MULTI-RELATIONAL DATA MINING CLASSIFICATION METHODS

Madhusudhan CH[1], Mrithyunjaya Rao K[2]

ABSTRACT

Multi-Relational Data Mining (MRDM) methods search for patterns that involve multiple tables (relations) from a relational database. In this paper we are proposed an improved method for constraint based classification of the Multi-relational data. MRDM aims to discover knowledge directly from relational data. Multi-Relational classification aims to build a classification model that utilizes information in different relations. Various techniques like Inductive Logic Programming (ILP), and Tuple ID propagation approaches are used in MRDM.

Keywords:

Data Mining, Multi-Relational Data Mining, MRDM architecture, MRDM methods, MRDM classification, MRDM classification algorithm.

I. INTRODUCTION

Multi-Relational data mining (MRDM) stores information in multiple tables. The fact that most data mining algorithms operate on a single relation or table, while most real-world databases store information in multiple tables and MRDM has a precursor going back over a decade in the field of inductive logic programming(ILP). Multi-relational data Mining (MRDM) search for patterns that involve multiple tables (relations) from a relational database. In a database for Multi-Relational classification, there is one target relation, Rt, whose tuples are called target tuples and are associated with class labels. the other relations are nontarget relations. Each relation may have one primary key (which uniquely identifies tuples in the relation) and several foreign keys (where a primary key in one relation can be linked to the foreign key in another relation). If we assume a two-class problem, then we pick one class as the positive class and the other as the negative class. The most important task for building an accurate Multi-Relational classifier is to find relevant features in different relations that help distinguish positive and negative target tuples.

II. ISSUES IN MULTI-RELATIONAL DATA MINING

2.1 Data Mining

The primary ingredient of any Data Mining [1][2][3] exercise is the database. A data base is an organised and typically large collection of detailed facts concerning some domain in the outside world. The aim of Data Mining described by the database. In Data Mining we generally assume that the database consists of a collection of individuals. Depending on the domain, individuals can be anything from customers of a bank to molecular compounds or books in a library. For each individual, the

[1]Research Scholar, Dept of CSE, Acharya Nagarjuna University, Nagarjuna Nagar, Guntur Dist, Anhrapradesh, India

[2]Director, Dept of MCA, Vaagdevi College of Engineering and Technology Warangal, Andhrapradesh, India

database gives us detailed information concerning the different characteristics of the individual, such as the name and address of customer of a bank, or the accounts owned.

When considering the descriptive information, we can select subsets of individuals on the basis of this information. For example we could identify the set of customers younger then 18. Such intentionally defined collections of individuals are referred to as subgroups. While considering different subgroups, we my notice that certain subgroups have characteristics that set them apart from other subgroups. For instance, the subgroup age under 18 may have a negative balance on average. The discovery of such a subgroup will lead us to believe that there is a dependency between age and balance of customer. Therefore, a method call survey of potentially interesting subgroups will lead to the discovery of dependences in the database. Clearly, a good definition of the nature of the dependency (e.g. deviating average balance) is essential to guide the search for interesting subgroups. Such a statistical definition is known as interestingness measure or score function.

Interesting subgroups are a powerful and common component of Data Mining, as they provide the interface between the actual data in the database and the higher-level dependencies describing the data. Some Data Mining algorithms are dedicated to the discovery of such interesting subgroups. However, interesting subgroups are a limited means of capturing knowledge about the database because by definition they only describe parts of the database. Most algorithms will therefore regard interesting subgroups not as the end product, but as mere building blocks for comprehensive descriptions of the existing regularities. The structures that are the aim of such algorithms are known as models, and the actual process of considering subgroups and laboriously constructing a complete picture of the data is therefore often referred to as modeling.

