MINING TEXT DATA USING SIDE INFORMATION

B.Chakradhar[1], B.Sreedhar[2]
Department of CSE, Sri Sivani College Of Engineering, Srikakulam, A.P
Department of CSE, Sree Vidyanikethan Engineering College, Tirupathi, A.P

Abstract

In this project side information plays a key role. In the existing system side information is used but applied on a pure clustering data rather than in the proposed system side information is used on some new techniques such as COATES and COLT. In many text mining applications, side-information is available along with the text documents. Such side-information may be of different kinds, such as document provenance information, the links in the document, user-access behavior from web 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. Side information is nothing that Metadata is "data about data". There are two metadata types of structural metadata, about the design and specification of data structures or "data about the containers of data"; and descriptive metadata about individual instances of application data or the data content.

1. Introduction to side information:

In this project side information plays a key role. In the existing system side information is used but applied on a pure clustering data rather than in the proposed system side information is used on some new techniques such as COATES and COLT.

In many text mining applications, side-information is available along with the text documents. Such side-information may be of different kinds, such as document provenance information, the links in the document, user-access behavior from web 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. Side information is nothing that Metadata is "data about data". There are two metadata types of **structural metadata**, about the design and specification of data structures or "data about the containers of data"; and **descriptive metadata** about individual instances of application data or the data content.

The capacity of the degraded relay channel is known when side information at both the source and relay is identical. However, in many naturally interesting cases the source and relay may not have, or may not be able to exploit, exactly the same side information. This may occur either when side information is privileged information that is available only at one of the nodes (e.g. watermarking problems) or when the relay does not have the ability to collect or utilize full information about channel state and/or interference. In this paper we calculate capacity expressions for degraded discrete memory less relay channels where the source and relay do not have identical side information. As a special case we consider the case where the relay has no side information.

2. Extension to Classification

This Module is divide database into the number of clusters by using the supervised K-means algorithm. The supervised K-means algorithm performs the divides the database into the clusters and purified the cluster into purified clusters.

The K-means clustering algorithm is modified, so that each cluster only contains records of a particular class.

3. COATES Algorithm

COATES is stands for “Content and auxiliary attribute based text clustering algorithm”. This Module is converts purified clusters into the

supervised clusters by using the cosine similarity, gini index and posterior probability.

Here cosine similarity performs add closest cluster to the cluster group. Gini index performs the find impurity levels in the each and every clusters and posterior probability performed. It provides output as the supervised clusters.It contains the two methods those are:

a) An efficient projection based clustering approach which adapts the k-means approach to text. This approach is widely known to provide excellent clustering results in a very efficient way. We refer to this algorithm as SchutzeSilverstein [text only].

b) We adapt the k-means approach with the use of both text and side information directly. We refer to this baseline as K-Means.

3.1 Implementation of the COATES Algorithm

Algorithm COATES (NumClusters: k, Corpus: $T_1 \ldots T_N$, Auxiliary Attributes: $X_1 \ldots X_N$);
begin
Use content-based algorithm in [27] to create initial set of k clusters $C_1 \ldots C_k$;
Let centroids of $C_1 \ldots C_k$ be denoted by $L_1 \ldots L_k$;
$t = 1$;
while not(termination criterion) do
begin
{ First minor iteration }
Use cosine-similarity of each document Ti to centroids $L_1 \ldots L_k$ in order to determine the closest cluster to Ti and update the cluster assignments $C_1 \ldots C_k$;
Denote assigned cluster index for document Ti by $qc(i, t)$;
Update cluster centroids $L_1 \ldots L_k$ to the centroids of updated clusters $C_1 \ldots C_k$;
{ Second Minor Iteration }
Compute gini-index of Gr for each auxiliary attribute r with respect to current clusters $C_1 \ldots C_k$;
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 use the method discussed in section 2 to determine the posterior probability $Pn(Ti \in Cj \mid Ri)$;
Denote $qa(i, t)$ as the cluster-index with highest posterior probability of assignment for document Ti;
Update cluster-centroids $L_1 \ldots L_k$ with the use of posterior probabilities.

4. COLT Algorithm

COLT is stand on "content and auxiliary attribute based text classification algorithm". This Module is a classification module it is performs the after COATES. It takes input as supervised clusters and returns majority class labels by using the cosine similarity.

Here cosine similarity performs the remove non zero attributes which are related to the class label. Then it provides the majority class label. The COLT contains the three methods those are:

a) We tested against a *Naive* Bayes Classifier which uses only text.
b) We tested against an SVM classifier which uses only text.
c) We tested against a supervised clustering method which uses both text and side information.
d) Supervised clusters as output.

