ISSN 2395-1621

Clustering Approach For Mining Text Data

#1Akshay Bhujudade, #2Amol Dammewar, #3Ravindra Bondare

1akshaybhujugade65@gmail.com
2amoldammewar77@gmail.com
3ravindrabondare101@gmail.com

#1234PVPIT College of Engineering , Bavdhan, Pune-41

ABSTRACT

In many text mining applications, side-information is available along with the text documents. Such sideinformation may be of different kinds, such as document provenance information, the links in the document, user-access behavior from b logs, or other non-textual attributes which are embedded into the text document. Such attributes may contain a tremendous amount of information for clustering purposes. Hover, the relative importance of this side information may be difficult to estimate, especially when some of the information is noisy. In such cases, it can be risky to incorporate side-information into the mining process, because it can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, need a principled way to perform the mining process, so as to maximize the advantages from using this side information. In this paper, design an algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach. then show how to extend the approach to the classification problem. present experimental results on a number of real data sets inorder to illustrate the advantages of using such an approach.

ARTICLE INFO

Article History

Received :4th May 2016

Received in revised form :

6th May 2016

Accepted : 12th May 2016

Published online :

17th May 2016

***Keyword*:** mining, algorithm, partition algorithm, data mining.

I. INTRODUCTION

Clustering is the process in which data objects are organized into a set of disjoint classes called clusters. Objects that are in the same cluster are similar among themselves and dissimilar to the objects belonging to other clusters. The problem of clustering has been studied widely in the database and statistics literature in the context of a wide variety of data mining tasks. The clustering problem is defined as the process of finding groups for similar objects in the data. This similarity between the objects is measured by the use of a similarity function. Clustering is very useful in the text domain, where the objects to be clusters are of different sizes like documents, paragraphs, sentences or terms. Clustering is useful for organizing documents for improving retrieval and support browsing. Document clustering has been investigated for use in a number of different areas of text mining and information retrieval. Document clustering has also been used to automatically generate hierarchical clusters of documents. The problem of text clustering arises in the context of many application domains such as the social networks, web and other digital collections. The rapid increase in amount of text data in the context of these large online collections has led to an interest in creating scalable and effective mining algorithms. A large amount of work has been done in recent years on the problem of clustering in text collections. The survey of text clustering algorithms is given in [4]In many application domains, a large amount of side information is also associated along with the documents. Because text documents typically occur in the variety of applications in which there may be a large amount of other kinds of database attributes or metainformation which may be useful to the clustering process. Examples of such side-information are discussed below. The application in which the system tracks user access behaviour of web documents. The access behaviour of use may be captured in the form of web logs. For every document, the meta-information corresponds to the browsing behaviour of the different users. These logs can be used to enhance the quality of the mining process

which is more meaningful to the user, and also applicationsensitive. Because the logs can often pick up subtle correlations in content, which cannot be picked up by theraw text alone.Many text documents contain links in between them, which can also be treated as attributes. These links contain a lot of useful information for mining purposes. Many web documents have meta-data associated with them which correspond to different kinds of attributes such as the provenance or other information like ownership, location, or even temporal information about the origin of the document. In a number of network and user-sharing applications, documents which are associated with user-tags, may also be quite informative. Such side-information can be useful in improving the quality of the clustering process [1]. The primary goal of this paper is to study the clustering of data in which auxiliary information is available with text. Such scenarios are very common in a wide variety of data domains. Therefore, the paper extend the clustering approach to the problem of classification, which provides superior results because of the incorporation of side information. Goal of this paper is to show that the advantages of using side-information extend beyond a pure clustering task, and can provide competitive advantages for a wider variety of problem scenarios. The above section discusses the introduction of the clustering, classification and some examples of Meta information available with the documents. Section II describes the literature survey of classification and clustering. Sections III formalize the problem of text clustering with side information. Sections IV discuss how to extend these clustering techniques to the classification problem. Section V contains the conclusion.

II. LITERATURE REVIEW

A tremendous amount of work has been done in recentyears on the problem of clustering in text collections in the database and information retrieval communities. The Survey of Text Clustering Algorithms is studied in [3] [9]. There are many types of Distance-based Clustering Algorithms like Agglomerative and Hierarchical Clustering Algorithm, Distance-based Partitioning Algorithms like k-means clustering, A Hybrid Approach for clustering like scatter/gather technique etc. In Distance-based Clustering Algorithms there is a use of similarity function which measures the closeness between the text objects takes place. The most well-known similarity function which is used commonly in the text domain is the cosine similarity function. These similarity functions can be used in conjunction with a wide variety of traditional clustering algorithms like Agglomerative.