We can think of the database as a collection of raw measurements concerning a particular domain. Each individual serves as an example of the rules that govern this domain. The model that is induced from the raw data is a concise representation of the workings of the domain, ignoring the details of individuals. Having a model allows us to reason about the domain, for example to find causes for diseases in genetic databases of patients. More importantly, Data Mining is often applied in order to derive predictive models. If we assume that the database under consideration is but a sample of a large or growing population of individuals, we can use the induced model to predict the behavior of new individuals. Consider,, for example, a sample of customers of a bank and how they responded to a certain offer. We can build a model describing how the response depends on different characteristics of the customers, with the aim of predicting how other customers will respond to the offer. A lot of time and effort can thus be saved by only approaching customers with a predicted interest.

2.2 Propositional Data Mining

An important formalism in Data Mining is known as Propositional Data Mining. The main assumption is that each individual is represented by a fixed set of characteristics, known as attributes. Individuals can thus be thought of as a collection of attribute-value pairs, typically stored as a vector of values. In this representation, the central database of individuals becomes a table rows (or records) correspond to individuals, and columns correspond to attributes. The algorithms in this formalism will typically employ some form of propositional logic to identify subgroups, hence the name Propositional Data Mining.

Considering subgroups identified by constraints on the propositional data, e.g. 'the group of male customers who own more than one credit card' or 'the adults'.

Due to its straightforward structure, the Propositional Data Mining paradigm has been extremely popular, and is in fact the dominant approach to analysis a database. A wide range of techniques has been developed, many of which are available in commercial form. In terms of designing Data Mining algorithms, the propositional paradigm has a number of advantages that explain its popularity:

- Every individual has the same set of attributes. An individual may not have a value for a particular attribute (i.e. have a NULL value), but at least it makes sense to inquire about that particular attribute. Also, each attribute only appears once, and has a single value.
- Individuals can be thought of as points in an n-dimensional space. Distance measures can be used to establish the similarity between individuals. Density estimation techniques can be used to help discover interesting regions of the space.
- Attribute values are complementary; constraints on attributes divide the individuals in complementary subgroups. This makes gathering and using statistics concerning individual attributes- an important tool in Data Mining, as we shall see later on – a straightforward operation. Also, the use of operators is simple, as complementary operators correspond to complementary subgroups (e.g.= and #, or < and >).
- The meta-data describing the database is simple. This meta-data is used to guide the search for interesting subgroups, which in Propositional Data Mining boils down to adding propositional expressions on the basis of available attributes.

There is a single, yet essential disadvantage to the propositional paradigm: there are fundamental limitations to the expressive power of the propositional framework. Objects in the real world often exhibit some internal structure that is hard to fit in a tabular template. Some typical situations where the representational power of Propositional Data mining is insufficient are the following:

- Real world objects often consist of parts, differing in size and number from one object to the next. A fixed set of attributes cannot represent this variation in structure.
- Real-world objects contain parts that do not differ in size and number, but that are unordered or interchangeable. It is impossible to assign properties of parts to particular attributes of the individual without introducing some artificial and harmful ordering.
- Real-world objects can exhibit a recursive structure.

2.3 Structured Data Mining

The Propositional Data Mining paradigm has been popular because of the simple tabular structure it proposes. This property is, at the same time, its weakness. Many databases, especially of large industrial nature, are simply too complex to analyse with a propositional algorithm without ignoring important information. Rather than working with individuals that can be thought of as vectors of attribute-value data, we will have to deal with structured objects that consist of parts that may be connected in a variety of ways. Data Mining algorithms will have to consider not only attribute value information concerning parts (which may be absent), but also important information concerning the presence of different types of parts, and how they are connected.

Although a range of representations for structured data has been considered in the literature, structured individuals can conceptually be thought of as annotated graphs. Nodes in the graphs correspond to parts of the individual. A node can typically be of a class, selected from a predefined set of classes, and will have attributes associated with it. Available attributes depend on the class.

