ISSN:0975-9646

Supervised and Unsupervised Document Classification-A survey

Deepshikha Kalita

Department of IT, Gauhati University
Jalukbari, Guwahati, Assam

***Abstract*- All users want to have their documents in a more systematic and secured way. Assume a situation. We have huge collections of books. It may contain novels, storybooks, and fictions, books on Culture and Heritage, History, and Geography etc. Suppose someone enquires of a book on History. It is quite difficult for us to find it in the midst of all books. If we manually go for searching, it may take several hours, may be days also. If we can categorize the books in different categories with respect to some criteria, it would have been more efficient to search and more secured too. A major problem faced by institutions, organizations, and businesses nowadays is that of information overload. Sorting out useful documents from collection that are not of interest challenges the ingenuity and resources of both individuals and organizations. Keyword search engines can be helpful but there are some limitations in this. Keyword searches don't discriminate by context. On the other hand if we manually go for classifications and clustering, it is not feasible for large volumes of documents. So we need to develop an automatic classifier to manage the documents in a more secure way. By classifying a document we can establish the required level of protection with less manual effort. Documents' classification and clustering are two very important techniques for achieving this goal. In this survey report we will discuss various methods of documents' classification and its various approaches used till date. We will also present a review of comparisons of the existing methods along with their advantages and disadvantages.**

***Keywords*-Documentation, supervised, unsupervised, clustering, class, feature selection, training data, test data.**

1. INTRODUCTION:

1.1 DOCUMENT CLASSIFICATION: Document classification aims to classify textual documents automatically, based on the words, phrases and word combinations. Document classification or document categorization is a problem in library science, information science and computer science. The task is to assign a document to one or more classes or categories. This can be done manually (or intellectually) or algorithmically, the intellectual classification of document has mostly been the province of library science while the algorithmic classification of document is used mainly in information science and computer science. The documents to be classified may be text, images; music etc., if not stated otherwise text classification is implied. Each kind of document possesses special kind of problems. According to subjects or other attributes documents can also be classified. There are the basic methods to classify the documents:

a) Rule based document classification
b) Supervised document classification
c) Unsupervised document classification

Rule based classification: Here we group the documents together, decide on categories and formulate the rules that define those categories; these rules are actually query phrases. It is accurate for small document sets. Here rules are written by the writers themselves. This approach is very accurate for small document set. Results are always based on what writers define, since it is fully dependent on what writers write but defining rules can be tedious for large document sets with many categories. As the document set grows writers may need to write correspondingly more rules.

Supervised classification: Documents are classified on the basis of supervised learning. In supervised learning external knowledge or information is provided. Here each example is a pair consisting of an input object and a desired output value. A supervised algorithm analyses the training data and produces an inferred function which can be used for mapping other examples. The data (observations, measurements, etc.) are labeled with pre-defined classes. It is like that a "teacher" gives the classes (supervision).Like human learning from past experiences, a computer does not have "experiences". A computer system learns from data, which represent some "past experiences" of an application domain. Our focus is to learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk. The task is commonly called supervised learning, classification, or inductive learning.

Supervised learning process in two steps:

- **Learning (training)**: Learn a model using the training data.
- **Testing:** Test the model using unseen test data to access the model accuracy.

Accuracy=no of training data/no. of test cases

Unsupervised classification:

Entirely without reference to external information[4]. It can be achieved through clustering. A way of grouping together data samples that is similar in some way according to some criteria we just pick out. So, it's a method of data exploration, a way of looking for patterns or structure in the data that are of interest. It involves the use of descriptors and descriptors extraction. Descriptors are set of words that describe the contents within the cluster. It is considered to be a centralized process. E.g.: Web document clustering for

search users. It requires no predefined classes or category. In unsupervised clustering, we have unlabelled collection of documents. The aim is to cluster the documents without additional knowledge or intervention such that Documents within a cluster are more similar than documents between clusters. Traditional Clustering techniques can be categorized into two major groups as partitional and Hierarchical. Here we discuss these groups and their main representatives.

2. SUPERVISED DOCUMENT CLASSIFICATION:

There are different types of supervised documents' classifications.

