International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

Survey on Clustering of Text using COATES Methodology

Sneha S. Bhatkulkar†* and M. V. Vaidya†

Department of Information Technology, Shri Guru Gobind Singhji Institute of Engg & Technology, Nanded, India-431606

Accepted 25 May 2016, Available online 02 June 2016, **Vol.6, No.3 (June 2016)**

Abstract

In many text mining applications, side-information is available along with the text documents. Side-information, such as document provenance information, the links in the document, user-access behavior from weblogs, or other non-textual attributes which are present into the documents. Such attributes lead to better clustering results. However, the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy. We require a better way to perform the mining process, to maximize the advantages of side information. In this paper, we design an algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach.

Keywords: *Clustering, Data Mining*

1. Introduction

Generally, data mining refers to the process of analyzing data from different perspectives and summarizing it into useful information. Data Mining is mostly associated with the broader process of Knowledge Discovery in Databases (KDD). The rapidly increasing amounts of collections of data have led to an interest in introducing effective data mining algorithms. Clustering is the main task of exploratory data mining and a *cluster* is a collection of objects which are having similarity between them and are having dissimilarity to the objects belonging to other clusters. Lot of work has been done on clustering of text document and still going on (A. Banerjee *et al*, 2007; D. Cutting *et al*, 1992; H. Schutze *et al*, 1997; M. Steinbach *et al*, 2000; S. Zhong, 2005). The work mostly has been done for pure text clustering.

Mining text involves the process of parsing with some attribute features and removal of stop words and stemming process with structured data. Initial mining consists of text text categorization (preprocessing) text clustering (splitting into groups) Concept evolution production of new class. Objective of this methodology is to show the favorable features of using side information beyond a pure text clustering task.

2. Related Work

The known technique for clustering text is scatter gather technique (D. Cutting *et al*, 1992) under the combination of agglomerative and partitioned clustering. The related methods in text clustering are co-clustering methods for text data with expectation Maximization (EM) method (T. Liu *et al*, 2003) with text clustering is processed with Matrix-factorization techniques (W. Xu *et al*, 2003). Technique of document information based with clustering process, closely related to area of event tracking , topic-modeling, text-categorization is the context, with method in topic-driven clustering in text data was been proposed for text clustering the context in keyword extraction (C. C. Aggarwal *et al*, 2007; H. Frigui *et al*, 2004; M. Franz *et al*, and W. J. Zhu, 2008).

A hierarchical clustering is a method of cluster analyais which seeks to build a hierarchy of clusters. A measure of dissimilarity between sets of observations is required in order to decide which clusters should be combined for agglomerative or where a cluster should be split for divisive. CURE, a hierarchical clustering algorithm which is scalable and robust to outliers, and identifies clusters having non-spherical shapes and wide variances in size is given in (S. Guha *et al*, 1998). (T. Zhang *et al*, 1996) presents the hierarchical data clustering method BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) and it is used to perform hierarchical clustering over large data-sets. A comparative study of different clustering methods may be found in (M. Steinbach *et al*, 2000). Methods for text clustering in the context of keyword extraction are discussed in (M. Franz *et al*, 2001).In this way, the whole study of clustering techniques has been done in past time. In this paper, we will cover first an approach for using various kinds of meta-information along with text clustering. We will show the benefits of using such type of an approach over pure text clustering.

*Corresponding author: **Sneha S. Bhatkulkar**

3. Mining with Auxiliary Attributes

In this paper, we will discuss an approach for text clustering with auxiliary attributes (side information). Assumption is, we have corpus S of text documents. N be the total number of documents which are denoted by $T_1 \ldots T_N$. Assumption is, the set of distinct words in a corpus S denoted by W. Set of auxiliary attributes (side information) associated with the each document T_i is X_i. Each set of auxiliary attributes X_i has dimension d. Note that we are referring side information as auxiliary attribute or meta-information throughout the paper. Each side information is treated as binary for ease in notation and analysis. Attributes can be numerical or categorical. Because both of the above can be converted into binary format easily. Let's see the some of the examples of side-information which is referred as attributes. Examples are listed below:

- In an application in which we track user access behavior of web documents, the user-access behavior may be captured in the form of web logs. For each document, the meta-information may correspond to the browsing behavior of the different users. Such logs can be used to enhance the quality of the mining process in a way which is more meaningful to the user, and also application sensitive. This is because the logs can often pick up subtle correlations in content, which cannot be picked up by the raw text alone.
- Text documents, which can also be treated as attributes. Such links contain a lot of useful information for mining purposes. As in the previous case, such attributes may often provide insights about the correlations among documents in a way which may not be easily accessible from raw content.
- Many web documents have meta-data associated with them which correspond to different kinds of attributes such as the provenance or other information about the origin of the document. In other cases, data such as ownership, location or even temporal information may be informative for mining purposes.

