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. 3, Issue 2, February 2015

Side Information Gathering for Mining Text Data

Naveena.M[1], Karthik.R[2], Balaji.M[3]

P.G. Scholars, Department of CSE, Karpagam University, Coimbatore, India[1, 3]

Assistant Professor, Department of CSE, Karpagam University, Coimbatore, India [2]

ABSTRACT: 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 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 in order to illustrate the advantages of using such an approach.

KEYWORDS: Text mining, clustering, Report Generation

I.INTRODUCTION

The rapidly increasing amounts of text data in the context of these large online collections has led to an interest in creating scalable and effective mining algorithms. In many application domains, a tremendous amount of side information is also associated along with the documents. This is because text documents typically occur in the context of a variety of applications in which there may be a large amount of other kinds of database attributes or Meta information which may be useful to the clustering process. Many text documents contain links among them, 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 b 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. In a number of network and user-sharing applications, documents may be associated with user-tags, which may also be quite informative. While such side-information can sometimes be useful in improving the quality of the clustering process, it can be a risky approach when the side-information is noisy. In such cases, it can actually worsen the quality of the mining process. Therefore, will use an approach which carefully ascertains the coherence of the clustering characteristics of the side information with that of the text content. This helps in magnifying the clustering effects of both kinds of data. The core of the approach is to determine a clustering in which the text attributes and side-information provide similar hints about the nature of the underlying clusters, and at the same time ignore those aspects in which conflicting. While our primary goal in this paper is to study the clustering problem, note that such an approach can also be extended in principle to other data mining problems in which auxiliary information is available with text. Such scenarios are very common in a wide variety of data domains. Therefore, will also propose a method in this paper in order to extend the approach to the problem classification. Will show that the extension of the approach to the classification problem provides superior results because of the incorporation of side information. Our goal 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. This paper is organized as follows. The remainder of this section will present the related work on the topic. In the next section, will formalize the problem of text clustering with side information. Will also present an algorithm for the clustering process.

II. RELATED WORK

Text clustering has been studied widely by the database community the major focus of this work has been on scalable clustering of multi dimensional data of different a general survey of clustering algorithms may be found. The problem of clustering has also been studied quite extensively in the context of text data. A survey of text clustering methods may be found in. One of the most ll known techniques for text-clustering is the scatter gather technique [1] which uses a combination of agglomerative and partitioned clustering. Other related methods for text clustering which use similar methods are discussed. Co-clustering methods for text data are proposed in [2]. An Expectation Maximization (EM) method for text clustering has been proposed. Matrix-factorization techniques for text clustering are pro- posed in [3]. In this paper, will provide a first approach to using other kinds of attributes in conjunction with text clustering. 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. will also extend the method to the problem of text classification, which has been studied extensively in the literature.

III. CLUSTERING WITH SIDE INFORMATION

Will discuss an approach for clustering text data with side information. Assume that have a corpus S of text documents. The total number of documents is N, and they are denoted by $T_1 \dots T_N$. It is assumed that the set of distinct words in the entire corpus S *is* denoted by W. Associated with each document T_i, have a set of side attributes X_i. Each set of side attributes X_i has d dimensions, which are denoted by $(x_{i1} \dots x_{id})$. Refer to such attributes as *auxiliary* attributes. For ease in notation and analysis, assume that each side-attribute x_{id} *is* binary, though both numerical and categorical attributes can easily be converted to this format in a fairly straightforward way. This is because the different values of the categorical attribute can be assumed to be separate binary attributes, whereas numerical data can be discredited to binary values with the use of attribute ranges. Note that our technique is not restricted to binary auxiliary attributes, but can also be applied to attributes of other types. When the auxiliary attributes are of other types (quantitative or categorical), they can be converted to binary attributes with the use of a simple transformation process. For example, numerical data can be discredited into binary attributes. Even in this case, the derived binary attributes are quite sparse especially when the numerical ranges are discredited into a large number of attributes. In the case of categorical data, can define a binary attribute for each possible categorical value. In many cases, the number of such values may be quite large. Therefore, will design our techniques under the implicit assumption that such attributes are quite sparse.

