ISSN:2249-5789

Survey on Structure and Structure-content Classification of XML Document

Thasleena N T
Department of Computer Science
Rajagiri School of Engineering
& Technology, Kochi
thalu555@gmail.com

Varghese S C
Department of Computer Science
Rajagiri School of Engineering
& Technology, Kochi
varghesesc@rajagiritech.ac.in

***Abstract*—In recent years, XML has become a popular way of storing many data sets because of its semi-structured nature. It allows modeling of a wide variety of databases as XML documents. XML data thus form a significant part in data mining domain, and it is valuable to develop classification methods for such data. Due to increase in XML documents, researchers are now focusing on applying the typical text mining tasks such as text classification, text clustering and other related tasks on XML corpus. In this work, our objective is to give a survey on the classification of XML documents. As XML documents are basically text documents containing the content and structure information, they can be classified based on i) structure only and ii) a combination of both structure and content. This paper gives a brief survey based on this classification**

Keywords:XML,classification,clustering,ontology,feature extraction,data mining,frequent pattern,XSLT,WordNet.

I. Introduction

XML is one of the popular structure for data representation that allows organizing textual content into logical structures. In case of traditional information retrieval systems that deal with only flat documents but XML retrieval systems must also take the structure of documents along with its textual contents. Every XML document includes both logical and physical structures. So based on these two information XML can be classified based on two approaches. One approach uses only the structural information of XML data in classification. Another one performs the classification, by considering both the content and structural information of XML data.

The classification problem is defined as follows. We have an input data set called the training data which consist of a set of features (attribute of data) along with a special variable called the class. This class variable takes its value from a different set of classes. The training data is used to construct a model which relates the feature variables in the training data to the class variable. The training model is used in order to predict the class variable for such test instances. The test instances of the classification problem consist of a set of records for which only the feature values are known while the class value is unknown. The training model is used in order to predict the class variable for such test instances. In the training phase, sets of XML documents belonging to each class are used to create representations of the classes. In the classification phase, new documents are compared with these representations in order to assign them to one or more classes. However for XML classification which involves the documents structure, the task of finding representations is highly difficult and expensive

In recent years, XML has become a popular way of storing many data sets because of its semi-structured nature. It allows modeling of a wide variety of databases as XML documents. XML data thus form a significant part in data mining domain, and it is valuable to develop classification methods for such data. Due to increase in XML documents, researchers are now focusing on applying the typical text mining tasks such as text classification, text clustering, document summarization and other related tasks on XML corpus.In this work, our objective is to give a survey on the classification of XML documents. As XML documents are basically text document containing the content and structure information, they can be classified based on i) content only ii) structure only and iii) a combination of both structure and content. A natural tendency for content based XML document classification would be to use conventional text classification approaches where each XML document could be treated as Bag-Of-Words (BOW). This approach is not an efficient one as it totally ignores the structural component of XML documents; thereby defeating the whole purpose of the XML document itself. The second method which is based on structure generally model XML documents as labeled trees, where the interior nodes represent the XML tags and the leaf nodes represent the content. Besides structure, contents also have a major role to play in XML documents which is ignored in this approach. As the XML documents are made up of both structure and content it is quite natural to give equal importance of structure and content (method 3) rather than considering only either of them. Usually, classification is preceded by a training step

Rest of this paper is organized as follows. Section I is a text document classification. Section III is a Classification techniques based on structure Section IV explains a classification method based on structure and contents Section V is conclusions and future work

II. Text Document Classification

In case of text document classification there are set of training document and each document is labeled with a class value taken from set of class values. Training data is used to construct a classification model which relates features in the document to one of the class label.

For a given test document with unknown class label, classification model is used to predict the class of test document. There are a wide variety of techniques have been designed for text document classification. Some method are shown below.

SVM [15] is a machine learning system with learning as an optimization problem. Here d-dimensional vectors are used to represent training and test document. Each training document belongs to either positive or negative classes. SVM tries to create a hyperplane or a set of hyperplanes of the form w.x-b=0 to separate positive instance from negative instance. Here, w is the normal of the hyperplane and the constant b defines the position of the plane in space. There are many hyperplane than can classify documents. So select the hyperplane that has largest separation between two classes. When a new test document arrives they are classified based on its distance from hyperplane.

