www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3, Issue 11 November, 2014 Page No. 9016-9018

On the Use of Side Information for Text Mining using Clustering and Classification Techniques-A Survey

Subamanikandan A[1], Arulmurugan R[2]

[1]PG-Scholar, Bannari Amman Institute of Technology, Anna University,
Sathyamangalam, Erode, India
subamanikandan@hotmail.com

[2]Assistant Professor, Bannari Amman Institute of Technology, Anna University,
Sathyamangalam, Erode, India
arulmr@gmail.com

Abstract: ***Text mining application, side information is available along with text documents. Such side information may be contain different kinds, such as links in the document, document provenance information, user-access behavior from web logs or other non-textual attributes. Such attributes may contain large amount of information in the clustering purposes. However, the relative information is difficult to estimate, when some of information is noisy data. 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. In this paper, we design an algorithm which combines classical partitioning algorithms with probabilistic models in order to create an effective clustering approach. We then show how to extend the approach to the classification problem.***

Keywords: Text Mining, Classification, Clustering, Side Information

1. Introduction

Text Mining [1] is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted side information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation. Text mining is different from what are familiar with in web search. In search, the user is typically looking for something that is already known and has been written by someone else. The problem is pushing aside all the material that currently is not relevant to your needs in order to find the relevant information. In text mining, the goal is to discover unknown information, something that no one yet knows and so could not have yet written down.

Text mining is a variation on a field called data mining [2] that tries to find interesting patterns from large databases. Text mining, also known as Intelligent Text Analysis, Text Data Mining or Knowledge-Discovery in Text (KDT), refers generally to the process of extracting interesting and non-trivial information and knowledge from unstructured text. Text mining is a young interdisciplinary field which draws on information retrieval, data mining, machine learning, statistics and computational linguistics. As most information (over 80%) is stored as text, text mining is believed to have a high commercial potential value. Knowledge may be discovered from many sources of information, yet, unstructured texts remain the largest readily available source of knowledge.

Text mining [1] is similar to data mining, except that data mining tools are designed to handle structured data from databases, but text mining can work with unstructured or semi-structured data sets such as emails, full-text documents and HTML files etc. As a result, text mining is a much better solution for companies. To date, however, most research and development efforts have centered on data mining efforts using structured data. The problem introduced by text mining is obvious: natural language was developed for humans to communicate with one another and to record information, and computers are a long way from comprehending natural language. Figure 1 on next page, depicts a generic process model for a text mining application.

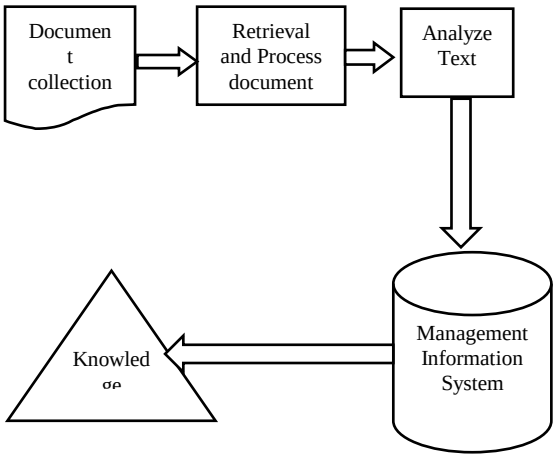

Figure 1: An Example of Text Mining

Starting with a collection of documents, a text mining tool would retrieve a particular document and preprocess it by checking format and character sets. Then it would go through a text analysis phase, sometimes repeating techniques until

information is extracted. Three text analysis techniques are shown in the example, but many other combinations of techniques could be used depending on the goals of the organization. The resulting information can be placed in a management information system, yielding an abundant amount of knowledge for the user of that system.

2. Side Information

The problem of text clustering arises in the context of many application domains such as the web, social networks, and other digital collections. A tremendous amount of work has been done in recent years on the problem of clustering in text collections [4], [5], [6], [7] in the database and information retrieval communities. However, this work is primarily designed for the problem of pure text clustering, in the absence of other kinds of attributes. 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 [9] which may be useful to the clustering process. Some examples of such side-information are as follows

2.1 User Access Web Documents

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.

2.2 Text Document Contains Links

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.

2.3 Meta-data

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. In a number of network and user-sharing applications, documents may be associated with user-tags, which may also be quite informative.

3. Clustering Algorithm

3.1 Hierarchical clustering

Hierarchical clustering [10] builds a cluster hierarchy or, in other words, a tree of clusters, also known as a dendogram. Every cluster node contains child clusters; sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are categorized into agglomerative (bottom - up) and divisive (top - down).An agglomerative clustering starts with one point clusters and recursively merges two or more most appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until a stopping criterion (frequently, the requested number k of clusters) is achieved. Advantages are 1) Embedded flexibility regarding the level of granularity 2) Ease of handling of any forms of similarity or distance 3) Consequently, applicability to any attribute types. Disadvantages are 1) Vagueness of termination criteria 2) The fact that most hierarchical algorithms do not revisit once constructed (intermediate) clusters with the purpose of their improvement.

