Knowledge Discovery in Remote Access Databases

By

Zakaria Suliman Awad Zubi

A summary of the thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Mathematics and Computer Science
(Information Technology) at the

Institute of Mathematics and Informatics at

Debrecen University

May 2002

1. Motivation

The growths of the data in the databases are increased day by day and extracting all the interesting information from databases is impossible sometimes. Knowledge Discovery in Databases (KDD) is the overall process of discovering useful knowledge from databases, but also KDD is the nontrivial process of identifying valid novel, potentially useful, and ultimately understandable patterns in databases. Data Mining (DM) is not new to statistician; it is a term synonymous with data dredging or fishing and has been used to describe the process of trawling through data in the hope of identifying patterns, but for us DM is the fitting model to extract patterns from the databases. A clear distinction between KDD and DM is drawn under their conventions; the knowledge discovery process takes the raw results from DM and carefully and accurately transforms them into useful and understandable information. This information is not typically retrievable by standard techniques but is uncovered through the use of KDD or DM techniques. Some times we considered DM the same as KDD process, meanwhile, DM is a part of KDD process since it is only one step in the KDD process. A clear definition for DM and KDD could be the extraction of interesting *non-trivial, implicit, previously unknown and potentially useful information or patterns* from data in large databases. Alternative name such as mining databases led to knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, and business intelligence [3,121]. This information is not typically retrievable by standard techniques but is uncovered through the use of KDD or DM techniques. The basic steps of KDD are presented in Figure 1.1; these steps are defined also in [122].

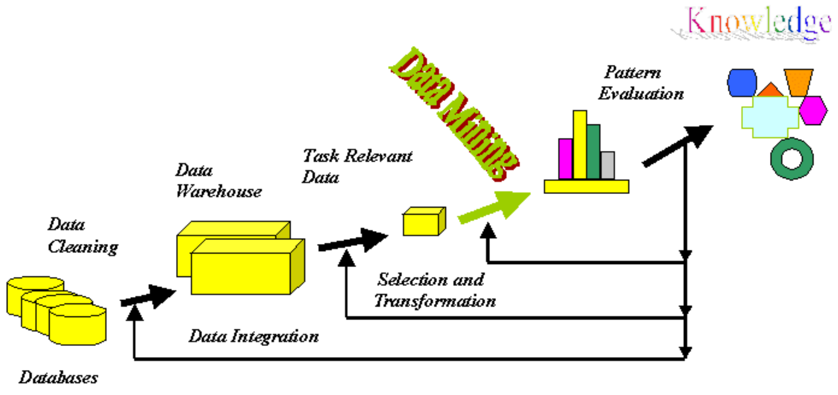

Figure 1.1 KDD and DM process.

The KDD process comprises of a few steps leading from raw data collections to some form of new knowledge. The iterative process consists of the following steps:

- **Data cleaning**: also known as data cleansing, it is a phase in which noise data and irrelevant data are removed from the collection.
- **Data integration**: at this stage, multiple data sources, often heterogeneous, may be combined in a common source.

- **Data selection**: at this step, the data relevant to the analysis is decided on and retrieved from the data collection.
- **Data transformation**: also known as data consolidation, it is a phase in which the selected data is transformed into forms appropriate for the mining procedure.
- **Data mining**: it is the crucial step in which clever techniques are applied to extract patterns potentially useful.
- **Pattern evaluation**: in this step, strictly interesting patterns representing knowledge are identified based on given measures.
- **Knowledge representation**: is the final phase in which the discovered knowledge is visually represented to the user. This essential step uses visualization techniques to help users understand and interpret the data mining results.

KDD steps also can be merged into two classes such as:

- *Data cleaning + data integration = data pre-processing.*
- *Data selection + data transformation = data consolidation.*

Before the terms "*knowledge discovery*" and "*data mining*" became popular, researchers in areas such as *artificial intelligence*, *machine learning*, *mathematical statistics*, *database management*, *visualization*, *pattern recognition, decision support,* and many others were all working on the same kinds of problems. However, without a name under which to unite, the research suffered from fragmentation. KDD unites all of these disciplines together under the premise that there exists much valuable knowledge in databases, but that due to the tremendous and growing volumes of data involved, advanced computer algorithms running on fast hardware are a more economical way to discover this knowledge than tedious manual searches and queries by human analysts [3].This fact can be shown in Figure 1.2.