Hierarchical Clustering Algorithms

The general concept of agglomerative clustering is to successively merge documents into clusters based on
their similarity with one another. Almost all the hierarchical clustering algorithms successively merge groups based on the best pairwise similarity between these groups of documents. The main differences between these classes of methods are in terms of how this pairwise similarity is computed between the different groups of documents. Conceptually, the process of agglomerating documents into successively higher levels of clusters creates a cluster hierarchy for which the leaf nodes correspond to individual documents, and the internal nodes correspond to the merged groups of clusters. When two groups are merged, a new node is created in this tree corresponding to this larger merged group. The two children of this node correspond to the two groups of documents which have been merged to it Fig 1. Types of Text Clustering Techniques A hierarchical clustering algorithm called CURE that is more robust to outliers, and identifies clusters having non-spherical shapes and wide variances in size is given in [7]. Another hierarchical Clustering algorithm i.e. ROCK: A Robust Clustering Algorithm for Categorical Attributes for data with Boolean and categorical attributes is studied in [8]. It employs links and not distances for merging clusters. [11] Presents the hierarchical data clustering method BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) and it demonstrates that it is especially suitable for very large databases. The next method of document clustering is Distance based Partitioning algorithm. Partitioning algorithms are widely used in the database literature in order to efficiently create clusters of objects. K-means clustering algorithm is a partitioning algorithm [3]. It uses a set of k representatives, around which the clusters are built. In particular, K-means uses the notion of a centroid, which is the mean or median point of a group of points. Note that a centroid almost never corresponds to an actual data point. The simplest form of the k-means approach is to start off with a set of k seeds from the original corpus, and assign documents to these seeds on the basis of closest similarity. In the next iteration, the centroid of the assigned points to each seed is used to replace the seed in the last iteration. In other words, the new seed is defined, so that it is a better central point for this cluster. This approach is continued until convergence. Simultaneous Clustering and Dynamic Keyword Weighting for Text Documents takes place in [6]. It uses the approach to extend K-means algorithm, that in addition to partitioning the dataset into a given number ofclusters, also finds the optimal set of feature weights for each clusters. [12] Combines an efficient online spherical k-means (OSKM) algorithm with an existing scalable clustering strategy to achieve fast and adaptive clustering oftext streams. The OSKM algorithm modifies the spherical k-means (SPKM) algorithm, using online update (for cluster centroids) based on the well-known Winner-Take-
All competitive learning**.**
Text Clustering
Distance Based
Clustering
Hierarchical
Clustering
Distance
Partitioned
Clustering
K-Means
Clustering
Hybrid
Clustering
The third type of document clustering is the Hybrid Technique. Scatter-gather technique is the hybrid clustering technique [3] [5].A classic example of this is the Scatter/Gather method, which provides a systematic browsing technique with the use of clustered organization

of the document collection. Initially the system scatters the collection into a small number of document groups, or clusters, and presents short summaries of them to the user. Based on these summaries, the user selects one or more of the groups for further study. The selected groups are gathered together to form a sub collection. The system then applies clustering again to scatter the new sub collection into a small number of document groups, which are again presented to the user. With each successive iteration the groups become smaller, and therefore more detailed. Ultimately, when the groups become small enough, this process bottoms out by enumerating individual documents. The scatter-gather approach can be used for organized browsing of large document collections, because it creates a natural hierarchy of similar documents. However, all of these methods are designed for the case of pure text data, and do not work for cases in which the text-data is combined with other forms of data. Some limited work has been done on clustering text in the context of network-based linkage information like graph mining and algorithms of graph mining in [2] [10]. A wide variety of techniques have been designed for text classification in [4]. There are different techniques for classification of the data such as Probabilistic and Naïve Bayes Classifiers. Probabilistic classifiers are designed to use an implicit mixture model for generation of the underlying documents. This mixture model typically assumes that each class is a component of the mixture. Each mixture component is essentially a generative model, which provides the probability of sampling a particular term for that component or class. Next type of classifier is Decision Tree Classifiers. Decision tree is essentially a hierarchical decomposition of the (training) data space, in which a predicate or a condition on the attribute value is used in order to divide the data space hierarchically. The division of the data space is performed recursively in the decision tree, until the leaf nodes contain a certain minimum number of records, or some conditions on class purity. The majority class labelling the leaf node is used for the purposes of classification. Fig 2. Types of Text Classification Techniques Another type of classifier is SVM classifiers, the main principle of SVM is to determine separators in the search space which can best separate the different classes. SVM training algorithm builds a model that assigns new examples into one category or the other, making it a nonprobabilistic binary linear classifier. SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into highdimensional feature spaces. All this work is not applicable to the case of general side information attributes. This paper will provide a first approach to using other kinds of attributes in conjunction with text clustering. It will show the advantages of using such an approach over pure text-based clustering. Such an approach is especially useful, when the auxiliary information is highly informative, and provides effective guidance in creating more coherent clusters. It will also extend

III. CLUSTERING TECHNIQUE FOR THE SIDE INFORMATION

The focus of this paper is to show the advantages of using side-information for mining text data extend beyond a pure clustering task which provides competitive advantages for a wider variety of problem scenarios.
The algorithm used for clustering of side information is COATES Algorithm, which corresponds to the fact that it is a COntent and Auxiliary attribute based TExt cluStering algorithm. The algorithm requires two Phases

A. *Initialization*
It is a lightweight initialization phase in which a standard text clustering approach is used without any sideinformation. For this purpose, it uses the k-means clustering algorithm. The reason that this algorithm is used, it is a simple algorithm which can quickly and efficiently provide a reasonable initial starting point. The partitioning and the centroids created by the clusters formed in the first phase provide an initial starting point for the second phase. The first phase is based on text information only , not the auxiliary information.