Text Clustering with Side Information will use the auxiliary information in order to pro- vide additional insights, which can improve the quality of clustering. In many cases, such auxiliary information may be noisy, and may not have useful information for the clustering process. Therefore, will design our approach in order to magnify the coherence beten the text content and the side-information, when this is detected. In cases, in which the text content and side-information do not show coherent behavior for the clustering process, the effects of those portions of the side-information are marginalized.

IV. PROPOSED ALGORITHM

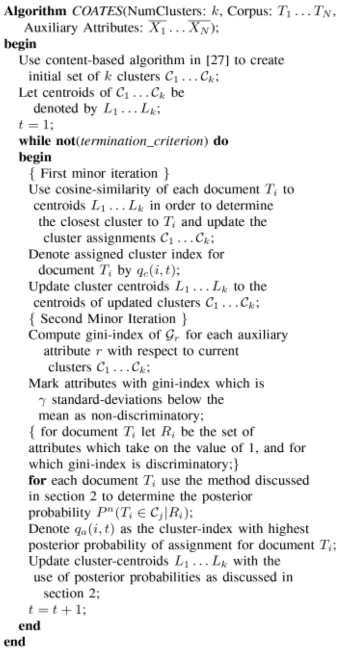
COATES Clustering Algorithm

Will describe our algorithm for text clustering with side information. Refer to this algorithm as COATES throughout the paper, which corresponds to the fact that it is Content *and Auxiliary attribute based Text clustering* algorithm. Assume that an input to the algorithm is the number of clusters k. As in the case of all text-clustering algorithms, it is assumed that stop-words
Have been removed, and stemming has been performed in order to improve the discriminatory poor of the attributes. The algorithm requires two phases:

Initialization: use a highlight initialization phase in which a standard text clustering approach is used without any side-information. Provide a reasonable initial starting point. The cancroids and the partitioning created by the clusters formed in the first phase provide an initial starting point for the second phase. Note that the first phase is based on text only, and does not use the auxiliary information.

Main Phase: The main phase of the algorithm is executed after the first 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. This phase performs alternating iterations which use the text content and auxiliary attribute information in order to improve the quality of the clustering. call these iterations as content iterations and auxiliary iterations respectively. The combination of the two iterations is referred to as a major iteration. Each major iteration thus contains two minor iterations, corresponding to the auxiliary and text-based methods respectively. The focus of the first phase is simply to construct an initialization, which provides a good starting point for the clustering process based on text content. Since the key techniques for content and auxiliary information integration are in the second phase, will focus most of our subsequent discussion on the second phase of the algorithm. The first phase is simply a direct application of the text clustering algorithm proposed in [4]. The overall approach uses alternating minor iterations of content-based and auxiliary attribute-based clustering. These phases are referred to as content-based and auxiliary attribute-based iterations respectively. The algorithm maintains a set of seed cancroids, which are subsequently refined in the different iterations. In each content-based phase, assign a document to its closest seed cancroids based on a text similarity function. The goal of this modeling is to examine the coherence of the text clustering with the side-information attributes. Before discussing the auxiliary iteration in more detail, will first introduce some notations and definitions which help in explain- ing the clustering model for combining auxiliary and text variables. Assume that the k clusters associated with the data are denoted by $C_1 \dots C_k$. In order to construct a probabilistic- tic model of membership of the data points to clusters, assume that each auxiliary iteration has a prior probability of assignment of documents to clusters (based on the execution of the algorithm so far), and a posterior probability of assignment of documents to clusters with the use of auxiliary variables in that iteration. Since are focusing on sparse binary data, the value of 1 for an attribute is a much more informative event than the default value of 0. Therefore, it suffices to condition only on the case of attribute values taking on the value of 1. For example, let us consider an application in which the auxiliary information corresponds to users which are browsing specific b pages. In the next *content-based* iteration, assign the documents to the modified cluster-cancroids based on the cosine similarity of the documents to the cluster cancroids [5]. Each document is assigned to its closest cluster cancroids based on the cosine similarity. The assigned documents are then aggregated in order to create a new cancroids Meta document which aggregates the frequency of the words in the documents for that cluster.