KNN [18] classifier is a similarity based learning method. In this approach, given a test document it finds k-nearest neighbors among training documents. A similarity score between test document and each neighbor's document is taken as a weight of categories of neighbor's document. The calculate scores are sorted by values and then find top k-matches. The classes to which each of the top k-neighbors belong are already known. If some of the k-neighbors share a class then the their weight is added together to get the overall class of weight. Finally the scores are ranked to determine the classes to which document belongs.

In neural network [16] supervised learning and unsupervised learning are used for training. Supervised learning contains set of input features of each example and expected output for that example. It is called supervised learning because the weight of the network is adjusted unlit its obtained output closer to expected output in the training phase. Backpropagation is used for this approach. In unsupervised method there is only input features and network perform some technique to find classes of test document.

Decision trees [19] are most widely used inductive learning methods. It is a hierarchical representation of a training data. Here predicate or a condition on the attribute values are used to divide the data. In case of train data these are condition in the presence or absence of words in the document. Division on the data node done in a recursive manner until it reaches a leaf with minimum records. Given a test document applies the predicate on nodes to determine the path of relevant leaf.

Rulebased classifier [20] model data space as set of rules. Here left hand side of the rule represent condition on the feature set and the right hand side is the class label. Rule sets are generally constructed from training set based on support and confidence. Support is the total number of instances in training data which are relevant to the rule and confidence is the conditional probability of right hand side of the rule is satisfied if we are given left hand side of the rule are satisfied. Given a test instance, we have to find a set of rules in which test document satisfies condition on the left hand side of the rule.

Naivebased classifier [17] is simplest and most commonly used classifier. It represents a model to distribute documents into classes with independent assumption. So it assumes presence or absence of each feature is independent of other features. Two model (Multivariate Bernoulli model and multinomial model) are used to produce posterior probability of classes based on the distribution of terms in document. In first model the presence or absence of word in document the as a feature. But in latter case we capture the frequencies of terms in a document.

In case of XML document, text categorization is not beneficial because XML documents contain structure and text features. Text categorization ignores its structural feature. So many literatures proposed classifiers in XML document after extracting their structure and structure-content information.

III. Classification Based on Structure of XML Document

Different method for Structural classification of XML documents have been proposed in the following literature (e.g., [Garboni 2006; Knijf 2007; Kurt 2006; Zaki 2006]). Here classification relies on structural features only. In several cases, the classification behavior of the XML document is hidden within the structure info out there within the document. In such cases, the use of Informational Retrieval based classifiers is probably going to be ineffective for XML documents.

Paper [1] has focused on the use of rule based classifiers as an effective tool for data classification. Rule based classifiers are motivating technique that integrates the problem of associations and classification. XRULE [1] Discuss the problems of constructing structural rules in so as to perform the classification task. The training phase finds the structures that are most closely related to the class variable. After the completion of the training phase perform testing phase in which these rules are used to predict the class of unknown XML documents. Here XML documents are modeled as ordered, labeled, rooted trees. There is no distinction between attributes and elements of an XML document; both are mapped to the label set. The training phase uses a collection of structures with known classes to build a classification model. It initially enumerates all frequent embedded subtrees related to the individual classes of the XML documents. These subtrees are then exploited to form a set of predictive structural rules. The testing phase takes as input a database of structures with unknown classes, and the goal is to use the classification model to predict their classes

The limitation of this method is a huge number of rules are produced by rule generator algorithm, and it is very difficult to store the rules, retrieve the related rules, and set the rules. In most cases, XRULE achieves high-classification accuracy by using considerably large number of rules in the classifier, which successively might cause overfitting, particularly for small training datasets. Furthermore, XRULE completely ignores the content of XML documents.