- Content based classifications
- Request based classifications

Content based classifications: It is a classification in which a weight given to a particular subject in a document determines the class to which the document is assigned. Eg: a rule in library classification that at least 20% of the content of a book should be about the class to which the book is assigned[1].In automatic document classification it could be no. of times given words appear in a document[2].
Automatic Document Classification is also a content based classification. It is generally defined as content based assignment of one or more predefined categories to documents. Usually, machine learning, statistical pattern recognition or neural network approaches are used to construct classifiers automatically.
Request based classifications: It is a classification in which anticipated request from users is influencing how document are being classified. The classifier asks itself: "under which descriptors should this entity be found and think of all possible queries and decide for entities which are relevant."

DOCUMENT PREPROCESSING: Before using the classifier all documents have to be processed through the some steps irrespective of the classifier. We can see several attributes or words which are of no use (such as the word „a", „the", etc.) or does not have any strong meaning in the entire document. These are known as stopwords. Thus, stop word removing algorithm has been applied. To initialize the algorithm, a set of stop word (such as a, a's, able, about, above, according, accordingly, and across) has set by the human beforehand and hence stored in a text file. Then, the model can simply match the attributes with those preset stopword. .
Another algorithm applied in the preprocessing phase is the stemming. Since some words carry similar meanings but in different grammatically form (such as "school" and "schools"), so we deal only with the root words (the base word). Here we will stem every word to its origin form.
FEATURE SELECTION: Feature selection is one of the most important preprocessing steps in data mining. It is an effective dimensionality. There are many techniques to select features from a set of documents such as mutual information, chi square test, maximum entropy etc.. The whole classification of documents depends on these features.

2.1 SOME TECHNIQUES OR APPROACHESUSED IN SUPERVISED CLASSIFICATION:

2.1.1 CENTROID BASED APPROACH [5] :

In this technique, each document D is represented by vector called Document Vector V(D) in the feature space and each component of the vector is represented by tf*idf value i.e.V(D)=(tf*idf1, tf*idf2,.....tf*idfn) where n is the nth word in the document. Also, centroid vector of each class is computed and then the Euclidean distance between document vector and centroid vector of each class is calculated. Then the document that is having minimum distance from centroid vector of a particular class is assigned to that class. The computational complexity of the learning phase of this centroid-based classifier is linear on the number of documents and the number of terms in the training set. The computation of the vector-space representation of the documents can be easily computed by performing at most three passes through the training set. Similarly, all k centroids can be computed in a single pass through the training set, as each centroid is computed by averaging the documents of the corresponding class.

2.2.2 STATISTICAL BASED APPROACH: 2.22a) SUPPORT VECTOR MACHINE:

This is also a supervised learning model with associated learning algorithms that analyze data and recognize patterns, used for classification analysis. It is a representation of the examples as points in space, mapped so that the example of the separate categories is divided by a clear gap that is as wide as possible. Support Vector Machines (SVM) is a technique introduced by Vapnik in 1995, which is based on the Structural Risk Minimization principle [11]. It is designed for solving two-class pattern recognition problems. The problem is to find the decision surface that separates the positive and negative training examples of a category with maximum margin. Support vector machines find the hyper plane 'h' that separates positive and negative training examples with maximum margin. Support vectors are marked with circles. A decision surface in a linearly separable space is a hyper plane. **Margin** is the distance between the parallel lines.

2.2.3 PROBABILITY BASED APPROCH: 2.2.3a) NAIVE BAYES CLASSIFIER:

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions.This model is a "independent feature model". The Bayesian classification represents a supervised learning method as well as a statistical method for classification. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. Here in this classifier firstly data preprocessing steps are taken. Some data are useless (i.e. do not affect the classification result even removing them such as stop words) and some carries similar meanings (i.e. eat and eats), therefore these words are removed. In this way, the dataset can be more precise. After the data preprocessing phase, critical or the main attribute that represents the document have to be selected. For example, the word "banker" categorized in "business" class has the highest score in term of term frequency, therefore it is analyzed that "banker" is one of the critical

attributes to represent the documents fell in the "business" class. Thus, less important features can be removed and so the computational time can be improved significantly.
The probabilistic characteristic of Naïve Bayes is each document is vectorized by the trained Naïve Bayes classifier through the calculation of the posterior probability value for each existing. Finally, the model is evaluated by a set of testing data. In order to test the classification ability of the model, several evaluation measures (such as precision, recall, and F-measure) are adopted. Furthermore, to interpret whether Naïve Bayes is best to use as the classifier, its testing result will be compared with other classifiers results as well.

