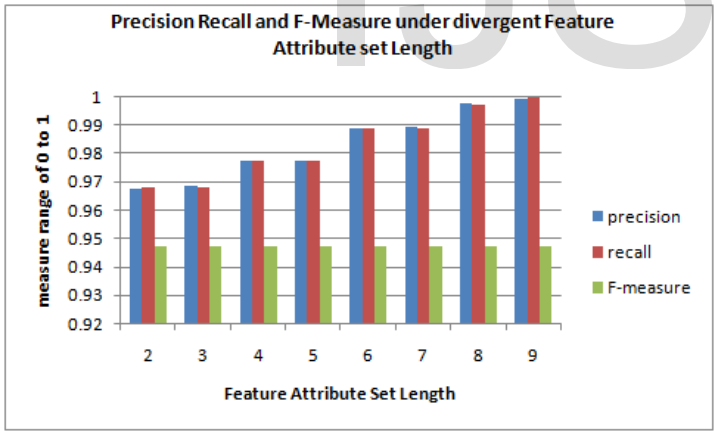

Fig 1: DCCR performance analysis under divergent feature attributes set size

Fig 2 indicates the influence of *k* (minimum number of clusters) value in accuracy. It is clear evident that DCCR performed well under max conceptual relevance factor (see fig 1). We limited the word sequence (concept) length to 9 since the beyond that value leads to create clusters with number of documents potentially low in each cluster. The given dataset is benchmarked with 90 categories of documents. As DCCR generates minimum clusters of count of k value, when k value is beyond the actual number of categories the performance of DCCR significantly falling to low (see fig 2 with k=100).

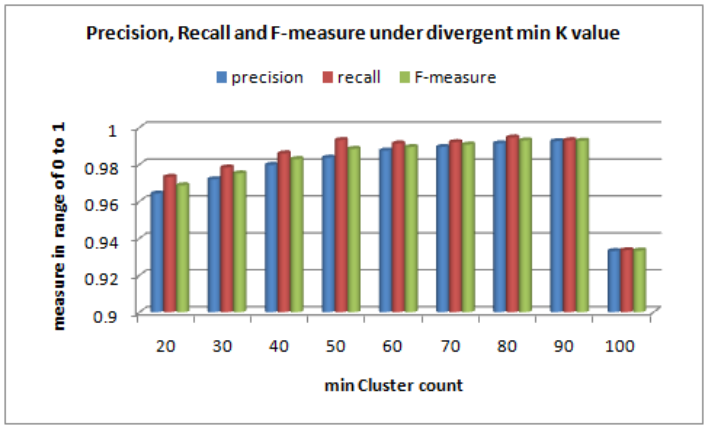

Fig 2: Performance Analysis of DCCR under divergent min cluster count

5. CONCLUSION AND FUTURE WORK

Here in this article we devised an unsupervised learning strategy that clusters text documents under conceptual relevance as clustering factor, which we referred as Document Clustering by Concept Relevance (DCCR) as factor of unsupervised learning. The said model is clustering the documents, which is based on their concept relevance rather than by using term frequency that used by many of the current state of the models available in literature. The performance analysis of the said model is indicating that it performs extremely well under optimal length of the feature attribute set and optimal *k* (minimum cluster count) value. The current work demands the domain knowledge to fix the word sequence length and minimum clusters required, as these two are the prime factors influencing the performance of the DCCR. In future, our work will focus on forming the concepts by the correlation analysis of the word tokens instead of forming concepts by the word sequences as we done in this present solution. Another direction of extending this work would be including the compound words to estimate the concept relevance. The better conceptual relevance representation can help to further improve the performance of unsupervised learning strategies in text document mining.

REFERENCES

[1]. Similarity Measures For Text Document Clustering Anna Huang 2008.

[2]. F. Sebastiani. Machine Learning In Automated Text Categorization. Acm Computing Surveys, 34(1):1-47, 2002.

[3]. S. Clinchant And E. Gaussier. Information-Based Models For Ad Hoc Ir. Proceedings Of 33rd Annual International Acm Sigir Conference On Research And

Development In Information Retrieval, Pages 234-241, 2010.

[4]. Sentence Similarity Measures For Essay Coherence Derrick Higgins Jill Burstein,2007

[5]. Similarity Measures for Short Segments of Text Donald Metzler1, Susan Dumais2, Christopher Meek,2007

[6]. Multi-Label Classification Algorithm Derived From K-Nearest Neighbor Rule With Label Dependencies Zoulficar Younes, Fahed Abdallah, And Thierry Denoeux,2003

[7]. Daphe Koller And Mehran Sahami, Hierarchically Classifying Documents Using Very Few Words, Proceedings Of The 14th International Conference On Machine Learning (Ml), Nashville, Tennessee, July 1997, Pages 170-178.

[8]. S. Kullback And R. A. Leibler. On Information And Sufficiency. Annuals Of Mathematical Statistics, 22(1):79-86, 1951.

[9]. H. Chim And X. Deng. Efficient Phrase-Based Document Similarity For Clustering. Ieee Transactions On Knowledge And Data Engineering, 20(9):1217 - 1229, 2008.

[10]. http://web.ist.utl.pt/ acardoso/datasets/.

[11]. http://www.cs.technion.ac.il/ ronb/thesis.html.

[12]. http://www.daviddlewis.com/resources/testcollections/reuters21578/

[13]. http://www.dmoz.org/

[14]. Yongmei Liu; Yong Guan, "FP-Growth Algorithm for Application in Research of Market Basket Analysis," Computational Cybernetics, 2008. ICCC 2008. IEEE International Conference on , vol., no., pp.269,272, 27-29 Nov. 2008

[15]. M.J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li; New Algorithms for Fast Discovery of Association Rules; Proc. 3rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'97, Newport Beach, CA), 283-296; AAAI Press, Menlo Park, CA, USA 1997