We refer to the class of techniques that support the analysis of structured objects as Structured Data Mining. These techniques differ from alternative techniques, notably propositional ones, in the representation of the individuals and of the discovered models. Many of the concepts of Data Mining are relatively representation-independent, and can therefore be generalized from propositional Data Mining. For example, individuals and interesting subgroups play the same role. What is different is the definition of subgroups in terms of structural properties of the individuals. Much of the remainder of this paperis dedicated to finding good ways of upgrading powerful concepts and techniques from Propositional Data Mining to the richer structured paradigm. Structured Data Mining deals with a number of difficulties that translate from the advantages of Propositional Data Mining listed in the previous section:

- Individuals do not have a clear set of attributes. In fact, individuals will typically consist of parts that may be queried for certain properties, but parts may be absent, or appear several times, making it harder to specify constraints on individuals. Furthermore, it will be necessary to specify subgroups on the basis of relationships between parts, or on groups of parts.
- Individuals cannot be thought of as points in an n-dimensional space. Therefore, good distance measures cannot be defined easily.
- Complementary subgroups can not be obtained by simply taking complementary values for certain properties, such as attributes of parts.
- The meta-data describing the database is extensive. Typically, the meta-data will not only describe attribute of the different parts, but also in general terms how parts relate to each other, i.e. what type of structure can be expected. Good structured Data Mining algorithms will use this information to effectively and efficiently traverse the search space of subgroups and models.

Over the last decade, a wide range of techniques for Structured Data Mining has been developed. These techniques fall roughly into four categories, which can be characterized by the choice of representation of the structured individuals. Although within each category the identification with the chosen representation is often very strong, it makes sense to view them in the broader perspective of Structured Data Mining. The four categories are:

- **Graph Mining [4][5][6][7]** The database consists of labeled graphs, and graph matching is used to select individuals on the basis of substructures that may or may not be present.
- **Inductive Logic Programming(ILP)** [8][9][10][11][12][13][14][15][16][17][18][19]The database consists of a collection of facts in the first-order logic. Each fact represents a part, and individuals can be reconstructed by piecing together these facts. First-order logic (often Prolog) can be used to select subgroups.
- **Semi-Structured Data Mining [20][21][22][23][24]** The database consists of XML documents, which describe objects in a mixture of structural and free-text information.
- **Multi-Relational Data Mining (MRDM)** The database consists of a collection of tables (a relational database). Records in each table represent parts, and individuals can be reconstructed by joining over the foreign key relations between the tables. Subgroups can be defined by means of SQL or a graphical query language.

III. Multi-Relational Data Mining

The approach to Structured Data Mining that is the main subject of this thesis, Multi-Relational Data Mining**[25][26][27][28][29][30][31][32][33]**, is inspired by the relational model[34][35][36]. This model presents a number of techniques to store, manipulate and retrieve complex and structured data in a database consisting of a collection of tables. It has been the dominant paradigm for industrial database applications during the last decades, and it is at the core of all major commercial database systems, commonly known as relational database management systems (RDBMS). A relational database consists of a collection of named tables, often referred to as relations that individually behave as the single table that is the subject of Propositional Data Mining. Data structures more complex than a single record and implemented by relating pairs of tables through so-called foreign key relations. Such a relation specifies how certain columns inn one table can be used to look up information in corresponding columns in the other table, thus relating sets of records in the two tables.

Structured individuals (graphs) are represented in a relational database in a distributed fashion. Each part of the individual (node) appears as a single record in one of the tables. All parts of the same class for all individuals appear in the same table. By following the foreign keys(edges), different parts can be joined inn order to reconstruct an individual. In our search for patterns in the relational database, we will need to query individuals for certain structural properties. Relational database theory employs two popular languages for retrieving information from a relational database relational algebra and the Structured Query Language (SQL). The former is primarily used in the theoretical settings, whereas the latter is primarily used in practical systems. SQL is supported by all major RDBMSs. In this paperwe employ an additional (graphical) language that selects individuals on the basis of structural properties of the graphs. This language translates easily into SQL, but is preferable because manipulation of structural expressions is more intuitive.