2.2.3b) DECISION TREES:
This form of classification uses a decision tree algorithm for creating rules. A decision tree is a method of deciding between two choices. In document classification the choices are the "document matches the training set" or "the document doesn't matches the training set." Here a set of attributes are tested such as words from the document, stems of words from the documents, themes from the document (if supported for the language in use.).
A sample decision tree model for a particular retail shop indicating whether a customer is likely to purchase a computer is shown in fig:3 .Each internal node (non leaf node) represents test on an attribute. Every leaf node represents a class. Here, the classes are 'yes' or 'no'.

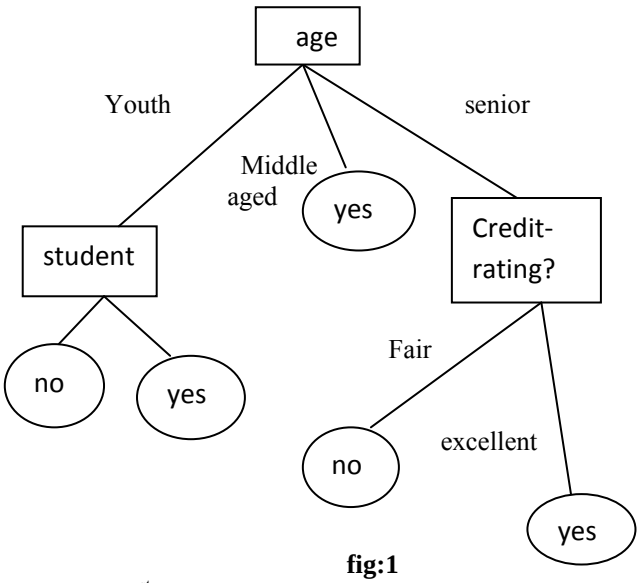

fig:1

One of the 1st decision tree algorithm used is known as ID3 (iterative Dichotomizer).

3. SOME TECHNIQUES OR APPROACHES USED IN UNSUPERVISED CLASSIFICATION TILL NOW:

In unsupervised clustering, we have unlabelled collection of documents. The aim is to cluster the documents without additional knowledge or intervention such that documents within a cluster are more similar than documents between clusters. Traditional clustering techniques can be categorized into two major groups as partitional and hierarchical. Here we will discuss these groups and their main representatives.

3.1 Partitional Clustering Techniques:
Partitional algorithms produce un-nested, non-overlapping partitions of documents that usually locally optimize a clustering criterion. The general methodology is as follows: given the number of clusters *k*, an initial partition is constructed; next the clustering solution is refined iteratively by moving documents from one cluster to another. In the following sub-sections we discuss the most popular partitional algorithm *k*-means, and its variant bisecting *k*-means which has been applied to cluster documents by Steinbach *et al.* in [18] and has been shown to generally outperform agglomerative hierarchical algorithms.
3.2 K-Means Clustering: The idea behind the *k*-means algorithm is that each of *k* clusters can be represented by the mean of the documents assigned to that cluster, which is called the centroid of that cluster. There are two versions of *k*-means algorithm known. The first version is the batch version and is also known as Forgy's algorithm.It consists of the following two-step major iterations:
(i) Reassign all the documents to their nearest centroids
(ii) Recompute centroids of newly assembled groups
Before the iterations start, firstly *k* documents are selected as the initial centroids. Iterations continue until a stopping criterion such as no reassignments occur is achieved. Initially, *k* documents from the corpus are selected randomly as the initial centroids. Then, iteratively documents are assigned to their nearest centroid and centroids are updated incrementally, i.e., after each assignment of a document to its nearest centroid. Iterations stop, when no reassignments of documents occur.
3.3 Bisecting K-Means: Although bisecting *k*-means is actually a divisive clustering algorithm that achieves hierarchy of clusters by repeatedly applying the basic *k*-means algorithm, we discuss it in this section as it is a variant of *k*-means. In each step of bisecting *k*-means a cluster is selected to be split and it is split into two by applying basic *k*-means for $k = 2$. The largest cluster, that is the cluster containing the maximum number of documents, or the cluster with the least overall similarity can be chosen to be split.
3.4 Hierarchical Clustering Techniques: Hierarchical clustering algorithms produce a cluster hierarchy named a dendrogram. These algorithms can be categorized as divisive (top-down) and agglomerative (bottom-up). [10]
3.4.1 Divisive Hierarchical Clustering:
Divisive algorithms start with one cluster of all documents and at each iteration split the most appropriate cluster until a stopping criterion such as a requested number *k* of clusters is achieved. A method to implement a divisive hierarchical algorithm is described by Kaufman and Rousseeuw. In this technique in each step the cluster with the largest diameter is split, i.e. the cluster containing the most distant pair of documents. As we use document similarity instead of distance as a proximity measure, the cluster to be split is the one containing the least similar pair of documents. Within this cluster the document with the least average similarity to the other documents is removed to form a new singleton cluster. The algorithm proceeds by iteratively assigning the documents in the cluster being split

