International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

A Relative Analysis of Multi-Relational Decision Tree Learning Algorithm

Archie Jain[1], Abhimanyu Kumar[2]

[1]PDM College of Engineering, Haryana

[2]Suresh Gyan Vihar University, Jaipur

Abstract: ***This paper provides a comparative delve of the working and execution of MRDT algorithm with that of MRDTL-2, which works on the theory initially propounded by Knobbe et al. This paper also delineates some of the foible of MRDTL viz. calculation speed, exactitude and most importantly handling of missing values. We had used some of the real world data sets from multiparous data mining sweepstakes and accomplished a graphical comparison for the forenamed two approaches. Conclusion from the experiments implies that MRDTL-2 is convincingly more efficacious approach than its forerunners.***

Keywords: MRDTL, Graphical Analysis, Shortcomings of MRDTL, MRDTL-2, Efficiency of MRDTL-2

1. Introduction

The great advancement in the field of digital storage, massive through put data acquisition, and communication technologies has carved it achievable to withstand very enormous amounts of data in majority scientific and commercial domains. Relational database houses this great amount of data. Even though when the data repository is not a relational database, these are many times viewed suitably as heterogeneous data sources as if they area collection of relations [1] which are then deployed for the purpose of extracting, inferring and organizing information from multiple sources. Therefore, this topic of relational learning from relational databases started to gain significant considerations in the literature [2], [3], [5], [6], [7], [8], [9], [10], [11], [12], [13]. Knobbe et al. [4] proposed and highlighted a general framework formulti- relational data mining that uses structured query language (SQL) to extract the information required for building classifiers (e.g., graphs and decision trees)from multi-relational data. Based on this very framework, [14] proposed a multi-relational decision tree learning algorithm (MRDTL). Experiments reported by this demonstrated that decision trees framed by employing MRDTL have much precise results which are comparable to ones obtained using other algorithms on several multi-relational datasets.

2. Existing approaches to Relational

Learning

Various techniques which were proposed earlier for relational datamining are discussed as below:

Inductive Logic Programming (ILP) evolved from Induction which works as programming paradigm which uses first order logic to represent relations used as a major technique to develop models through machine learning algorithm and Logic Programming. ILP got evolved from its prominent focus on building algorithms for the processing of logic programs from domain and background knowledge (i.e. inferring or obtaining knowledge for some sources) to latest considerations for classification, regression association, clustering and analysis [10].As a reason of its flexible and expressive ways of defining domain knowledge and examples, the single-table and multiple-table representation of the data is acknowledged. Considering the other learning approaches, ILP has been one of the first and most detailed ones. Use of ILP in relational data mining has been limited due the differences in input specification and non- supportability of language in different ILP engines. In order to deal with different input specifications for various ILP engines and deal with the logic formalism and to integrate different input specifications for different ILP engines, Unified Modeling Language (UML) was proposed. [15]

First order extension of Bayesian networks is **Bayesian Logic Programming**, introduced as an explication and reformulation of PLPs, but also as a common framework for the previous mentioned approaches. In this kind of BLPs, the qualitative part of the Bayesian net is presented by a set of Bayesian definite clauses. The difference between this type of clauses and classical clauses is that every node in a BLP shows a random variable. Set of random variables are analogous to the least Herbr and model of this logical program, i.e.the group of all ground nodes that are logically necessitated by it. All facts directly influencing n are the parents of some random variable *n*. [8]

Multi Relational Data Mining as a term was initially used by [4] in a way to mark out a novel approach for knowledge discovery and relational learning from relational data bases and data consisting of complex/structured objects. In multi-relational data mining framework, the data model consists of many tables; each recounting features of particular objects, only one view of the objects is central to the perusal. By selecting one of the tables as targettable, the user can select the kind of objects to be analyzed. The main importance is that each record in the targettable will point to a single object in the database. Descriptive attribute from that table can be selected for classification or regression purposes can be chosen once the target table has been selected this is termed as the target attribute within the targettable.

