RESEARCH ARTICLE OPEN ACCESS

A Study on Pruning Techniques in Web Content Mining

S.Balan [1], Dr.P.Ponmuthuramalingam [2]
Ph.D Research Scholar [1], Controller of Examinations & Associate Professor [2]
PG & Research Department of Computer Science,
Government Arts College (Autonomous), Coimbatore
Tamilnadu - India

ABSTRACT
In the modern decade, information's are stored electronically in manner. It aims to relate the accurate data and resources. Some of the existing machine learning algorithms are automatically detects the structure of the data from a given data set. Those algorithms carried in to two step process first, identifying the structure and second, human-readable form. Both techniques are identified by manual. Disadvantages of this technique are not accurate and it takes time to produce the information. To overcome that decision trees are used to predict the accurate data via pruning techniques. Pruning is defined as process of cutting off non predictive parts. It determines the size and accuracy of relationship between the domains. This paper is concerned with study and analysis of pruning techniques, types of pruning, decision trees and existing methods. The advantages of these techniques are useful in practice.

Keywords :— Pruning, Pre-Pruning, Post-Pruning, Types of Pruning, Decision Trees.

I. INTRODUCTION

Decision tree is type of classification, predictive task and clustering. It main focuses the problem search using divide conquer technique to split the data in to sub sets. Some of the tree classification examples are neural networks, instance base learning and Bayesian networks. It offers the root node, input field used as splitting criterion, field values will be noted as (1,2...n and n+1,n+2....N), node or leaf, branch or segment formed by splitting rule. Some of the benefits are able to give input data as text, numeric and nominal, various tasks, and computational effort. It consists three phases namely construction phase, pruning phase and processing the pruning tree. Construction phase is based on entire training data set. It requires root node, variable name, alternative or competing splits, selected split or partitioning variable. Pruning phase is based on internal branch it contains lower nodes to improve performance from a set of rules.

Processing the pruning tree is based on to improve the understand ability of node identifier and multi way split or branch. The main strength of decision tree is understandable rules, clear fields for splitting the root node of the tree, continuous and categorical variables. The weakness is time series, less appropriate tasks, expensive, error prone to train the number of classes. Attribute selection measure is used to select the splitting criteria of the best separate partition of data set. There are three popular attribute selection measures namely information gain, gain ratio and gini index. The splitting criteria of a tree are categorized in many ways according to the origin of the measure and its structure. Some of the common criteria's are impurity based criteria, likelihood-ratio chi-squared statistics, normalized impurity based criteria, distance measure, binary criteria, towing Criterion, orthogonal criterion and kolmogorov-smirnov criterion.

Pruning is also called as accuracy for simplicity. There are various techniques are available for pruning decision trees. Some of the popular techniques are cost complexity pruning, reduced error pruning, minimum error pruning, pessimistic pruning, error based pruning, optimal pruning and minimum description length pruning. To compare the different pruning techniques the result indicates the issues of weighting instance, misclassification costs and handling missing values. There are two types of pruning namely pre pruning and post pruning. Pre pruning is defined by a tree halting construction early. Pre pruning is problem of attribute selection, multiple sets, parametric vs. non-parametric tests and statistical tests for attribute selection. Post pruning is remove sub trees from a fully grown tree, decision tree pruning and statistical significance, approximation, sparseness and testing significance.

II. PREVIOUS WORK: PRUNING AND DECISION TREES

Association rule mining is used to find the particular relationship in various data elements. It is used for high dimensionality, business or scientific purpose and fraud detection. The various methods of association rule mining to finding the frequent elements as follows: apriori algorithm, FP-Growth algorithm – market analysis, medical diagnosis [1, 9].

Some basic steps of KDD Process to prune data is data selection, data pre-processing, data transformation, pattern evaluation, knowledge representation, prune based data mining, reducing error rate in pruning [6].

Existing decision tree algorithms are J48 (implementation of C4.5 algorithm, sub tree replacement, moved the nodes to upward of the tree). REPTree (it is reduced error pruning tree fast decision tree learning based on information gain or reducing the variance). PART (simple algorithm used to produce the accurate set of rules called decision lists, combination of C4.5 and RIPPER rule learning). RIDOR (Ripple down Rule Learner is used to generate default rule first and least error rate to generate the exceptions) and JRip (it is used to increase the accuracy of rules by replacing or revising individual rule) [7].