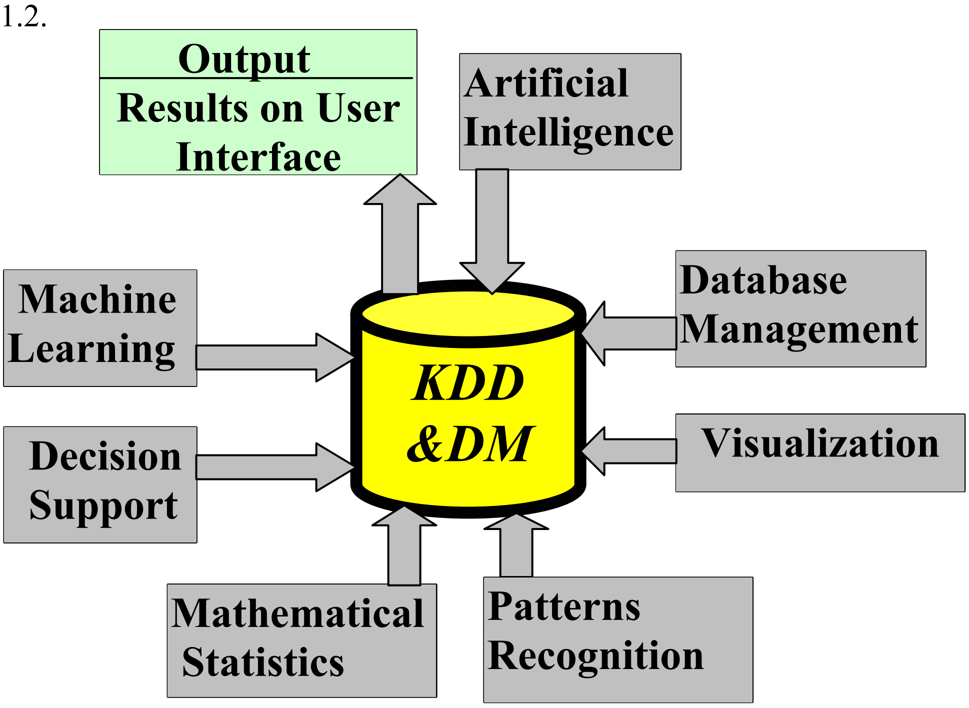

Figure1.2 KDD & DM shared with several topics

2. Main problems

In this section, the research problem of the thesis is considered, and related work is briefly reviewed. The task of using KDD queries to extract hidden information from remote databases can be summarized as follows.

- The growths of the data in the databases are increased day by day and extracting all the interesting information from databases is impossible sometimes. The size of the corresponding answer obtained by the query can be very large. This effect is called sometimes combinatorial explosion. Because of the large size of the result the user can't have an overview on the answer. Having stochastic nature and governed by statistical rules, knowledge remains hidden at combinatorial querying. The extraction of such kind of knowledge requires more sophisticated and compound methods.

- A massive amount of data are distributed and stored randomly in databases located in the internet. *Open Database Connectivity (ODBC)* enables us to get information from databases maintained by a remote server. KDD methods can be applied then to the retrieved data. However, this approach could be very expensive. Large amount of data should be transferred before starting the KDD processing. An alternative could be the implementation of some KDD processing facilities together with the database server. In this case the KDD processing could be carried out locally. This approach requires a clear formal way to formulate the goals of knowledge discovery, i.e., we need a formal *knowledge discovery query language (KDQL)*. On the other hand, the system should be capable to transfer and handle syntactically correct KDD queries over the network. The solution could be the use of the extended ODBC tools. These ODBC tools were suggested to handle KDD queries instead of classical SQL.

- What kind of architecture that could support the best remote access knowledge discovery? A possible model architecture that fulfills this requirements is *ODBC_KDD(2)* model. This model is introduced in Chapter 3 [26].

- How the knowledge discovery process could be formalized? This problem requires a clear modeling of the KDD process. We investigated different approaches to Logical Foundations in Data Mining issue. In this issue we used First-Order Language FOL to enable the discovered patterns to be described in a concise way, which in most cases increases readability of the output. Multiple relations can be naturally handled without explicit (and expensive) joins. We introduced a new syntactical query language denotation for KDQL syntax. This syntax could be found in Appendix A.

- The KDD process is a very time consumed one. Consequently, it is very important to find methods which can accelerate the knowledge discovery process. The concept of *I-extended databases* could help us to solve this problem.1 To formalize this concept and to show how it can be used throughout the whole process of DM we used the closure property of the framework. Then, it is possible to perform various tasks on these descriptions, like optimizing the selection of interesting properties or comparing two processes.