A key issue for the algorithm is the convergence of the algorithm towards a uniform solution. In order to compute convergence, assume that have an identifier associated with each cluster in the data. This identifier does not change from one iteration to the next for a particular cancroids .Within the *t*th major iteration; compute the following quantities for each document for the two different minor iterations: Compute the cluster identifier to which the document *Ti* was assigned in the content-based step of the *t*th major iteration. This is denoted by *qc(i, t)*. Compute the cluster identifier to which the document *Ti* had the highest *probability of assignment* in the auxiliary-attribute set of the *t*th major iteration. This is denoted by *qa(i, t)*. In order to determine when the iterative process should terminate, would like the documents to have assignments to similar clusters in the $(t - 1)$th and *t*th steps at the end of both the auxiliary-attribute and content-based steps, An important point to be remembered is that the output to the algorithm the clustering process is inherently designed to converge to clusters which use both content and auxiliary information some of the documents cannot be made to agree in the clustering behavior with the use of different criteria.



```
Algorithm COATES(NumClusters: $k$, Corpus: $T_1 \ldots T_N$,
    Auxiliary Attributes: $\overline{X_1} \ldots \overline{X_N}$);
begin
  Use content-based algorithm in [27] to create
    initial set of $k$ clusters $\mathcal{C}_1 \ldots \mathcal{C}_k$;
  Let centroids of $\mathcal{C}_1 \ldots \mathcal{C}_k$ 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 $T_i$ to
     centroids $L_1 \ldots L_k$ in order to determine
      the closest cluster to $T_i$ and update the
       cluster assignments $\mathcal{C}_1 \ldots \mathcal{C}_k$;
    Denote assigned cluster index for
      document $T_i$ by $q_c(i, t)$;
    Update cluster centroids $L_1 \ldots L_k$ to the
     centroids of updated clusters $\mathcal{C}_1 \ldots \mathcal{C}_k$;
    { Second Minor Iteration }
    Compute gini-index of $\mathcal{G}_r$ for each auxiliary
       attribute $r$ with respect to current
        clusters $\mathcal{C}_1 \ldots \mathcal{C}_k$;
    Mark attributes with gini-index which is
      $\gamma$ standard-deviations below the
      mean as non-discriminatory;
    { for document $T_i$ let $R_i$ be the set of
    attributes which take on the value of 1, and for
    which gini-index is discriminatory;}
    for each document $T_i$ use the method discussed
    in section 2 to determine the posterior
    probability $P^n(T_i \in \mathcal{C}_j | R_i)$;
    Denote $q_a(i, t)$ as the cluster-index with highest
    posterior probability of assignment for document $T_i$;
    Update cluster-centroids $L_1 \ldots L_k$ with the
     use of posterior probabilities as discussed in
       section 2;
    $t = t + 1$;
  end
end
```

Fig. 1. COATES algorithm

COLT Clustering Algorithm

Refer to our algorithm as the *COLT* algorithm throughout the paper, which refers to the fact that it is Content *and auxiliary attribute-based Text classification algorithm*.
The algorithm uses a supervised clustering approach in order to partition the data into *k* different clusters. This partitioning is then used for the purposes of **Clustering**. The steps used in the training algorithm are as follows:

Feature Selection: In the first step, use feature selection to remove those attributes, which are not related to the class label. This is performed both for the text attributes and the auxiliary attributes.

Initialization: In this step, use a *supervised k* means approach in order to perform the initialization, with the use of purely text content. The main difference between a supervised *k*-means initialization and an unsupervised initialization is that the class memberships of the records in each cluster are pure for the case of supervised initialization. Thus, the *k*-means clustering algorithm is modified, so that each cluster only contains records of a particular class.