3.1 MRDM Architecture

Data Mining, and specifically the multi-relational variety, is a computation-intensive activity. Especially with serious applications of industrial size, a lot of data needs to be processed and scanned multiple times, in order to validate the large number of hypotheses generated by the algorithms such as covered in this text. We demand implementations of these algorithms to be at least moderately efficient and more importantly scalable, that is to perform predicatively with increasing data volumes. These demands force us to pay attention to the architecture of such implementations. In this section we propose a three-tiered client/server architecture that satisfies these demands. It has been applied with success in Propositional Data Mining software [37][38][39]. An important subject in Data Mining architecture is how data access if organized.

The architecture consists of three tiers: the presentation tier, the Data Mining tier and the database tier. Each tier will typically run on a separate workstation that is tuned towards the specific needs of the tier in question. For example the database tier will ideally run on a large server with ample main memory and fast access to secondary storage, whereas for the presentation tier, a regular desktop machine will suffice. Although separate workstations for each tier are optimal, several tiers can be run on a single machine. The presentation and Data Mining tier are an obvious choice, as they both have moderate computation requirements.

The database tier has exclusive access to the data. It is responsible for the actual scanning of this data, and answering questions about the frequency of certain patterns. Furthermore its task is to find an efficient way of processing the many (similar) queries. The Data Mining tier is responsible for guiding the search, and producing interesting hypotheses for the database tier to test. It will often consist of an MRDM kernel that deals with the basics of MRDM, such as the representation of data models, selection graphs and refinements etc., as well as a number of search algorithms that implement the different mining strategies. Finally, the presentation tier is responsible for interfacing with the user.

Each tier has a limited responsibility and can be optimized for this specifically. Communication between tiers is relatively limited. The graphical user interface communicates with the data Mining algorithm only at the beginning and end of a run. The Data Mining tier communicates with the database tier through predefined data mining primitives. This amounts to sending a small query and receiving back a compact list of statistics that describes the coverage of a list of patterns, as implied by the query.

Because of the relative independence of tiers, parts of the overall system can be replaced or optimized without interference with the remaining components. Specifically the database tier allows a range of RDBMSs to be employed. By using a vendor-independent database connectivity layer, such as ODBC or JDBC to communicate between the database and Data Mining tier, alternative databases may be tried without affecting the other components.

Because the hypothesis testing is separated from the mining process, and organized into primitives, the RDBMS is allowed to optimize the data access, for example by parallelizing the query processing. Also, queries stemming from different Data Mining tools working on the same data could be integrated. An interesting development is this respect is the work by Manegold et al.[40] on the database system Monet[41]. They observe that due to the nature of top-down (Propositional) Data Mining algorithms, many of the data mining primitives are concerned with very similar collections of data. By eliminating common sub-expressions in the query-graphs, and buffering partial results of previous queries, the query processing can be optimized considerably. This method works in a manner that is transparent to the Data Mining tier, and can thus easily be inserted in place of more mainstream RDBMSs.

IV. Multi-Relational Data Mining Methods

4.1 ILP Approach to Multi-Relational Classification

Inductive Logic Programming (ILP) is the most widely used category of approaches to Multi-Relational classification. There are many ILP approaches. In general, they aim to find hypotheses of a certain format that can predict the class labels of target tuples, based on background knowledge (i.e., the information stored in all relations).

Many ILP approaches achieve good classification accuracy , most of them are not highly scalable with respect to the number of relations in the database. The target relation can usually join with each nontarget relation via multiple join paths. Thus in a database with reasonably complex schema, a large number of join paths will need to be explored. In order to identify good features, many ILP approaches repeatedly join the relations among different join paths and evaluate features based on the joined relation. this is time consuming, especially with the joined relation contains many more tuples than the target relation.

4.2 Tuple Id Propagation

Tuple ID propagation is a technique for performing virtual join, which greatly improves efficiency of Multi-Relational classification. instead of physically joining relations, they are virtually joined by attaching the IDs of target tuples to tuples in nontarget relations. In this way the predicates can be evaluated as if a physical join were performed. Tuple ID propagation is flexible and efficient, because IDs can easily be propagated between any two relations, requiring only small amounts of data transfer and extra storage space. By doing so, predicates in different relations can be evaluated with little redundant computation.