In [4] each XML document is represented as a sequence of node labels corresponding to the structural feature of the document obtained through a depth-first traversal of its

tree structure. Here structural features are discovered by sequential pattern mining, For this purpose, use a method intended to remodel any XML tree into a sequence. First it extracts frequent tag entrenched in the document collection. The main idea is to remove irrelevant tags(ie. that is extremely frequent within the whole collection could also be thought of as irrelevant since it'll not facilitate in separating a document from another). Then perform a data mining step on each cluster from the training collection. For each cluster, the goal is to transform each XML document into a sequence during the mapping operation, after that frequent tags extracted from first step are removed. Then on each set of sequences corresponding to the original clusters, perform a data mining step meant to extract the sequential patterns. Finally the last step of method depends on a matching between each document of the collection and each cluster which is characterized by a group of frequent structural using the longest common subsequence. An unlabeled XML document is mapped into the class whose characteristic sequential patterns exhibit the highest average matching score

The limitation of this approach is the inability at discriminating the classes, when all of the available XML documents share an undifferentiated structure explore discriminative

Paper [5] describes a classification method for XML data based on frequent attribute trees. An attribute tree is essentially a subtree that also takes into account the attributes associated with the nodes of the tree structures of the original XML documents. From these frequent patterns we select so called emerging patterns, and use these as binary features in a decision tree algorithm. An attribute tree is said to be emerging if it frequently occurs in the XML trees of one class and rarely within those in any other class. Frequent tree mining is a data mining technique that is able to exploit the information on the structure of the data present in the XML-databases. Frequent tree mining is an instance of frequent pattern mining, specialized on tree structured data. The goal of frequent tree mining algorithms is to find all frequently occurring subtrees in a database. Classification of XML documents is as follows:

- Compute the frequent patterns for the different classes in the training set.
- Select from these frequent patterns the emerging patterns.
- The emerging patterns are used to learn the classification model on the training set.
- Evaluate the classification model on the test set.

Main criticism to the approaches is that the search of frequent subtrees within the individual classes is a very time-expensive step that may make model induction computationally infeasible when the number of underlying XML documents is extremely large

Paper [14] combines the advantages of XML and HTML for classification of document using XSLT transformation (XSLT classification). XSLT is as the language used for transforming XML document into HTML or other format. XSLT exploit XSL standard which contains some transformation rule for XML document transformation. XSLT Stylesheet and XML documents (data + content) are given to XSLT processors that are available in most languages as programming API. XSLT classification is a curious technique that extracts structural tag in XML and presentation tag in HTML. The framework for classification consists of three workings, such as Preprocessor, Semi-Structured Document Modeler, and Classifiers. In the preprocessing step, XSLT-to-XSLT stylesheet is given to the original XSLT stylesheet to create formatted-XSLT stylesheet (i.e. Transformation rule in original XSLT). After applying formatted XSTL into XML document create a formatted modeler produce feature vector for term frequency vector for classification purpose. The main advantage is utilization of both structured data in XML and relevant data in HTML using the transformation rules in XSLT stylesheets. But the approach has limited applicability in uncommon application settings.

IV. Classification of XML document Based on Contents and Structure

Different method for Structural classification of XML documents have been proposed in the literature (e.g., Theobald [2003]; Yi and Sundaresan [2000]; Costa [2013]; Wu and Tang [2008]; Mohammad Khabbaz [2012]). Classification method based on this approach captures content and structural features of XML documents. Previous approaches only ignores Content part of an XML document. Some relevant information may occur in content part of the document. So we have to consider them along with structural information for classification.