Using J-Pruning to reduce over fitting in classification trees such as automatic induction of classification rules, inducing the decision trees, using classification trees for prediction, over fitting of rules to data, pre and post pruning of decision trees, using J-Measure in classification tree generation, measuring the information content of the rule, rule generation, interpretation of over fitting, limitations of the decision tree representation. Fuzzy rule pruning is used to classify or predict the patterns. Various rule simplification methods are fuzzy rule association based on weight, heuristic method, numeric data, shortening procedure, representation and simplification, simplify rules deals with removing the identical rules, specific rules, and retaining the generalized rules [5, 11, and 13].

Comparison of classification algorithms in supervised algorithm advantage is as follows: decision tree (easy & generate rules), Naive Bayes (fast, building & scoring), K-Nearest neighbour (effective & robust), Support vector machine (more accurate than decision tree classification), Neural networks (classify patterns and high tolerance). The above algorithms feature comparison is speed, accuracy, scalability, interpretability, transparency, missing val interpretation. Classification parameter are accuracy in general, learning speed is based on attributes and instances, classification speed, tolerance of missing values, irrelevant attributes, redundant attributes, noise and over fitting.[10].

The applications of decision trees are largely used in business, intrusion detection, energy modelling, E- Commerce, Image processing, Medicine, industry, Intelligent Vehicles, Remote sensing and web applications. Some of the frequent data sets used in decision trees are weka (Waikato Environment for Knowledge Analysis), GATree (Genetically Evolved Decision Trees), Alice d'ISoft, See5/C5.0, Balance scale, kr vs kp, glass, abalone, heart dieases, image segmentation, breast cancer, protein secondary structure, labor, bank marketing, credit approval, segment [2,3,4,8]

Some of the recent issues related to decision tree is fragmentation problem, replication problem, partitioning in continuous data, repetition problem, ID3, 4.5 tends to take multi valued attributes, ID3 algorithm does not backtrack in searching, decision tree algorithm does not provide incremental learning, handling range of input, XOR parity and Multiplexer Problem, Over fitting Decision Trees [12,14,15]

III. CURRENT STATUS OF WORK

Top-K retrieval with Prune setting our goal is to find the Top- K match for each possible subsequence of tokens in a text record $x = x_1, \ldots, x_n$. Given a maximum segment length L, for each segment sub-query $Q = q_i, q_{i+1}, \ldots, q_{i+j}$ where $1 \leq i \leq n - L + 1$ and $0 \leq j \leq L - 1$, we need to find the Top-K matches for Q with a given relation R provided the match-score is $\geq \in$.

The easiest way to share work across multiple sub queries is to cache data fetches. Tidlist fetches are cached because a token occurs in many sub-queries. We also cache records fetched during point queries because it is common for the same record to appear in the candidate set of overlapping segments. We call this algorithm that shares only data fetches but otherwise executes each Top-K independently our baseline Iterative algorithm. The sharing of computation, in particular, the list merges is more challenging because the Top-K algorithm above crucially depends on token scores V (t,Q) which change as Q changes. Therefore, the bounds of one query might make the results of merging a set of tidlists (tidlist refers to list of tuple-ids containing a given token) unusable for another.

Given a string **x**, we use the criterion of simple Top-k above to mark the tokens in a sub-query Q of **x** as weak or strong. A token is globally strong if it is strong in any of the sub queries, otherwise it is globally weak. We use L_i to denote the tidlist of a token x_i if it is weak; otherwise we denote it by S_i. The strong and weak tokens might be arbitrarily interleaved in x. Our strategy is to merge the lists for strong tokens completely for each sub query $Q_{ji} = \{x_j, \ldots, x_i\}$ progressively. As these are short tokens, a complete merge will not take much time compared to the time taken for partial merge in case of previous simpleTopK. But we gain since we can reuse the complete merge for other sub queries, where as we could not reuse the partial merged lists in Iterative algorithm.