Suppose that the primary key of the target relations is an attribute of integers, which represents the ID of each target tuple (we can create such a primary key if there isn't one). Suppose two relations, R1 and R2 , can be joined by attributes R_1.A and R_2.A. in tuple ID propagation, each tuple t in R_1 is associated with a set of IDs in the target relation, represented by IDset(t). For each tuple u in R_2, we set ID set (u) $=\cup_{t\in R_1, t.A=u.A}$ IDset(t). That is, the tuple IDs in the IDset for tuple t of R_1 are propagated to each tuple, u, in R_2 that is joinable with t on attribute A.

4.2.1 Multi-Relational classification using Tuple ID Propagation

CrossMine approach that uses tupe Id propagation for Multi-Relational classification. To better integrate the information of ID propagation, CrossMine uses complex predicates as elements of rules. A complex predicate, p, contains two parts:

1. prop-path: Tis indicates how to propagate IDs. If no ID propagation is involved, prop-path is empty.
2. Constraint: This is a predicate indicating the constraint on the relational to which the IDs are propagated. it can be either categorical or numerical.

CrossMine builds a classifier containing a set of rules, each containing a list of complex predicates and a class label. The algorithm of CrossMine is also a sequential covering algorithm like FOIL. It builds rules one at a time. After a rule r is built, all positive target tupels satisfying r are removed from the data set. To build a rule, CrossMine repeatedly searches for the best complex predicate and appends it to the current rule, until the stop criterion is met. a relation is active if it appears in the current rule. Before searching for the next best predicate, each active relation is required to have the IDset of propagated IDs for each of its tuples. When searching for a predicate, CrossMine evaluates al of the possible predicates on any active relation or any relation that is joinable with an active relation. When there are more than two classes of target tuples, CrossMinie builds a set of rules for each class. CrossMine uses tuple ID propagation to search for the beset predicate in all of the active relations,

Algorithm: CrossMine. Rule-based classification across multiple relations.

Input:

```
   D, a relational database;
   R_t a target relation
Method
   rule set R⇐ 0
   while (true)
   rule rê empty-rule;
   set R_t to active;
   repeat
   Complex predicate pê the predicate with highest foil gain;
if foil_gain(p)<MIN_FOIL_GAIN then
    break;
    else
rê r+p; // append predicate, increasing rule length by 1
remove all target tuples not satisfying r;
update IDs on every active relation;
   if p. constraint is on an inactive relation then
set that relation active;
endif
        until (r. length=MAX_RULE_LENGTH)
if r=empty-rule then break;
    Rê R∪ {r};
    remove all positive target tuples satisfying r;
set all relations inactive;
    endwhile
    return R;
```

This algorithm fails to find good predicates in databases containing relations that are only used to join with other relations.

V. Constraint-based Classification across Multiple Relations (Improved CrossMine Algorithm)

This algorithm facilitates to find good predicates in databases containing relations that are only used to join with other relations.

Input:

```
D, a relational database;
$R_t$ a target relation
```

Method

```
rule set R← 0
while (true)
rule rê empty-rule;
set $R_t$ to active;
for each r.length = MAX_RULE_LENGTH
Complex predicate pê the predicate with highest foil gain;
if foil_gain(p) < MIN_FOIL_GAIN then
rule= Seek_One_Rule(D,$R_t$,r);
break;
else
rê r+p; // append predicate, increasing rule length by 1
remove all target tuples not satisfying r;
update IDs on every active relation;
if p. constraint is on an inactive relation then
set that relation active;
endif
end for
if r=empty-rule then break;
Rê R υ { r};
remove all positive target tuples satisfying r;
set all relations inactive;
endwhile
return R;
```