Auxiliary attributes noted above, are quite sparse in many real time applications. Such types of sparsity can be a challenge from an effective purpose. Therefore our approach is designed to account for such sparsity. The problem statement for clustering with auxiliary attributes is given below.

Clustering with auxiliary information: *Determine a clustering of the documents into k clusters which are denoted by $C_1 \ldots C_K$ based on both the text content and the auxiliary variables from a given corpus S of documents by $T_1 \ldots T_N$ and a set of auxiliary variables X_i associated with document T_i.*

4. COATES Algorithm

In this section we will go through the procedure for text clustering used in this algorithm. COATES algorithm stands for *COntent and Auxiliary attribute based TExt*

cluStering algorithm. It is assumed that the number of clusters k is the input to the algorithm. And also assumption that preprocessing has been done i.e. stop words has been removed and stemming has been done so as to improve discriminatory power of the attributes. The algorithm undergoes two steps: First is initialization step and second one is main step.

In the first step of initialization we use a standard method of text clustering but without meta information. For this we used algorithm referred in (H. Schutze and C. Silverstein,1997).Because it provides a good initial starting point which is efficient. The output of this standard text clustering approach includes the centroids and partitioning created by clusters. This output provides starting point for the second step. This step does not have any use of meta-information. It is noted that this step is for pure text only.

The second step comes into action after the execution of first step. This step reconstructs the clusters formed above (using standard clustering algorithm) with the use of both pure text content and meta-information. This step performs two alternating iterations which uses both pure text content and auxiliary information so as to improve the quality of clustering. These two iterations are referred as *content* iterations and *auxiliary* iterations respectively. The specified algorithm maintains set of seed centroids, which are subsequently refined in each different iterations.

In content based iteration, we assign a document to its closest seed centroid using the text similarity function. $L_1 \ldots L_k$ are the centroids for k clusters created during this step. We are using cosine similarity function here. Cosine similarity is one of the most popular similarity measure applied to text documents.Cosine similarity of two vectors is computed by dividing the dot product of the two vectors by the product of their magnitudes.The cosine of the angle between the vectors ends up being a good indicator of similarity because at the closest the two vectors could be, 0 degrees apart, the cosine function returns its maximum value of 1.

In each auxiliary phase, we create a probabilistic model, which relates the attribute probabilities to the cluster-membership probabilities, based on the clusters which we have already been created in the most recent text-based phase. Before going into detail of this step, let's know some notations and terminologies which helps in understanding the algorithm more clearly.

Gini Index: The gini index provides a quantification of the discriminatory power of each attribute with

respect to clustering process. This is useful in removing noise attributes so as to have the robustness of approach. When there is large number of auxiliary attributes then it is most useful. We compute the gini index at the beginning of each auxiliary iteration. Gini index is computed as follows:

Let f_{rj} be the fraction of records in cluster C_j and the attribute r of value 1. Now we need to find the relative presence of attribute r in jth cluster. This is done as follows.

$$Prj = \frac{frj}{\sum_{m=1}^{k} frm} \quad (1)$$

The value of P_{rj} ranges from 0 to 1 over the attribute r and cluster j. We can say attribute values equally distributed among different clusters, when there is 1/k value of P_{rj}. Completely noisy attributes do not have any relevance to text content and not even useful for mining applications.While Auxiliary attributes may have different clustering behavior than textual attributes. Therefore, we would like the values of P_{rj} to vary across the different clusters. We can call it as Skew. The gini index is used to quantify the skew level. The gini index of attribute r is defined and denoted by:

$$Gr = \sum_{j=1}^{k} Prj^2 \quad (2)$$

Prior Probability: Depending on the first clustering algorithm, each auxiliary iteration has a prior probability of assignment of documents to clusters. This is required for constructing the probabilistic model of membership. Prior probability of document T_i belongs to cluster C_j is $P(T_i \in C_j)$. After execution of the content based clustering, the prior probability of auxiliary attributes to cluster is generated. Apriori value is nothing but the fraction of documents which have been assigned to cluster.

Posterior Probability: The posterior probabilities of cluster membership is noted at the end of the auxiliary attribute based iteration. We take the auxiliary attributes X_i which is present in T_i while computing the posterior probabilities $P(T_i \in C_j \mid X_i)$. The posterior probabilities is useful in re-adjusting the cluster centroids during the auxiliary iteration.