3.2 K- medoid clustering algorithm

In k-medoid clustering algorithms, [11] we use a set of points from the original data as the anchors (or medoids) around which the clusters are built. The key aim of the algorithm is to determine an optimal set of representative documents from the original corpus around which the clusters are built. Each document is assigned to its closest representative from the collection. This creates a running set of clusters from the corpus which are successively improved by a randomized process. The algorithm works with an iterative approach in which the set of k representatives are successively improved with the use of randomized inter-changes. Specifically, we use the average similarity of each document in the corpus to its closest representative as the objective function which needs to be improved during this interchange process. In each iteration, we replace a randomly picked representative in the current set of medoids with a randomly picked representative from the collection, if it improves the clustering objective function. This approach is applied until convergence is achieved.

3.3 Online spherical K-means algorithm

The spherical k-means algorithm, [12] i.e., the k-means algorithm with cosine similarity, is a popular method for clustering high-dimensional text data. In this algorithm, each document as well as each cluster mean is represented as a high-dimensional unit-length vector. This algorithm investigates an online version of the spherical k-means algorithm based on the well-known Winner-Take-All competitive learning. In this online algorithm, each cluster centroid is incrementally updated given a document.

Document clustering has become an increasingly important technique for unsupervised document organization, automatic topic extraction, and fast information retrieval or filtering. For example, a web search engine often returns thousands of pages in response to a broad query, making it difficult for users to browse or to identify relevant information. Clustering methods can be used to automatically group the retrieved documents into a list of meaningful categories. Similarly, a large database of documents can be pre-clustered to facilitate query processing by searching only the cluster that is closest to the query.

4. CLASSIFICATION ALGORITHM

4.1 Decision Trees

Decision trees are designed with the use of a hierarchical division of the underlying data space with the use of different text features [13]. The hierarchical division of the data space is designed in order to create class partitions which are more skewed in terms of their class distribution. For a given text instance, we determine the partition that it is most likely to

belong to, and use it for the purposes of classification.

4.2 Pattern (Rule)-based Classifiers

In rule-based classifiers we determine the word patterns which are most likely to be related to the different classes. We construct a set of rules, in which the left- hand side corresponds to a word pattern, and the right-hand side corresponds to a class label. These rules are used for the purposes of classification.

4.3 SVM Classifiers

SVM Classifiers [14] attempt to partition the data space with the use of linear or non-linear delineations between the different classes. The key in such classifiers is to determine the optimal boundaries between the different classes and use them for the purposes of classification.

4.4 Neural Network Classifier

Neural networks are used in a wide variety of domains for the purposes of classification. In the context of text data, the main difference for neural network classifiers is to adapt these classifiers with the use of word features. We note that neural network classifiers are related to SVM classifiers; indeed, they both are in the category of discriminative classifiers, which are in contrast with the generative classifiers [15].

4.5 Bayesian Classifiers

In Bayesian classifiers (also called generative classifiers), we attempt to build a probabilistic classifier based on modeling the underlying word features in different classes. The idea is then to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word presence in the document.

4.6 Naive Bayes Classifier

A Bayes classifier [16] is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

5. Conclusion

In this paper, 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. 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.

References

[1] Vishal Gupta, Gurpreet S. Lehal, "A Survey of Text Mining Techniques and Applications," in Journal of Emerging Technologies in Web Intelligence, Vol. 1, No. 1, Augest 2009, pp. 60-76.

[2] Navathe, shamkant B., and Elmasri Ramez, (2000), "Data Warehousing and Data Mining", in "Fundamentals of Database Systems", Pearson Education pvt Inc, Singapore, 841-872.

[3] Berry Micheal W., (2004), "Automatic Discovery of Similar Words", in "Survey of Text Mining: Clustering, Classification and Reterieval", Springer Verlag, New York,LLC, 24-43.

[4] 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.

[5] D.Cutting, D.Karger, J.Pedersen, and J.Tukey, "Scatter/Gather:A Cluster-based to browsing large document collections," in Proc. ACMSIGIR Conf., New York, NY, USA, 1992, pp. 318-329.

[6] H. Schutze and C.Silverstein, "Projections for efficient document clustering," in Proc. ACMSIGIR Conf., New York, USA, 1997, pp. 74-81.

[7] M. Steinbach, G.Karypis, and V.Kumar, "A Comparison of document clustering techniques," in Proc. Text Mining Workshop KDD, 2000, pp. 109-110.

[8] S. Zhong, "Efficient streaming text clustering." Neural Netw, vol. 18, no. 5-6, pp. 790-798, 2005.

[9] C.C.Aggarwal, Yuchen Zhao, and P.S.YU, "On the Use of

[10] Side Information for Mining Text Data", in Knowledge and Data Engineering, vol.26, no.6, 2014, pp. 1415-1419.

[11] A. Jain and R. Dubes, "Algorithm for Clustering Data" Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1988.

[12] Charu C. Aggarwal, ChengXiang Zhai "A Survey of Text Clustering Algorithms", pp. 92-94.

[13] Mrs. Sayantani Ghosh, Mr. Sudipta Roy, and Prof. Samir K. Bandyopadhyay, "A tutorial review on Text Mining Algorithms" in International Journal of Advanced Reserch in Computer and Communication Engineering, Vol. 1, Issue 4, June 2012, pp. 223-233.

[14] Shi Zhong, "Efficient Online Sphercal K-means Clustering", in Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, jii1 - August 4, 2005, pp. 3810-3815

[15] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, Khairullah khan, "A Review of Machine Learning Algorithm for Text Classification " in Journal of Advances in Information Technology, Vol. 1, no. 1, February 2010

[16] C.C.Aggarwal, Chengxiang Zhai, "Mining Text Data" in Kluwer Academic Publishers.

[17] Http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Naive_Bayes_classifier.html.