VI. Conclusion

In this paper we have demonstrated that Multi-Relational Data Mining is inherently more powerful than Propositional Data Mining. There clearly is a large class of Data Mining problems that cannot be successfully approached using a single table a representational setting. These problems, which can be characterized by the presence of internal structure within the individuals they deal with, can successfully be approached by the multi-relational tools and techniques that are the subject of this thesis. MRDM techniques are not the only ones that deal with structured data. We have presented a genus of Structured Data Mining paradigms that each approaches the representation of data, and consequently the manipulation and analysis of the database, from a unique 'tradition'. MRDM is an important member of this family of paradigms. Its particular strength lies in how it employs a number of concepts from relational database theory to capture more complex characteristics of the data and achieve efficiency and scalability. Additionally the dominance in industry of its underlying relational tradition makes MRDM an obvious choice. Although our main emphasis has been on MRDM, we recognize the value of approaching problems in the more abstract setting of Structured Data Mining. By combining achievements that have been made relatively independently of one another, a richer set of techniques becomes available, and redundant development can be prevented. Furthermore a unified approach aids and comparison of existing techniques, which are mainly representation-specific. We therefore see the generalization of techniques from the individual paradigms, and integration of common ideas in SDM, as an important direction for future research.

VII. Acknowdedement

My thanks to all the experts who have contributed towards the development of the template.

References

[1] Dasu, T., Johnson, T. Exploratory Data mining and Data Cleaning, Wiley, 2003

[2] Fayyad, U., Piatetsky-shapiro, G., Smyth, P., Uthusamy, R., Advances in Knowledge Discovery and Data Mining, MIT Press, 1996

[3] Hand, D., Mannila H., Smyth, P., Principles of Data Mining, MIT Press, 2001.

[4] Cook, D., Holder, L. Graph-Based Data Mining, Intelligent Systems & their Applications, 15(2):32-41,2000.

[5] Inokuchi, A., Washiio, T. and Motoda, H. An Apriori-Based Algorithm for Mining Frequent Substructures from graph Data, In Proceedings of PKDD 2000, LNAI 1910,2000.

[6] Matsuda, T., Horiuchi, T., Motoda, H., Washio, T., et al. Graph-bases induction for general graph structured data, In Proceedings of DS'99, 340-342, 1999.

[7] Miyahara, T., Shoudai, T., Uchida, T., Kuboyama, T., Takahashi, K., Ueda, H. Discovering New Knowledge from Graph Data Using Inductive Logic Programming, In Proceedings of ILP '99, LNAI 1634, 1999.

[8] Blockeel, H. Top-Down Induction of First Order Logical Decision Trees, Ph.D. thesis, 1998.

[9] Blockeel, H., De Raedt, L.Relational Knowledge discovery in databases, In Stephen Muggleton, editor, In Proceedings of ILP '96, LNAI 314, pp. 199-211, 1996.

[10] Blockeel, H., De Raedt, L. Top-down induction of first order logical decision trees, Artificial Intelligence 101(1-2)285-297, June 1998

[11] Blockeel, H., De Raedt, L., Jacobs, N., Demoen, B. Scaling up inductive logic programming by learning from interpretations. Data Mining and knowledge Discovery, 3(1):59-93, March 1999.

[12] Blockeel, H., De Ratedt, L., Ramon, J. Top-down induction of clustering trees, In Proceedings of ICML '98, 55-63, 1998.

[13] Dehaspe, L. Frequent Pattern Discovery in First-Order Logic, Ph.D. thesis, 1998.

[14] Dzeroski, S. Inductive Logic Programming and Knowledge Discovery in Databases, in [33], AAAI Press, 1996.

[15] Dzeroski, S., Lavrac, N., An Introduction to Inductive Logic Programming, in [31].

[16] Dzeroski, S., Lavrac, N., Relational Data Mining, Springer-Verlag, 2001.

[17] Lavrac, N., Dzeroski, S. Inductive Logic Programming: Technique and Applications Ellis Horwood, 1994.

[18] Muggleton, S. Inverse entailment and Prolog. New Generation Computing, 13:245-286.

[19] de Raedt, L. (Ed). Advances in Inductive Logic Programming, IOS Press, 1996.

[20] Abe, K., Kawasoe, S., Asai, T., Arimura, H., Arikawa, S.Optimized Substructure Discovery for Semi-Structured Data. In Proceedings of PKDD 2002, LNAI 2431, 2002

[21] Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., and Arikawa, S. Efficient substructure discovery from large semi-structured data, In Proceedings of SDM2002, 2002.