4.1 Implementation of the COLT Algorithm

Algorithm *COLT*(NumClusters: k, Corpus: $T_1 \ldots T_N$,
Auxiliary Attributes: $X_1 \ldots X_N$ Labels $l_1 \ldots l_N$);
begin
Perform feature selection on text and auxiliary attributes with the use of class labels and gini index as explained in section 3;
Use supervised version of algorithm in [27] to create initial set of k clusters denoted by $C_1 \ldots C_k$, so that each cluster Ci contains only records of a particular class;
Let centroids of $C_1 \ldots C_k$ be denoted by $L_1 \ldots L_k$;
$t = 1$;
while not(termination criterion) do
begin
{ First minor iteration }
Use cosine-similarity of each document Ti to centroids $L_1 \ldots L_k$ in order to determine the closest cluster to Ti (which belongs to same class) and update the cluster assignments $C_1 \ldots C_k$;
Denote assigned cluster index for document Ti by $qc(i, t)$;
Update cluster centroids $L_1 \ldots L_k$ to the centroids of updated clusters $C_1 \ldots C_k$;
{ Second Minor Iteration }
Compute gini-index of Gr for each auxiliary attribute r with respect to current clusters $C_1 \ldots C_k$;

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* use the method discussed in section 2 to 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* which also belongs to the same class;
Update cluster-centroids $L_{1.....}L_k$ with the use of posterior probabilities as discussed in section 2;
$t = t + 1$;
end
end

4.2 Algorithm COLT Classify

Clusters: *C*1 . . . *k*, Test Instance: *T*i, Auxiliary Attributes of Test Instance: *X* i)
begin
Determine top *r* closest clusters in $C_1 \ldots C_k$ to *T*i based on cosine similarity with the text attributes;
Derive the set *R*i from *X*i , which is the set of non-zero attributes in *X*i ; Compute *Ps*(*TI* €*Cj* |*R*i) with the use of Equation 8;
Determine top *r* clusters in $C_{1.....}C_k$ to *X*i based on the largest value of *Ps*(*T*i€ *C*j |*R*i);
Determine the majority class label from the 2 · *r* labeled clusters thus determined; return majority label;
end

5. System Architecture

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

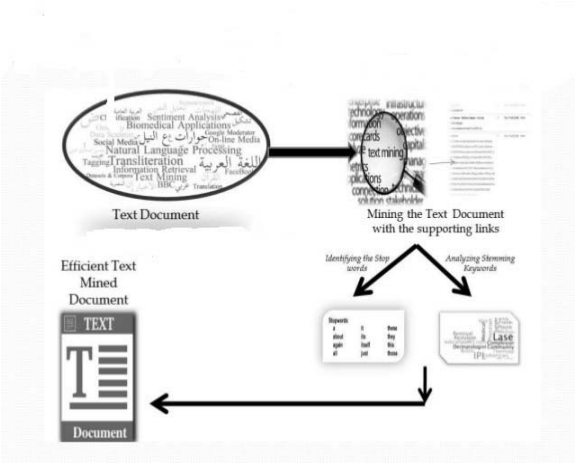
System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

On Using Side Information Mining Text Data high level architecture is shown in below. The architecture mainly in three parts those are:

- **a)** Text Document
- **b)** Mining the text document
- **c)** Efficient Text Mined Document

These three parts are represents the input of the project, applying algorithms to the documents, and final results of the project those are shown in the following figure.

5.1 Architecture Diagram



5.1.1 Text Document

The Text Document contains the lot of information in different languages like English, Hindi, Urdu, Tamil, etc. that means the data with side information is taking as input to the mining process.

5.1.2Mining the Text Document

Mining the data from text document we are getting only related data to class label. Here perform the two operations those are identifying stop words and analyzing the stemming keywords.

5.1.3 Efficient Text Mined Document

The Document contains purified data only. It is a final result for mining process and getting efficient value.

6. Results

Here we are selecting any type of data set for example we select a “Toy Dataset”. The Toy dataset contains the total details about internet that means how many members are used and those are which age group and they are browse

which type of data. The following figures are shows the how we are execute the Toy dataset and finally which type of output we are get.