B. *Main Phase*
This phase starts off with these initial groups, and iteratively reconstructs these clusters with the use of both the text content and the auxiliary information. Alternating iterations which use the text content and auxiliary attribute information in order to improve the quality of the clustering are performed in this step. These iterations are content iterations and auxiliary iterations respectively. The combination of the content iteration and auxiliary iteration is referred to as a major iteration. Each major iteration contains two minor iterations, which corresponding to the auxiliary and text-based methods respectively.

IV. SYSTEM ARCHITECTURE

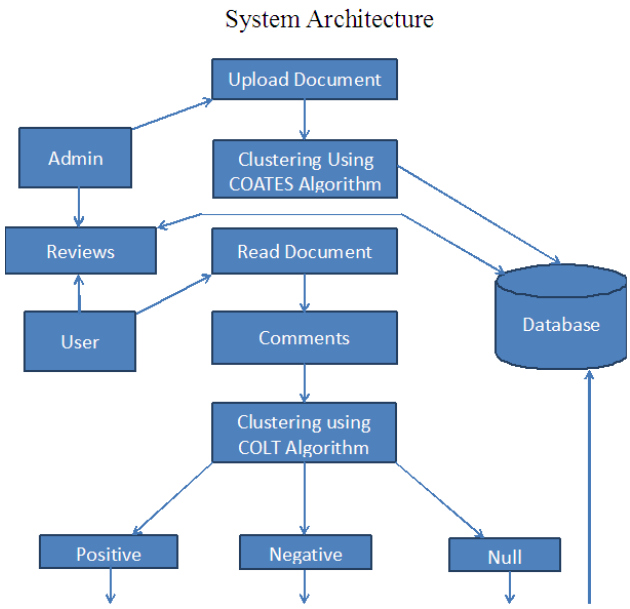

Fig:-System Architecture

Coates Algorithm:

Begin
Use content-based algorithm in 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
Using content-similarity of each document Ti to centroids L1 ... Lk in order to determine the closest cluster to Ti and update the cluster assignments C1 ...Ck;
Match word of document with dataset
Update cluster centroids L1 ... Lk to the centroids of updated clusters C1 .. .Ck;
Count the positive comments on the file.
End

COLT:

Begin
Perform feature selection on text with the use of class labels
Create initial set of k clusters denoted by C1 ...Ck, so that each cluster CZ contains only records of a particular class
Let centroids of C1 ...Ck be denoted by L1 ... Lk;
While not (review terminate) do begin
Use words matching of each comment Ti to centroids L1 ... Lk in order to determine the closest cluster to Ti
Denote assigned cluster index for document Ti,
Update cluster centroids L1 ... Lk to the centroids of updated clusters C1 ...Ck;
End

V. OUTPUT

Fig : Select File Form

Fig User Home

Fig : Show All User

VI. CONCLUSION

In this paper, we presented methods for mining text data with the use of side-information. Many forms of textdatabases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. In order to design the clustering method, we combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. This general approach is used in order to design both clustering and classification algorithms. We present results on real data sets illustrating the effectiveness of our approach. The results show that the use of side-information can greatly enhance the quality of text clustering and classification, while maintaining a high level of efficiency.

REFERENCES

[1] C. C. Aggarwal and P. S. Yu, “A framework for clustering massive text and categorical data streams,” in Proc. SIAM Conf. Data Mining, 2006, pp. 477–481.
[2] C. C. Aggarwal and H. Wang, Managing and Mining Graph Data. New York, NY, USA: Springer, 2010.
[3] S. Zhong, ―Efficient streaming text clustering,‖ Neural Netw.,vol. 18, no. 5–6, pp. 790–798, 2005.
[4] C. C. Aggarwal and C.-X. Zhai, Mining Text Data. New York, NY,USA: Springer, 2012.
[5] C. C. Aggarwal and C.-X. Zhai, “A survey of text classificationalgorithms,” in Mining Text Data. New York, NY, USA: Springer,2012.
[6] S. Guha, R. Rastogi, and K. Shim, ―ROCK: A robust clustering algorithm for categorical attributes,‖ Inf. Syst., vol. 25, no. 5, pp. 345–366, 2000.
[7] C. C. Aggarwal and P. S. Yu, “On text clustering with side information,” in Proc. IEEE ICDE Conf., Washington, DC, USA, 2012.
[8] R. Angelova and S. Siersdorfer, “A neighborhood-based approach for clustering of linked document collections,” in Proc. CIKM Conf., New York, NY, USA, 2006, pp. 778–779.