3. Methodology

3.1Multi-relational decision tree learning algorithm proposed by [16] is an add-on to the logical decision tree

induction algorithm called TILDE proposed by [17]. In order to deal with records in relational databases MRDTL broadens TILDE's [18] approach. First order logic clauses are used to represent decisions (nodes) in the tree. With the help of this decision trees are created whose patterns of nodes are multi-relational in nature i.e., selection graphs. MRDTL adds decision nodes to the tree through a process of successive refinement as far as some termination criteria are encountered unlike propositional version of the decision tree algorithm [19].Once some termination criterion is met, a leaf node relative to its classis introduced instead. The decision of choosing node to be added at every step is influenced by a suitable impurity measure (e.g., information gain). To represent the set of all objects of interest in that relational data base MRDTL initiates with a single node from the tree root. This node tends to targetable T_0 together with the specific target attribute. Below is shown a general outline for the algorithm taken from [14]. In the algorithm the function optimal- refinement deals with every possible refinement that can be done to the current pattern Swith respect to the database D and then select, in a greedy approach the optimal refinement (i.e., optimal information gain). The plausible and possible set of refinements to be made at particular point while the process is clearly noticeable by the current selection graph, the database structure, and the multiplicity of the associations involved. The complement of the selection graph is denoted by *S* (i.e., objects not selected by S is selected from the database here. Binary splits decision trees are created using the induction algorithm outlined below-

tree_induction (T: tree, D: database, S: selection graph)
Input database D, selection graph S
Step 1 *R: = optimal-refinement(S)*
Step 2 ***if*** *stopping criteria(S)*
Step 3 ***return*** *leaf*
Step 4 ***else***
Step 5 *:= tree_induction (D; R(S))*
Step 6 *:= tree_induction (D; R(S))*
Step 7 ***return*** *node (Tl; Tr; R)*

Figure 1: General working structure of a decision tree learning algorithm

Derived from the algorithm above, Fig 2 specifically defines the work flow paradigm and provides a generic outline of its structure.

3.1.1 Refinements of Selection Graphs

As noted above, once the target attribute and the target table have been selected (i.e. the kind of objects completely defined are central to the analysis) a pool of possible refinements can be applied to the starting node representing *T0* so that the hypothesis is found to be insync with the data in the training database. [4] Proposed some possible ways of fine-tuning selection graphs.

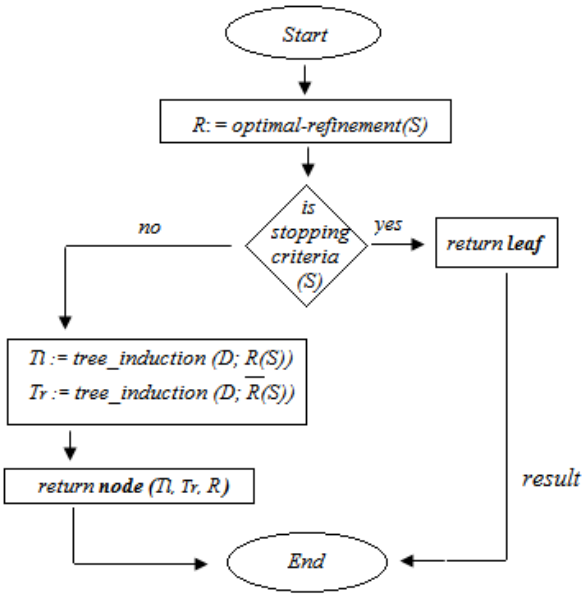

Figure 2: Flowchart depicting the work pattern of the algorithm

Add condition (positive): Without altering the structure of intended selection node *S*, this refinement simply adds a condition to it. Assuming the graph given below fig. 3(a) taken from [14], the refinement to be made was "atom. element = 'b' ". Prior to this operation, only the condition "atom.charge <= -0.392" was comprised of the set of conditions for atom node. Fig 3(a) shows the resultant graph after adding the discussed condition.

Add condition (negative): The complementary version of operation for the previous one is defined under this condition. This refinement adds a new absent edge from the parent node of that selection node and links to a new closed node that is a duplicate copy of the selection node that is being refined fig 3(b) if the node that is being refined does not represent the target table. Its condition list and join list (depicted by the edges coming out from this node) should be copied to the new closed node. Additionally the first list must be spread across by adding the new condition not negated.

Mutual Exclusion: The subsets which are derived from the same parent, associating with the patterns must be mutually exclusive hence two way of refinement, (add an edge and add a condition plus a node) are brought in with their complementary operations.