In this case, since we perform complete merges we scan the tidlists in rid order so as to allow for faster merges. Then we iterate over each sub-query Q_{ji}. If all the tokens of Q are strong, we output top k records of the merged list $Z_{j,i}$, where $Z_{j,i}$ denotes the complete merger of all strong tokens from j to i. If all the tokens are weak, there will not be any result $\geq \in$ for Q_{ji}. If Q_{ji} contains both weak and strong tokens, we determine the complete merged list of all the strong tokens of Q_{ji}. A domain D is a subdocument of M where there is at most one best match substring in D.

If a given entity r is the only possible reference that corresponds to the best match substring in D, then D is one of

r's domain in M. Given a domain D of r divided into several segments and intervals, with seg_s referring to the strong

Segment. Intuitively, any possible match substring of r should contain seg_s, such that we only need to consider substrings containing seg_s. The detailed algorithm is described above, where cur is the substring we are processing, Active is the active substrings set, δ is similarity threshold, L is a length threshold and CandSet is the candidate match substring set.

Then we filter it using the best estimated contribution from weak tokens, to obtain candidate list and find the actual scores for records in the candidate list by point queries against the database. To make point queries efficient, we also cache the result of the point queries. This proposed Top-K retrieval is an efficient, yet effortless to implement algorithm.

IV. CONCLUSIONS

This research is concerned with study and analysis of pruning techniques and decision trees. Pruning is used to find the accuracy of the information, here discussed about types of pruning and existing techniques. To retrieve the data accurate is improved by top k retrieval performance. To improve the quality of data set decision tree is used and the types of existing decision tree algorithms and various issues, applications of decision trees and data sets are discussed in this survey. It can be further extended in the following directions of improving the efficiency and quality of different types of data sets in pruning.

REFERENCES

[1]. Agrawal, R. Srikant, "Fast Algorithm for MiningAssociation Rules", Proc. of the Int. Conf on Very Large Database, pp. 487- 499, 1994.

[2]. C Rui min, Wang Mio(2010) A more efficient Classification scheme for ID3. IEEE 2th International Conference on computer Science & Education.

[3]. Devashish Thakur, Nisarga Makandaiah and Sharan Raj D (2010). Re Optimization of ID3 and C4.5 Decision tree. IEEE Computer & Communication Technology.

[4]. Horvath, Tamas; Yamamoto, Akihiro, eds. (2003). Inductive Logic Programming Lecture Notes in Computer Science 2835.

[5]. Hisao Ishibuchi, Member, IEEE, and Takashi Yamamoto, Student Member, IEEE, "Rule Weight Specification in Fuzzy Rule-Based Classification Systems", IEEE Transactions On Fuzzy Systems, Vol. 13, No. 4, August 2005.

[6]. Han, J. Pei, Y. Yin, "Mining Frequent Patterns without Candidate Generation". Proc. of the ACM SIGMOD Int. Conf. on Management of Data, pp.1-12, 2000.

[7]. Han, J. Pei, Y. Yin and R. Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent Pattern Tree Approach", In Data mining and Knowledge Discovery, Vol. 8, pp.53-87, 2004.

[8]. Han Jing-ti and Gu Yu-jia (2009) Study on Handling Range Inputs Methods On C4.5 algorithm. IEEE International Forum on Computer Science – Technology and Application.

[9]. Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Morgan- Kaufmann Publishers, 2000.

[10]. Kotsiantis, Department of Computer Science and Technology "Supervised Machine Learning: A Review of Classification Techniques" - , University of Peloponnese, Greece

[11]. Min-You Chen∗ , D.A. Linkens,"Rule-base self-generation and simplification for data-driven fuzzy models ", Fuzzy Sets and Systems 142 (2004) 243–265. Elseiver ScienceDirect.

[12]. Rudy Setiono and Huan Liu. Fragmentation problem and Automated Feature Constructions.

[13]. Shehzad, "Simple Hybrid and Incremental Postpruning Techniques for Rule Induction", IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 2, February 2013

[14]. Wu, Xindong, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan et al. "Top 10 algorithms in data mining." Knowledge and Information Systems 14, no. 1 (2008): 1-37.

[15]. Zheng Yao, Peng Liu, Lei and Junjie Yin (2005) R-C4.5 Decision Tree Model and its Applications to Health Care Dataset.