to the new cluster if they have greater average similarity to the documents in the new cluster. To our knowledge, divisive hierarchical clustering in this sense has not been applied to document corpora. This method is not robust to outliers and in our experiments we observe that documents in the cluster being split generally tend to remain in the larger old cluster and for small number of clusters *k*, clustering quality is not comparable with the other algorithms we evaluated. So, we made a slight modification to this algorithm. In our version we select the least similar pair of documents in the cluster being split and remove them to form two new singleton clusters. The rest of the documents in the cluster are assigned iteratively to one of the new clusters by taking the average similarity as criterion.

3.4.2 Agglomerative Hierarchical Clustering: Agglomerative clustering algorithms start with each document in a separate cluster and at each iteration merge the most similar clusters until the stopping criterion is met. They are mainly categorized as single-link, complete-link and average-link depending on the method they define inter-cluster similarity.[16] .

Single-link: The single-link method defines the similarity of two clusters Ci and Cj as the similarity of the two most similar documents.

Complete-link: The complete-link method defines the similarity of two clusters Ci and Cj as the similarity of the two least similar documents.

Average-link: The average-link method defines the similarity of two clusters *Ci* and *Cj* as the average of the pair wise similarities of the documents from each cluster.

CONCLUSION AND FUTURE WORK:

We have discussed here various approaches of text classification and there methodology. Based on this approaches we will try to build a classifier and compare with other existing classifiers to see the accuracy of results. No Assamese Text Classifier is available in the world to classify the Assamese Text Documents. On the basis of this literature survey we will try to design a classifier in Assamese language*, applying any one of the approaches mentioned here.

REFERENCES:

*****Assamese Language is one of the Modern Indian Languages used in the state of Assam which is located in the North-Eastern part of India.

[1] Library of congress (2008) Washington DC: Library of congress, Policy and Stds Division.

[2] Soergel, Dagobert. "Organizing Information: Principles of Database and Retrieval Systems.", Orlando, FL: Academic Press., 1985.

[3] Lanchester, F.W.: "Indexing and Abstracting in Theory and Practice", Library Association, London,2003

[4] Shailesh S.Deshpande and Girish Keshav Palshikar and G. Athiappan **"**An unsupervised approach to sentence classification"

[5] Eui-Hong (Sam) Han and George Karypis University of Minnesota, Department of Computer Science / Army HPC Research Center Minneapolis, MN 55455 "Centroid-Based Document Classification: Analysis & Experimental Results"

[6] David Martens, Foster Provost Working paper CeDER-11-01 "Explaining Documents' Classifications"

[7] Charu C. Aggarwal, ChengXiang Zhai "A Survey Of Text Clustering Algorithms"International Journal on Computational Sciences & Applications (IJCSA) Vol.3, No.6, December 2013 DOI:10.5121/ijcsa.2013.3604 31

[8] R.Jensi , Research Scholar, Dr.G.Wiselin Jiji, Dr.Sivanthi "A Survey On Optimization Approaches To Text Document Clustering"

[9] G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Informationby Computer. Addison-Wesley, 1989

[10] Data Mining- AK Pujari

[11] Burges, C. J. C., "A Tutorial on Support Vector Machines for Pattern Recognition",Data Mining and Knowledge Discovery, Vol. 2, No. 2, pp. 121–167, 1998.

[12] Joachims, T., "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", European Conference on Machine Learning (ECML),1998

[13] Yang, Y. and X. Liu, "A Re-examination of Text Categorization Methods", Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval , Berkeley, US, 1999.

[14] Joachims, T., "Making Large-Scale SVM Learning Practical", Report LS-8, University¨at Dortmund,1998

[15] Joachims, T., Advances in Kernel Methods-Support Vector Learning, chapter Making Large-Scale SVM Learning Practical, MIT-Press, 1999.

[16] Arzucan ¨Ozg¨ur B.S. in Computer Engineering, Bo˘gazi, ci University, 2002

[17] Steinbach, M., G. Karypis, and V. Kumar, "A Comparison of Document ClusteringTechniques", KDD Workshop on Text Mining, 1999.