Adding open node and present edge: In this refinement, an association is epitomized in the data model as a present edge with its table represented as an open node and then these are added to selection graph.

Adding closed node and absent edge: This is the complementary to the previous one. Here an association is instantiated in the data model as an absent edge along with its corresponding table depicted as a closed node and these are added to selection graph.

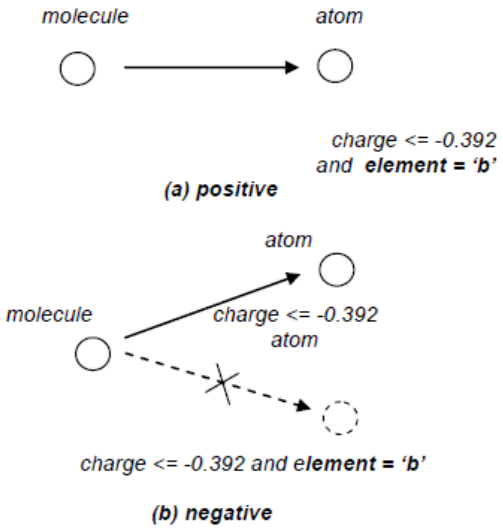

Figure 3: Graph depicting the "add condition"

Look-ahead refinement: In some cases employing optimal refinement may not give you optimum results in any kind of information gain while refining a selection graph. Thus in cases where any modification to particular selection graph does not result in any improvement, then that path is discontinued and instead a leaf node is introduced, even if it brings about future possible edges or conditions that are closely tied up to the search process.

3.1.2 Shortcomings of MRDTL

While implementing this algorithm [36] several shortcomings were found in literature in a nutshell many of the experiments reported in literature have shown that decision trees which were constructed using MRDTL have comparable accuracies to the ones obtained using other algorithms on many multi-relational data sets [36].However, from the viewpoint of multi-relational data mining on large data sets. MRDTL has two convincing shortfalls.

Slow running time: MRDTL [16] employs selection graphs for constructing the classifier in order to query the databases and to obtain the information, based on the multi-relational data mining framework. The experiments using MRDTL on data from KDD Cup 2001 [5] illustrated the plausible reason of hindrance in terms of the running time of algorithm that the results was the queries directed by such selection graphs which proved to be greatest hindrance.

Unable to handle missing attribute values: A significant portion of data has one or more missing valuesin many multi-relational real-world databases. For instance, in gene localization task from KDD Cup 2001 [21], 50% of COMPLEX, 70% of CLASS, and 50% of MOTIF attribute values are missing. The implementation of MRDTL [14] doesn't handle and consequently doesn't include any statistically well-managed methodologies to deal with missing values. Hence, the precision of decision trees constructed using MRDTL becomes a major concern as these missing value attribute are pretty common in real multi-relational datasets. For e.g. the accuracy of MRDTL on the gene localization task was reported approximately 50% in the literature

3.2 MRDTL-2

More efficient version of MRDTL is MRDTL-2 which is proposed by [22]. The concepts for MRDTF-2 were proposed by [14] which in a row are based on the algorithm proposed by [16] and the logical decision tree induction algorithm called TILDE [2]. MRDTL-2 and MRDTL both work in similar fashion, but in addition to the framework it suggested that some of the results of calculations that were performed in refinements in the decision tree and phase of adding nodes, can be reutilized at lower levels in the phase of further refinement of that given tree. Some unnecessary repeated work is performed each time by re-retrieving those instances already covered by selection graph previously in MRDTL while refining an existing selection was a problem. To avoid this redundancy storing those instances in a table which are covered by the selection graph from previous iteration in a table to can be a resolute. Hence, in MRDTL-2 with each iteration of the algorithm, primary keys from all front, open nodes of the selection graph for all the objects covered by it with its classification values are stored in a table cumulatively. The resulting table of primary keys is referred to as *sufficient table* for selection graph S and is denoted by Is. This table stores the 'skeletons' of the objects covered by that selection graph. The resultant table comprises wholly of the primary key as the table doesn't stores other attribute's information from the records except their primary keys. Following this technique, the number of tables that are needed to be joined becomes considerably less unlike in MRDTL in which the number of tables increases every time. The evident performance decline of MRDTL is due to this growth that was accountable as nodes get added to the decision tree. Hence, this mere change increases up the execution rate considerably.