Cluster-Training Model Construction: In this phase, a combination of the text and side-information is used for the purposes of creating a cluster-based model. As in the case of initialization, the purity of the clusters in maintained during this phase. Once the set of supervised clusters are constructed, these are used for the purposes of classification. will discuss each of these steps in some detail below.
Next, will describe the training process for the *COLT* algorithm. The first step in the training process is to create a set of supervised clusters, which are then leveraged for the cluster.
The first step in the supervised clustering process is to perform the feature selection, in which only the discriminative attributes are retained. In this feature selection process, compute the gini-index for each attribute in the data with respect to the class label. If the gini index is *γ* standard deviations (or more) below the average gini index of all attributes, then these attributes are pruned globally, and are never used further in the clustering process. Once the features have been selected, the initialization of the training procedure is performed only with the content attributes. Once the initialization has been performed, the main process of creating supervised clusters with the use of a combination of content and auxiliary attributes is started. As in the previous case, except that this is done only for luster indices which belong to the same class label. The document is assigned to one of the cluster indices with the largest

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. 3, Issue 2, February 2015

posterior probability. Thus, the assignment is always performed to a cluster with the same label, and each cluster maintains homogeneity of class distribution.

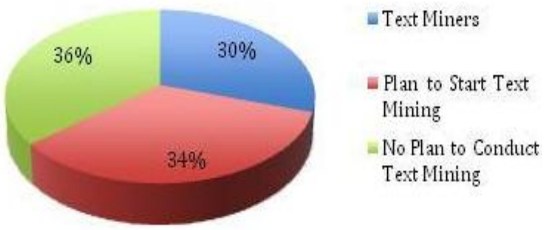
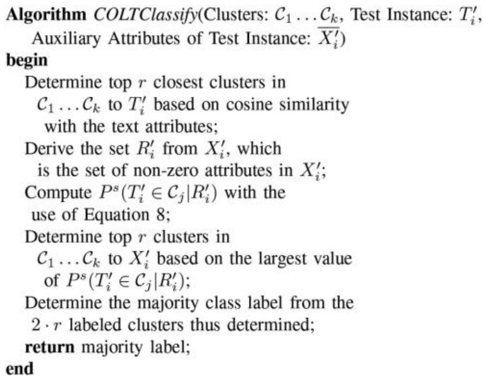


Fig. 2. TEXT training process

The *first two iterations* of the *k*-means type algorithm are run in exactly the same way as in , where the clusters are allod to have different class labels, and then adding supervision to the process. In order to achieve this goal, the *first two iterations* of the *k*-means type algorithm are run in exactly the same way as in , where the clusters are allod to have different class labels.



```
Algorithm COLTClassify(Clusters: $\mathcal{C}_1 \ldots \mathcal{C}_k$, Test Instance: $T_i'$,
    Auxiliary Attributes of Test Instance: $\overline{X_i'}$)
begin
  Determine top $r$ closest clusters in
    $\mathcal{C}_1 \ldots \mathcal{C}_k$ 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 $P^s(T_i' \in \mathcal{C}_j | R_i')$ with the
    use of Equation 8;
  Determine top $r$ clusters in
    $\mathcal{C}_1 \ldots \mathcal{C}_k$ to $X_i'$ based on the largest value
    of $P^s(T_i' \in \mathcal{C}_j | R_i')$;
  Determine the majority class label from the
    $2 \cdot r$ labeled clusters thus determined;
  return majority label;
end
```

Fig. 3. COLT process with the use of the supervised clusters.

V. EXPERIMENTAL SETUP

Refer to our clustering approach as *Content and Auxiliary attribute based Text clustering (COATES)*. As the baseline, used two different methods:

(1) 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. Refer to these algorithms as *Schutze Silverstein [text only]* in all figure legends in the experimental section.

(2) adapt the *k*-means approach with the use of both text and side information directly.