- The implementation of KDQL requires careful investigation of this problem. We considered this problem in a very simple case. Data visualization could be a very powerful DM solution because of its simplicity and expressiveness.

- The visualization results reports the generation of the patterns and the discovered results. These results help the user to determine, which patterns can be considered knowledge to make the correct conclusions.

- KDQL as a query language should have there own syntax formulation. We addressed this problem, and then we solved it by implementing new syntactical query language denotations for KDQL. The basic notions in KDQL came from SQL. SQL is extended to the notion of KDD query language that is defined theoretically in KDQL.

3. Organization of the dissertation

The thesis is structured in three parts. The first part presented the preface of this thesis into two Chapters. In Chapter 1 we presented the overview of the thesis. In Chapter 1 we gave an introduction to Data Mining (DM) and Knowledge Discovery in Databases (KDD). We also gave a motivation and the history of DM. Meanwhile, we indicated the importance, appearance and tools of DM and KDD approaches. In this Chapter also we motivated some related KDD and DM sub topics such as Open Database Connectivity (ODBC), Structured Query Language (SQL) and Data Visualization (DV) and we also described the importance, appearance and tools of each sub topic.

In Chapter 2 we expressed the aim and goal of this research work and the role of the current visualization approaches in KDD process. We also put some questions and problems, and then we listed the research goals in the end of this chapter.

The second part introduced Knowledge Discovery in Remote Access Databases Models, and it is divided into four Chapters. In Chapter 3 we presented Remote Access KDD Models and we introduced both ODBC_KDD models. We also addressed the main characteristics and architectures of both ODBC_KDD models. Moreover, we illustrated how to retrieve data by using one of ODBC_KDD models.

In Chapter 4, we presented a Logical Foundations in Data Mining (LFDM), and we pointed out the advantages and the disadvantages of using LFDM. We also located some DM tasks that are used in logical approaches such as Classification and Predication, Clustering and Data Summarization like Association Rules. In the final part of this chapter we used some SQL queries with extensions.

In Chapter 5, we presented Mining the Discovered Association Rules, we explained a general foundation of association rules, and then we tried to mine these rules by measuring the minimum and maximum support and confidence. We also measured the support and confidence by the properties of the itemset. We also expressed the interestingness measurements association rules.

In Chapter 6, we presented Data Mining Query Languages (DMQL) and we gave an introduction to the DMQL and how DM is a single step in KDD process?. We also pointed out some types of discovered data by DMQL. We also introduced discovered patterns, categorize, issues and environments in DM.

The third part presented the Knowledge Discovery Query Language Techniques. This part was divided into four Chapters. In Chapter 7, we presented Knowledge Discovery Query Language (KDQL). In this chapter we introduced the abstract and the introduction of KDQL which was proposed by us. We also showed the background of KDQL and how we can analyze the pre discovered data by visualization techniques using query languages tools.

In Chapter 8, we presented I-extended databases concept. In this chapter we presented I-extended database as a new database concept. We also illustrated how I-extended database could extract data from databases. We also introduced some association rules that could be proved in i-extended database.

In Chapter 9, we presented implementations of KDQL. In this chapter we motivated the KDQL and also we defined some rules that could be operated in KDQL. In this chapter, we showed the algorithms and architecture of KDQL. Moreover, we implemented a visualization tool to visualize i-extended database concepts and KDQL results.

In Chapter 10, we presented the conclusion of this thesis work. In this chapter, also we indicated the summary of this thesis work, the discussion and the research direction and the future work.

In this thesis, we presented two Appendices and the bibliography. In Appendix A, we presented the KDQL syntax and all the related denotations also we showed some syntactical examples of different query languages as well. In Appendix B we presented all the scripts that appear in the implemented I-extended database and KDQL tool. Finally, we listed all the references in the Bibliography.

4. New results and tools

In this thesis, we indicated some notions to make the results more reasonable. We proposed a remote access knowledge discovery model called ODBC_KDD (2). In ODBC_KDD (2) model we proposed a query language called KDQL. KDQL was suggested to interact to conceptual database called i-extended database. The KDQL and I-extended databases were theoretically implemented for mining the discovered association rules. Mining the association rules were defined by a set of expressions of a first order logical language. A similar approach to KDQL could be DMQL which is presented in [12, 122]. Syntax for KDQL RULES operation was also proposed in Appendix A and some examples was given as well. Visualizing the

KDQL results will be presented in Appendix B and some remarks regarding the implemented program was given also. Moreover, visualization mode in KDQL reviewed the result data chats in 2D and 3D forms. This technique was presented in a recent visualization query language called SQL+D query which was described in [29]. The major results of my thesis work are summarized as follows.