The algorithm COATES for clustering is given below.

```
Algorithm COATES(NumClusters: k, Corpus: T1…TN ,
  Auxiliary Attributes: X1….XN);
Begin
   Use content-based algorithm in [11] to create
      initial set of k clusters C1…Ck;
   Let centroids of C1…Ck be
      Denoted by L1…Lk;
     t=1;
    while not(termination_criterion) do
    begin
      { First minor iteration }
      Use cosine-similarity of each document Ti to
        centroids L1…Lk in order to determine the
          closest center to Ti and update the
            cluster assignments C1…Ck;
      Denote assigned cluster index for
        document Ti by qc(i, t);
      Update cluster centroids L1…Lk to the
        Centroids of updated clusters C1…Ck;
      { Second Minor Iteration}
      Compute gini-index of Gr for each auxiliary
       attribute r with respect to current
        clusters C1…Ck given in eq(2);
      Mark attributes with gini-index which is
       Γ standard-deviations below the
          mean as non-discriminatory;
        { for document Ti let Ri be the set of
          attributes which take on the value of 1, and for
          which gini-index is discriminatory;}
        for each document Ti , determine the posterior
        probability Pn(Ti∈ Cj| Ri );
        Denote qa(i, t) as the cluster-index with highest
        Posterior probability of assignment for document
Ti;
        Update cluster-centroids L1…Lk with the
         Use of posterior probabilities;

          t = t + 1;
        end
      ends
```

Fig.1 COATES algorithm.

5. Time complexity

We will find out the running time required for clustering purpose. Time complexity will be in the terms of number of clusters 'k' as input is number of clusters , the number of words present in the text d_t, the number of auxiliary attributes 'd' and total number of documents '*N*'. *O(k)* cosine distance computations are performed in each iteration. We need *O(N.k)* cosine computations for N documents. Since each cosine computation may require $O(d_t)$time, this running time is given by *O(N.k.dt).* In each iteration i.e. auxiliary and content-based iteration, we need to compute the similarity with the meta-information. Finally, the time needed for each iteration is
$O(N.\ k.\ (d+d_t\ ))$.

Conclusion

This survey paper includes a method of mining the data which is in the text format with the use of meta-information. A different type of applications contains different behavior of meta-information which might be used for improving the clustering process of text. If there is lot of meta-information then clustering with meta-information will be more efficient.

References

C. C. Aggarwal, Fellow, IEEE Yuchen Zhao, and P. S. Yu, Fellow, IEEE (June. 2014), On the Use of Side-Information

for Mining Text Data in IEEE Trans. Knowl. Data Eng. Vol. 26, No. 6.

D. Cutting, D. Karger, J. Pederson, and J. Turkey, Scatter/Gather (1992). A cluster-based approach to browsing large document collections, in Proc. ACM SIGIR Conf., New York, NY, USA, pp.318-329.

T. Liu, S. Liu, Z. Chen, and W.-Y. Ma (2003), An evaluation of feature selection for text clustering, in Proc. ICML Conf., Washington, DC, USA, pp. 488–495.

W. Xu, X. Liu, and Y. Gong (2003), Document clustering based on non-negative matrix factorization, in Proc. ACM SIGIR Conf., New York, NY, USA, pp. 267–273.

C. C. Aggarwal, S. C. Gates, and P. S. Yu (Feb. 2004), On using partial supervision for text categorization, IEEE Trans. Knowl. Data Eng., vol. 16, no. 2, pp. 245–255.

A. Banerjee and S. Basu (2007), Topic models over text streams: A study of batch and online unsupervised learning, in Proc. SDM Conf., pp. 437–442.

H. Frigui and O. Nasraoui (2004), Simultaneous clustering and dynamic keyword weighting for text documents, in Survey ofText Mining, M. Berry, Ed. New York, NY, USA: Springer, pp.45-70.

M. Franz, T. Ward, J. S. McCarley, and W. J. Zhu (2001), Unsupervised and supervised clustering for topic tracking, in Proc. ACM SIGIRConf., New York, NY, USA, pp. 310–317.

S. Guha, R. Rastogi, and K. Shim (1998), CURE: An efficient clustering algorithm for large databases, in Proc. ACM SIGMOD Conf., New York, NY, USA, pp. 73–84.

T. Zhang, R. Ramakrishnan, and M. Livny (1996), BIRCH: An efficient data clustering method for very large databases, in Proc. ACMSIGMOD Conf., New York, NY, USA, pp. 103–114.

H. Schutze and C. Silverstein (1997), Projections for efficient document clustering, in Proc. ACM SIGIR Conf., New York, NY, USA, pp. 74–81.

M. Steinbach, G. Karypis, and V. Kumar (2000), A comparison of docu-ment clustering techniques, in Proc. Text Mining Workshop KDD, pp. 109–110.

S. Zhong (2005), Efficient streaming text clustering, Neural Netw., vol. 18, no. 5–6, pp. 790–798.