This paper [6] uses term path and twigs of XML documents as extended features in addition to text term frequency vector. This extended feature can be combined with text terms occurs in XML. And also it uses ontological information(WordNet thesaurus) for the purpose of more expressive feature space. Its features for classification are tag term pair(content feature),twig and tag path(structural features)and mapping word into word senses (ontological feature). In case of content features extraction, the first step is the extraction of characteristic word of the element content using stop word removal or noun recognition. Then combine this term with element name(tag) gives encoded single feature tag $ term . Only subsets of this feature is selected by Mutual Information [10] [11] for the input vector. In structural features, it uses tag path length 2 (i.e parent tag, tag pairs or parent tag, tag, term triples) and also use twigs that are specific way to split the XML document into small unit with respect to sibling elements. Twigs are encoded in the form of left child tag parent tag right child tag. In case of ontological feature, for each tag word set look up each of the words in WordNet, or a special database constructed from the WordNets contents and based on that identify possible word sense. The mapping of a tag onto a word sense is to compare the tag context (i.e.full text content (including the tag name) of the corresponding element and all its subordinate elements)with the context of candidates(i.e. Synonyms, all immediate hyponyms, holonyms, and Hypernyms), each of these has a sunset and also a short explanatory text. We form the union of the synsets and corresponding texts in terms of a similarity measure between bags of words. After the construction of

the feature space classification model is constructed using SVM.

Although showing accuracy improvement, this approach suffers from efficiency issues when tag-path features are constructed to consider any length within the XML document. This may generate a huge number of structural features, which can degrade the performance of the classifier algorithm.

In paper [13] to include structural information it represent document as structured vector model (ie. Its element can be either terms or structure). It is done by some observation that is terms from same element are treated together and also differentiate from other XML element. It also improves the well known probabilistic classifier method based on Bernoulli document generation model. The classification of XML documents is performed by means of a NaveBayes classifier obtained by an adaptation of the traditional Bernoulli generative model to the structured vector model

Paper [7] used to construct a set of informative feature vectors that represents both structural and textual aspects of XML documents. To extract structural information, employ an existing frequent tree-mining algorithm combined with an information gain filter to retrieve the most informative substructures from XML documents. The rule-mining algorithm extracts all structural rules having the support and confidence greater than a predetermined threshold. Every feature vector corresponds to an individual frequent substructure. The value of every individual feature for any particular document is set to 1 if the documents contains the left-hand side of the rule, otherwise the value is set to 0. However, for extracting content information from bag of words (BOW) representing XML documents,four main steps are involved:

- Document preprocessing (stop word removal and stemming)
- Building an inverted index
- Soft clustering of words
- Building the feature vector for every document

The obtained feature vectors are concatenated later and every XML document is represented by a single vector. The task of building a classifier model in this approach is then to employ the combined feature vectors, each of which represents an XML document from the training set, into an effective learning algorithm which uses these feature vectors to train a classifier model. This paper has used support vector machines (SVM) and decision-tree (DT) algorithms to build the classifier model.

Paper [11] use a bottom-up approach, i.e., we start from the text first, and then embed the structural information. This is based on the observation that in XML documents, the most informative information is carried by the terms in the content. Also some leaf node needs their parental node label to be informative. And tags are used to define the scope of the child tag and textual content below them. These assumptions are used for classification. For this purpose, a distinguishing set of key terms is searched within each class. A set of Key terms are informative and discriminatory of the respective class, but not of any other classes. Key terms are selected based on binary MI on all term key terms are cooperated with structural information for classification. Within the generic class, a key path is a root- to-leaf path ending on a leaf that contains at least one key term for the class. The set of all key paths from the structures of the XML documents belonging to a same class is regarded as the model for that class. The class models allow classification of an unlabeled XML document by means of a similarity-based scheme.The documents can be classified into the class with which it has highest similarity.Specifically, for each class, the similarity between the key paths in the corresponding class model and those in the structure of the previously unobserved XML document is suitably evaluated and the latter is eventually classified into the most resemblant class. The similarity between an XML document and a class in terms of the sum of the similarities between two paths.

In paper [12] an approach called XCLASS is used as an appropriate type of tree like substructures are chosen for classification of XML documents. It proposes an algorithm to learn rule-based classification models from appropriated types of structural properties. It uses CAR(class association rule) to model association between subset of cooccuring substructures and distinguished classes. Structural classification is divided into model learning and prediction. The first one learns associative classifier from database of laballed XML tree. Latter use tree to predict the classes for unlabeled one. In model learning preliminary step contains the definition of feature space for representing discriminating structure, then a mapping of the XML tree database into transactional form over features are done. It uses MINECAR procedure which is an enhancement of aprioiri alogorithm(Aggarwal) by using minimum support and minimum complement class support to produce meaningful CAR from training data. After that the rule set R is distilled into a compact associative classifier C through the pruning method PRUNE. Finals prediction is performed on unlabeled document tree based on the classifier obtained from model learning.