1. The idea of proposing a new remote access KDD model called ODBC_KDD (2) has been introduced in [26]. The aim of proposing this model is to build an attractive model that could get results with more detailed description such as visualization, scripts, statistical inferences and more. In the architecture of ODBC_KDD (2) model we indicated a KDQL query language. This query language plays a very important role in this work.

2. The use of a conceptual KDD remote access model emphasis us to access a database. A database that fits into KDQL in the ODBC_KDD (2) model were issued as well. We proposed a database concept, called I-extended database (I-ED) to be maintained and accelerated by the use of KDQL. We currently study KDD processes which needs different classes of patterns. I-extended database concepts were the result of new database concepts that could extract the interested information from the databases, and some practical use was presented also [37,91].

3. Earlier a Data Mining Query Language (DMQL) has been described in [12, 122], it was designed to explore and discover relations in relational databases. These relations hold interesting patterns or rules. These rules or patterns could be investigated by the association rules. Mining the association rules require a logical investigation, which will be maintained by fist order logical language. KDQL is a similar KDD query language to the DMQL unless KDQL discovers only the association rules. KDQL is a result of a new KDD query language which could discover association rules.

4. KDQL represents the retrieved data results in different 2D and 3D visual forms such as pie, points, lines and bars.

5. In logical approaches we used the First Order Logic (FOL) languages for the description of such patterns offer DM the opportunity of discovering more complex regularities which may be out of reach for attribute-value languages and classical statistical algorithms [123]. Without logical foundation approach we are unable to locate the entire supported data item (patters), and then confidence will be concluded. As a result of using support and confidence of data item we located the important associated rules from the databases by using I-extended database to be established by KDQL.

6. One of the important results in this work is an ODBC_KDD (2) model we proposed a query language called KDQL. This KDQL is an attractive part of the ODBC_KDD (2) model. It was theoretically investigated by us. In the Figure 10.1 we showed the KDQL connections in their environments.

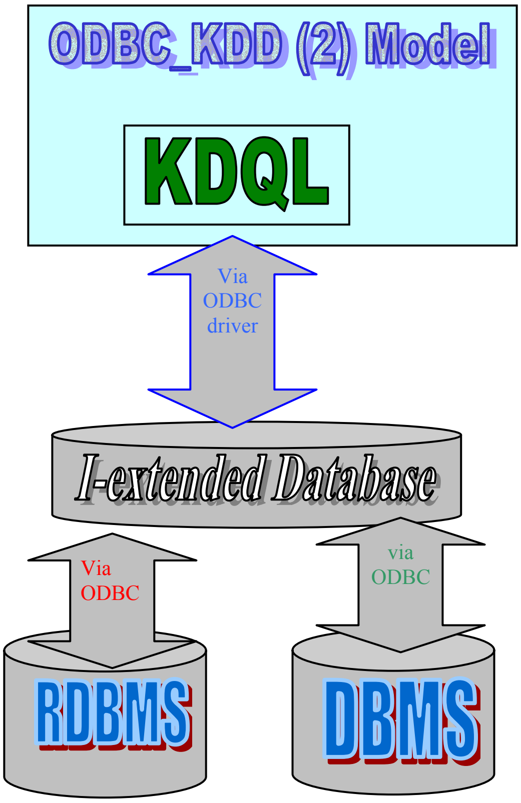

Figure 10.1 The environment of KDQL.

In this figure we illustrated that the KDQL is apart of the ODBC_KDD (2) model. KDQL called I-extended database via ODBC connection. I-extended database called all the requested information from the databases via specific ODBC diver from each database. KDQL was implemented to handle DM task and one of the DM common task is visualization. DM visualization techniques can be maintained to visualize interesting association rules discovered from the databases. The visual result of the KDQL query will be represented in 2D and 3D charts (i.e., Pie, Bar, Points, Lines). In KDQL, we use two different techniques of mining or discovering the association rules. We use a technique similar to the DMQL in DBMiner system in [12, 122]. The second is the visualization techniques. It was also similar to SQL+D visual query in [29], which can retrieves and then represents charts, graphs, and images stored in the database. The KDQL program was written in Delphi programming language. Syntax to the KDQL has been given and some examples were also shown in Appendix A.