3.2.1 Handling of Missing Values

In order to deal with missing attribute values in the data, MRDTL-2 incorporates a simple approach. For each attribute in a table a Naive Bayes model is constructed based on the other attributes without including the class attribute. Most acceptable values are then filled in the missing attribute with the most acceptable value which is predicted by the Naive Bayes for that corresponding attribute. Thereafter MRDTL-2 starts to build decision trees from the obtained tables, once the tables are pre-processed in the database using this technique, which contain no missing attribute values.

4. Experimental Results

The main focus of the experiment is on three data sets- the mutagenesis dataset which has been widely used in Inductive Logic Programming (ILP) research [20], the dataset for predicting thrombosis taken from PKDD 2001 Discovery Challenge [24] and the dataset for localization and function of protein/gene from KDD Cup 2001 [23].

The result we compared and analyzed graphically in context of accuracy factor with the best reported results in the literature which we were obtained using MRDTL-2 algorithm [22] and we also have concluded that MRDTL-2 is more accurate.

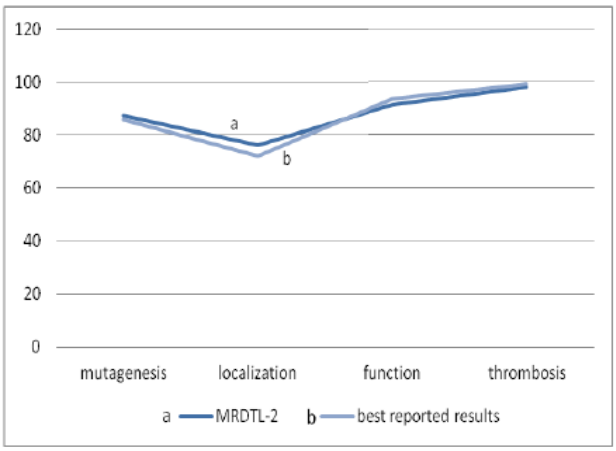

Figure 4: General performance (accuracy) comparison

Also, On the same datasets we also compared the execution time of the algorithm with those which are provided in the literature for other approaches and ended up with the results that MRDTL-2 outperforms all the other previous approaches.

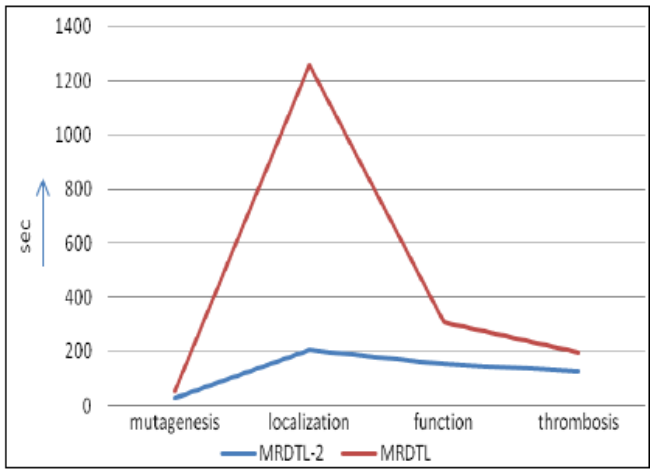

Figure 5: General performance (execution time) comparison

5. Conclusion and Discussion

It can be seen clearly that MRDTL-2 outplays all the other methods in the field of multi-relational data mining as a result of the comparison of MRDTL-2 performance with the best-known reported results for the same datasets from the literature. the ability of MRDTL-2 to handle missing attribute values which is a quite concerning problem is the major part of its better performance. To speed up the execution process and reducing the running time of the algorithm it also provides a better approach. So, MRDTL-2 outperforms all other previous approaches by overcoming aforementioned limitations and proves to be a significant method.

6. Acknowledgment

I would like to express my sincere gratitude to Mr. Abhimanyu Singh (B.Tech in CSE from Suresh Gyan Vihar University, Jaipur) for his valuable technical suggestions and guidance as a co-author towards the completion of this paper.