V. Conclusions and Future Work

The XML document is the semi structured representation for storing different dataset. classification of XML is challenging task because of its semi structured nature. This paper gives a brief survey about XML document classification based on two methods is by considering only its structural aspect and other one is by considering both its structure and contents. Our future work is to extend the idea of classification by applying semantic concept along with structure and content.

References

[1] M. J. Zaki and C. Aggarwal, XRULES: An effective structural classifier for XML data, in Proc. ACM SIGKDD, 2003, pp. 316325
[2] M. Theobald, R. Schenkel, and G. Weikum. Exploiting structure, annotation, and ontological knowledge for automatic classification of xml data. In Proc. of WebDB Workshop, pages 1 6, 2003
[3] X. Xin and J. Han. CPAR: Classification based on predictive association rules. In Proc. of SIAM Int. Conf. on Data Mining, pages 331335, 2003.
[4] C. Garboni, F. Masseglia, and B. Trousse. Sequential pattern mining for structure-based xml document classification. In Proc. of the INitiative for the Evaluation of XML Retrieval, pages 458 468, 2006.

[5] J. De Knijf. Fat-cat: Frequent attributes tree based classification. In Proc. of the INitiative for the Evaluation of XML Retrieval, pages 485 496, 2007.

[6] M. Theobald, R. Schenkel, and G. Weikum, Exploiting structure, annotation, and ontological knowledge for automatic classification of XML data, in Proc. WebDB, 2003, pp. 16.

[7] Mohammad Khabbaz, Keivan Kianmehr, and Reda Alhajj, Employing Structural and Textual Feature Extraction for Semistructured Document Classification, IEEE transactions on systems, man, and cyberneticspart c: applications and reviews, vol. 42, no. 6, november 2012

[8] R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval. Addison Wesley, 1999

[9] R. Baeza-Yates, B. Ribeiro-Neto: Modern Information Retrieval. Addison Wesley, 1999

[10] C.D. Manning, H. Schuetze: Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[11] Wu, J. and Tang, J. 2008. A bottom-up approach for xml documents classification. In Proceedings of the International Symposium on Database Engineering and Applications. 131137.

[12] Costa, G., Ortale, R., and Ritacco, E. 2011a Learning Effective XML Classifiers Based on Discriminatory Structures and Nested Content

[13] Yi, J. and Sundaresan, N. 2000. A classifier for semi-structured documents. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovey and Data Mining. 340344.

[14] Kurt, A. and Tozal, E. 2006. Classification of xslt-generated Web documents with support vector machines. In Proceedings of the International Workshop on Knowledge Discovery from XML Documents. 3342.

[15] Janez Brank; Marko Grobelnik; Nataa Milic-Frayling; Dunja Mladenic. Training text classifiers with SVM on very few positive examples

[16] H. T. Ng, W. B. Goh, and K. L. Low. Feature selection, perceptron learning, and a usability case study for text categorization. In Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 67-73, 1997

[17] McCallum, A. and Nigam, K. A Comparison of Event Models for Naive Bayes Text Classification. in AAAI-98 Workshop on Learning for Text Categorization, Madison,WI, 1998, 41-48.

[18] Guo G, Wang H, Bell D, Bi Y and Greer Y (2004): Using kNN Model for Automatic Text Categorization, Journal of Soft Computing, Springer-Verlag Heidelberg.

[19] J. R. Quinlan, Induction of Decision Trees, Machine Learning, 1(1), pp 81106, 1986.

[20] C. Apte, F. Damerau, S. Weiss. Automated Learning of Decision Rules for Text Categorization, ACM Transactions on Information Systems, 12(3), pp. 233251, 1994.