5. Future works

Some inserting future research problems will be presented, respectively, in this section. The main focus of my future work will be to improve all the related components of the ODBC_KDD (2) model which was proposed in [26]. Moreover, my future task work will be listed as follows:

1. In the ODBC_KDD (2) model we suggested to develop a special ODBC driver in both sides (client / server). This driver is called Extended ODBC driver (EODBC). The aim of this driver is to handle the KDQL syntax instead of the classical SQL query statements.

2. In the server side in ODBC_KDD (2) model we proposed an interpreter to be the interface between the Extended ODBC (EODBC) driver and the SQL driver. The main benefit of this interpreter is to translate the KDQL syntax into common SQL syntax and inversely, from SQL to KDQL syntax. This interpreter helps the requested KDQL syntax to reach the data safely. SQL is a standard query language that could access any type of traditional databases.

3. Databases served in the server side in ODBC_KDD (2) model the SQL application can access only one type of databases which is DBMS. This problem led us to propose a new database concept for the model. This database is called I-extended database. This I-extended database is used to give reliability to the model. It is clear, from the context in Chapter 8, that the operation can also be applied on I-extended database instances while formally; we should introduce new notations for them. This requirement could be identify if we add more practical capability to I-extended database by using I-extended database as a part of the whole KDD process which includes extracting all the interesting association rules from many other database sources. These rules could be temporarily stored in I-extended database. The other part of the KDD process will be maintained by the KDQL which collects the most interesting rules from I-extended database.

4. I would like to implement a new *core operator* that provides an encoding of association rules [42]. If I will succeed in implementing a new core operator, then it should be possible to design a new post-processor to decode the rules and put the relations containing the desired rules in the table which is stored in I-extended database.

5. It would be useful to add more capability to the visualization mode in the KDQL program. KDQL program require new implementation approaches such as *simplified display specifications syntax and instantiation of display elements*, *implementing of directed and undirected graphs as a visualization aid for extracted data*, *implementation and new syntax of charts with categorization*, *directed and undirected graphs for advanced visualization of data and implementation and expanded features for temporal presentations with time constraints using simple and intuitive in-the-query specifications*.

References

[3] Awad Zakaria, Fazekas Gábor **,** *On Some Software Tools For Data Mining* ,4th International Conference on Applied Informatics**,** Eger-Noszvaj, pages 331-336, Hungary, 30 August-3 September 1999.

[12] Han, J., Fu, Y., Wang, W., Koperski, K., Zaiane, O.: *DMQL: A Data Mining Query Language for Relational Databases*. In Proc. 1996 SiGMOD' 96 Workshop Research Issues on Data Mining and Knowledge Discovery (DMKD'96), pages 27-34, Montreal, Canada, June 1996.

[26] Awad Zakaria, Fazekas Gábor, *On ODBC_KDD models*, paper,5th International Conference on Applied Informatics, dedicated to the 70th birthday of Prof. Mátyás Arató´ and Prof. László Varga, P-12, 28 January-3 February 2001, Eger, Hungary,2001.

[29] Graciela Gonzalez, Chitta Baral and Amarendra Nandigam, *SQL+D: Extended Display Capabilities for Multimedia Database Queries*, paper, ACM Multimedia 98 - Electronic Proceedings,USA, 1998.

[37] De Raedt, L., Dehaspe, L. *Clausal Discovery*. Machine Learning, 26, 99-146. 1997.

[42] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, I. *Fast Discovery of Association Rules*. Advances in Knowledge Discovery and Data Mining. Ed. U. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy. MIT Press. 1996.

[91] A. Tuzhilin. *A pattern discovery algebra*. In SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Technical Report 9707 University of British Columbia, pages 71 -76, 1997.

[121] Zakaria Awad, *Using Analysis Tools to Represent Data Mining Solutions,* Technical Report No. P-12, 99/81, Institute of Mathematics and Informatics, University of Debrecen, Debrecen, Hungary, 1999.

[122] Zakaria Awad, Gábor Fazekas, *Data Mining Query Languages*, Technical Report No: 2001/15, Preprints No. 273, P-13, Institute of Mathematics and Informatics, University of Debrecen, Debrecen, Hungary, 2001.

[123] Mohamed Omar, Ali Miloud and Zakaria Awad, *Smart Query Answering by KDD Techniques*, paper, 3rd International Conference on Computer Science, In print, Tripoli, Libya, 2001.