References

[1] Jaime Reinoso-Castillo: Ontology-driven information extraction and integration from Heterogeneous Distributed Autonomous Data Sources. Master of Science Thesis. Department of Computer Science. Iowa State University (2002)
[2] Hendrik Blockeel: Top-down induction of first order logical decision trees. Department of C.S., Katholieke Universiteit Leuven (1998)
[3] Friedman, N. and Getoor, L. and Koller, D. and and Pfeffer: Learning probabilistic relational models. In: Proceedings of the 6th International Joint Conference on Artificial Intelligence (1999)
[4] Knobbe, J. and Blockeel, H. and Siebes, A. and V an der Wallen D.: Multi-relational Data Mining. In: Proceedings of Benelearn (1999)
[5] Koller, D: Probabilistic Relational Models. In: Proceedings of 9th International Workshop on Inductive Logic Programming (ILP-99)
[6] INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 8, AUGUST 2013 ISSN 2277-8616 127 IJSTR©2013 www.ijstr.org
[7] Krogel, M. and Wrobel, S.: Transformation-Based Learning Using Multirelational Aggregation. In: Proceedings of the 11th International Conference on Inductive Logic Programming, vol. 2157 (2001)
[8] Getoor, L.: Multi-relational data mining using probabilistic relational models: research summary. In: Proceedings of the First Workshop in Multi-relational Data Mining (2001
[9] Kersting, K. and De Raedt, L.: Bayesian Logic Programs. In: Proceedings of the Work-in Progress Track at the 10th International Conference on Inductive Logic Programming (2000)
[10] Pfeffer , A.: A Bayesian Language for Cumulative Learning. In: Proceedings of AAAI 2000. Workshop on Learning Statistical Models from Relational Data, AAAI Press (2000)
[11] Dzeroski, S. and Lavrac, N: Relational Data Mining. Springer-Verlag (2001)
[12] L. Dehaspe and L. De Raedt: Mining Association Rules in Multiple Relations. In: Proceedings of the 7th International Workshop on Inductive Logic Programming, vol. 1297, p. 125-132 (1997)
[13] Jaeger, M: Relational Bayesian networks. In: Proceedings of the 13th Annual Conference on Uncertainty in Artificial Intelligence (U AI-1997)
[14] Karalic and Bratko: First order regression. In: Machine Learning 26, vol. 1997
[15] Hector Ariel Leiva: A multi-relational decision tree learning algorithm. M.S. thesis. Department of Computer Science. Iowa State University (2002)
[16] Knobbe, A. J., Siebes, A., Blockeel, H., and Van der Wallen, D. : Multi-relational data mining using UML for ILP . In: Proceedings of the First Workshop in Multi-relational Data Mining, 2001.
[17] Knobbe, J., Siebes, A., and Van der Wallen, D. M. G. Multi-relational decision tree induction. In Proceedings of the 3rdEuropean Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD, 1999.

[18] Blockeel, H. Top-down induction of first order logical decision trees. PhD dissertation, Department of Computer Science, Katholieke Universiteit Leuven, 1998.
[19] Blockeel, H., and De Raedt, L. Top-down induction of Logical Decision Trees. In W. K. Van Marcke, editor, Proceedings of the Ninth Dutch Conference on Artificial Intelligence (NAIC'97), 1997.
[20] Quinlan, J. R. Induction of decision trees. Machine Learning, volume 1, 1987
[21] Mutagenesis Dataset (http://web .comlab .ox.ac.uk/oucl/research/areas/machlearn/mutagenesis.html)
[22] Cheng, J. and Krogel, M. and Sese, J. and Hatzis C. and Morishita, S. and Hayashi, H. and Page, D.: KDD Cup 2001 Report. In: ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations, vol. 3, issue 2 (2002)
[23] Anna Atramentov: Multi-relational decision tree algorithm - implementation and experiments.M.S thesis. Department of Computer Science. Iowa State University (2003)
[24] The KDD Cup 2001 dataset: www.cs.wisc.edu/$\sim$dpage/kddcup2001
[25] The PKDD 2001 Discovery Challenge dataset: http://www.uncc.edu/knowledgediscovery

Author Profile

Archie Jain completed her graduation in the field of 'Information Technology' from PDM College of Engineering, Haryana and is currently working with Xerox Business Services, India

Abhimanyu Singh completed his graduation in the field of 'Computer Science' from Suresh Gyan Vihar University, Jaipur and is currently working with Xerox Business Services, India