1. On the Use of Side Information for Mining Text Data results.

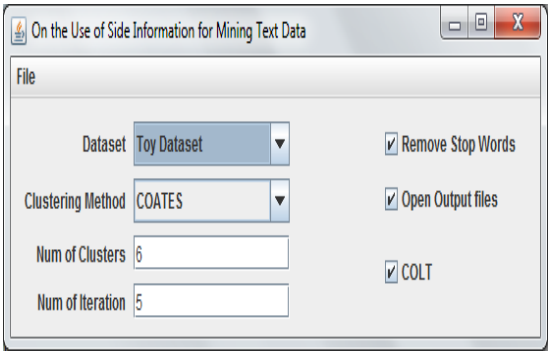


Figure.6.1. Initial Execution frame for result.

Here we are select the dataset with side information and unnecessary data. And enter the how many number of clusters and iteration we want to perform.

2. Selecting Files for Results are shown is Figure.

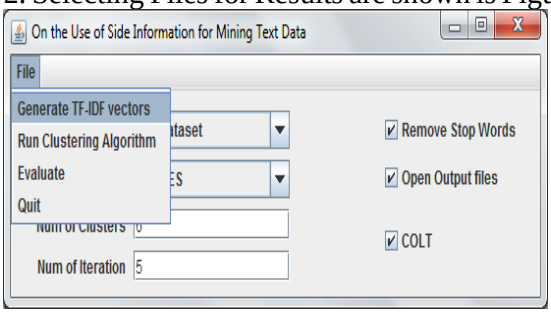


Figure.6.2. Selecting file for running

Go to the file and select which we want to execute are knows about the dataset files. Select Generate TF-IDF vectors for given dataset.

3. Generate TF-IDF vectors in notepad.

Figure.6.3. TF-IDF vectors for the Toy dataset.

We observe the TF-IDF vectors with frequency of that particular vectors with different fields are parts of data.

4. Run clustering algorithm result in notepad.

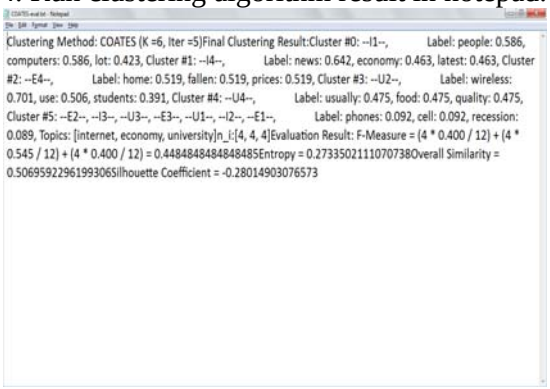


Figure.6.4. Number of clusters for the Toy dataset.

Here apply clustering algorithm that is COLT. Its generates the similar type of information will be merge in one cluster. That means similar attributes are combined in same cluster. And generates some of the clusters.

5. Evaluating results in notepad.

.

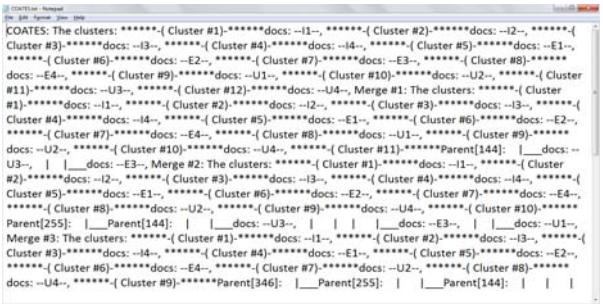


Figure.6.5. Co-efficient value of the Toy dataset.

Finally here we are get co efficient value with number of documents in the clusters.

7. Conclusion

We presented methods for mining text data with the use of side-information. Many forms of text-databases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. we add side information to increase the data efficiency because now a day's rapid increasing of data is high to avoid those unnecessary data from the pages we use side information. It can be applicable to all types of datasets like text data, images, any type of data. 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. These two algorithms are used to avoid unwanted data and to increase the efficiency. 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 with less Co-efficient value.

8. References

[1] C. C. Aggarwal and H. Wang, *Managing and Mining Graph Data*,Springer, 2010.

[2] C. C. Aggarwal, *Social Network Data Analytics*, Springer, 2011.

[3] C. C. Aggarwal and C. -X. Zhai, *Mining Text Data*, Springer, 2012.

[4] C. C. Aggarwal and C.-X. Zhai, "A Survey of Text Classification Algorithms," Book Chapter in *Mining Text Data*, Springer, 2012.

[5] C. C. Aggarwal and P. S. Yu, "A Framework for Clustering Massive Text and Categorical Data Streams," in *SIAM Conf. on Data Mining*, pp. 477–481, 2006.

[6] C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," in *IEEE TKDE*, vol. 16(2), pp. 245–255, 2006.