[22] Braga, D., Campi, A., Klemettinen, M., Lanzi, P. Mining Association Rules from XML Data, In Proceedings of Da Wak 2002, 2002.

[23] Miyahara, T., Shoudai, T., Uchida, T., Takahashi, K and Ueda, H Discovery of frequent tree structured patterns in semistructured web documents. In Proceedings PAKDD2001, 47-52, 2001.

[24] Wang, K., Liu, H. Discovering structural association of semistructured data, IEEE Trans, Knowledge and Data engineering (TKDE2000), 12(3):353-371, 2000.

[25] Appice, A., Ceci, M., Malerba, D. MR-SMOTI: A Data Mining System for Regression Tasks Tightly-Coupled with a Relational Database, In Proceedings of KDD-2003,2003.

[26] Dzeroski, S., Lavrac, N., Relational Data Mining, Springer-Verlag, 2001.

[27] Klosgen, W., May, M. Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database, In Proceedings of PKDD 2002, LNAI 2431, 2002

[28] Knobbe A., De Haas, M., Siebes, A.Propositionalisation and aggregates, In Proceedings of PKDD 2001, LNAI 2168, pp. 277-288, 2001.

[29] Knobbe, A., Siebes, A., Blockeel, H., Van der Wallen, D. Multi-Relational Data Mining, using UML for ILP, In Proceedings of PKDD 2000, LNAI 1920,2000

[30] Knobbe, A., Siebes A., Van der Wallen, D. Multi-Relational Decision Tree Induction, In Proceedings of PKDD'99, LNAI 1704, pp.378-383, 1999

[31] Knobbe, A., Siebes A., Marseille, B. Involving Aggregate Functions in Multi-Relational Search, In Proceedings of PKDD 2002, LNAI 2431, 2002.

[32] Leiva, H., Atramentov, A., Honavar, V. Experiments with MRDTL- A Multi-Relational decision Tree Learning Algorithm, In Proceedings of Workshop MRDM 2002, 2002.

[33] Wrobel, S. An algorithm for multi-relational discovery of subgroups, In Proceedings of PKDD '97, 78-87.

[34] Date, C. An Introduction to Database Systems, Volume I, the Systems Programming Series, Addison-Wesley, 1986.

[35] Ullman, I.D. Principles of Databases and Knowledge-Based Systems, volume I, computer Science Press, 1988.

[36] Ullman, J., Windom, J., A first Course in Database Systems, Prentice Hall, 2001.

[37] Hahn, MInfo charger Data Sheet, http://www.tektonic.dc/icdatash.htm, 1997.

[38] Holsheimer, M., Kersten, M., Siebes, A. Data Surveyor: searching the nuggets in parallel, In [33]

[39] Knobbe, A. Towards Scalable Industrial Implementations of ILP, ILPNet2 Seminar on ILP & KDD, Caminha, Portugal, 1998.

[40] Manegold, A., Pellenkoft, A., Kersten, M. A. Multi-Query Optimizer for Monet, In proceedings of BNCOD 2000, LNCS 1832, 2000.

[41] Boncz, P. Monet, a next-Generation DBMS Kernel for Query-Intensive Applications, Ph.D. thesis, 2002.

Author's Biography

CH.Madhusudhan, received his B.Sc. from Osmania University in 1997, Advanced Diploma In Software Technology from ECIL, Hyderabad, M.C.A. from Kakatiya University in 2002 and M.Phil from Madurai Kamaraj University in 2009. Currently he is pursuing Ph.D from Acharya Nagarjuna University. His research interests include Data Mining, Data Base Management Systems, Software Engineering and Object Oriented Programming, Web Programming, Middleware Technologies

Prof. K .Mrithyunjaya Rao, HOD, Vaagdevi College of Engineering and Technology. Warangal, Andhrapradesh, India. His areas of research include Data Warehousing and Data Mining, Database Management System, Software Engineering.