The *COLT* methods against the following baseline methods:

(1) Tested against a *Naive Bayes Classifier* which uses only text.

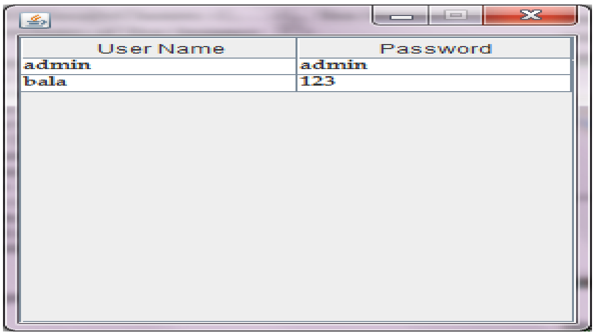
(2) Tested against an *SVM classifier* which uses only text.

(3) Tested against a supervised clustering method which uses both text and side information.

Will show that our approach has significant advantages for both the clustering and classification problems.

Effectiveness Results

The effectiveness results for the two baseline algorithms and *COATES* algorithms with increasing number of clusters for the *CORA*, *DBLP* and *IMDB* data sets are illustrated the *COATES* algorithm and the baselines. Notice that the purity will slightly increase when the number of clusters increases on all three data sets. This is quite natural, since larger number of clusters results in a finer granularity of partitioning. Further notice that *COATES* outperforms the baselines on all three data sets by a wide margin in terms of purity. On clustering, and therefore the classification accuracy is a horizontal line. In each case, it can be clearly seen



User Name	Password
admin	admin
bala	123

Fig 4. Login Data Gathering

that the accuracy of the COLT Cluster method was significantly higher than all the other methods. There re some variations in the classification accuracy across the different methods for different data sets. Hover, the COLT Cluster method retained the largest accuracy over all data sets, and was quite robust to the number of clusters.

For the entire range of values for the smoothing parameter _, the *COLT Cluster* method performs much more effectively with respect to the other schemes the only case where it does not do as ll is the case where the feature selection threshold is chosen to be too small.

VI. CONCLUSION

In this paper, presented methods for mining text data with the use of side-information. Many forms of text data gathered from databases 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, 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. 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. H. Frank, “Shortest paths in probabilistic graphs,” *Operations Research*, vol. 17, no. 4, pp. 583–599, 1969.
2. L. G. Valiant, “The complexity of enumeration and reliability problems,” *SIAM J. Comput.*, vol. 8, no. 3, pp. 410–421, 1979.
3. N. J. Krogan, G. Cagney, and al., “Global landscape of protein complexes in the yeast saccharomyces cerevisiae,” *Nature*, vol. 440, no. 7084, pp. 637–643, March 2006.
4. O. Benjelloun, A. D. Sarma, A. Halevy, and J. Widom, “Uldbs: Databases with uncertainty and lineage,” in *VLDB*, 2006, pp. 953–964.
5. N. N. Dalvi and D. Suciu, “Efficient query evaluation on probabilisticdatabases,” in *VLDB*, 2004. M. Potamias, F. Bonchi, A. Gionis, and G. Kollios, “k-nearest neighbors in uncertain graphs,” *PVLDB*, vol. 3, no. 1, pp. 997–1008, 2010.
6. Z. Zou, H. Gao, and J. Li, “Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics,” in *KDD*, 2010, pp. 633–642.
7. R. Shamir, R. Sharan, and D. Tsur, “Cluster graph modification problems,” *Discrete Applied Mathematics*, vol. 144, no. 1-2, pp. 173–182, 2004.
8. N. Bansal, A. Blum, and S. Chawla, “Correlation clustering,”*Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004.
9. U. Brandes, M. Gaertler, and D. Wagner, “Engineering graphclustering: Models and experimental evaluation,” *ACM Journal of Experimental Algorithmics*, vol. 12, 2007.
10. G. Karypis and V. Kumar, “Parallel multilevel k-way partitioning for irregular graphs,” *SIAM Review*, pp. 278–300, 1999.