[7] C. C. Aggarwal, and P. S. Yu., "On text clustering with side information," in *IEEE ICDE Conference*, 2012.

[8] R. Angelova and S. Siersdorfer, "A neighborhood-based approach for clustering of linked document collections," in *CIKM Conf.*, pp. 778–779, 2006.

[9] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in *SDM Conf.*, pp. 437–442, 2007.

[10] J. Chang and D. Blei, "Relational Topic Models for Document Networks," in *AISTASIS*, pp. 81–88, 2009.

[11] D. Cutting, D. Karger, J. Pedersen, and J. Tukey, "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," in *ACM SIGIR Conf.*, pp. 318–329, 1992.

[12] I. Dhillon, "Co-clustering Documents and Words using bipartite spectral graph partitioning," in *ACM KDD Conf.*, pp. 269–274, 2001.

[13] I. Dhillon, S. Mallela and D. Modha, "Information-theoretic Co-Clustering," in *ACM KDD Conf.*, pp. 89–98, 2003.

[14] P. Domingos, M. J. Pazzani. "On the optimality of the simple Bayesian classifier under zero-one loss," in *Machine Learning*, 29(2–3), pp. 103–130, 1997.

[15] M. Franz, T. Ward, J. S. McCarley, and W. J. Zhu, "Unsupervised and supervised clustering for topic tracking," in *ACM SIGIR Conf.*, pp. 310–317, 2001.

[16] G. P. C. Fung, J. X. Yu, and H. Lu, "Classifying text streams in the presence of concept drifts," in *PAKDD Conf.*, pp. 373–383, 2004.

[17] H. Frigui and O. Nasraoui, "Simultaneous clustering and dynamic keyword weighting for text documents, Survey of text mining," Michael Berry, Ed, *Springer*, pp. 45–70, 2004.

[18] S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases," in *ACM SIGMOD Conf.*, pp. 73–84, 1998.

[19] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," in *Inf. Syst.*, vol. 25(5), pp. 345–366, 2000.

[20] Q. He, K. Chang, E.-P. Lim and J. Zhang, "Bursty feature representation for clustering text streams," in *SDM Conf.*, pp. 491–496, 2007.

[21] A. Jain and R. Dubes, *Algorithms for clustering data*, Prentice-Hall, Inc., 1988.

[22] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, "An evaluation of feature selection for text clustering," in *ICML Conf.*, pp. 488–495, 2003.

[23] A. McCallum, "Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering," *http://www.cs.cmu.edu/ mccallum/bow*, 1996.

[24] Q. Mei, D. Cai, D. Zhang and C.-X. Zhai, "Topic Modeling with Network Regularization," in *WWW Conf.*, pp. 101–110, 2008.

[25] R. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," in *VLDB Conf.*, pp. 144–155, 1994.

[26] G. Salton, *an Introduction to Modern Information Retrieval*, McGraw Hill, 1983.

[27] H. Schutze and C. Silverstein, "Projections for Efficient Document Clustering," in *ACM SIGIR Conf.*, pp. 74–81, 1997.

[28] F. Sebastiani, "Machine Learning for Automated Text Categorization," *ACM Computing Surveys*, 34(1), 2002.

[29] C. Silverstein and J. Pedersen, "Almost-constant time clustering of arbitrary corpus sets," in *ACM SIGIR Conf.*, pp. 60–66, 1997.

[30] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Text Mining Workshop, KDD*, pp. 109–110, 2000.

[31] Y. Sun, J. Han, J. GAO, and Y. Yu, "iTopicModel: Information Network Integrated Topic Modeling," in *ICDM Conf.*, pp. 493–502, 2009.

[32] W. Xu, X. Liu, and Y. Gong, "Document clustering based on nonnegative matrix factorization," in *ACM SIGIR Conf.*, pp. 267–273, 2003.

[33] T. Yang, R. Jin, Y. Chi, and S. Zhu, "Combining link and content for community detection: a discriminative approach," in *ACM KDD Conf.*, pp. 927–936, 2009.

[34] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *ACM SIGMOD Conf.*, pp. 103–114, 1996.

[35] Y. Zhao and G. Karypis, "Topic-driven clustering for document datasets," in *SIAM Conf. on Data Mining*, pp. 358–369, 2005.

[36] Y. Zhou, H. Cheng and J. X. Yu, "Graph clustering based on structural/ attribute similarities," *PVLDB*, vol. 2(1), pp. 718–729, 2009.

[37] S. Zhong, "Efficient Streaming Text Clustering," *Neural Networks*, vol18, Issue 5–6, pp. 790–798, 2005.