ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

An Agent Based Analysis of Knowledge Discovery in Databases

R.Hemamalini*, Dr.L.Josephine Mary

*Research Scholar St.Peter's University, Asst. Professor,CS Dept., Jaya College of Arts and Science-Thiruninravur, India

Professor & HOD, MCA Dept, Sri Ram Engineering College, Veppampet, India

hemasureshbabu@yahoo.co.in

Abstract

Discovering useful knowledge from database is referred as KDD. In the quest of knowledge possible interpretation of patterns and evaluation makes the decision of what is knowledge and what is not. It also includes the choice of programming schemes, systematic series of actions, sampling, and the state or fact of the data prior to the data mining step. Data mining refers to the application of algorithms for extracting patterns from data without the additional steps of the KDD process.

The Knowledge Discovery in Databases model is a step by step process for finding interesting patterns in large amounts of data. Data mining is one step in the process. This paper "An Agent Based Analysis of Knowledge Discovery in Databases "defines the KDD process and discusses the agent work for the KDD process as a potential for performance evaluation . The focus of this paper is to first summarize exactly what the KDD process is and the work done by the KDD process, and association of the KDD agent and the software agent and the agent work with KDD. The activities of these agents are coordinated when a process is instantiated and executed. The vehicle for agent coordination during process execution is an agenda management system (AMS) is also seen.

Keywords:

Introduction

The world consists of lots of facts. The raw fact is called as data and processed data is called as information. And to know the world better is knowledge. And knowledge is just closer to Intelligence. In this speeded world there is a need for computational theories and knowledge to assist us to discover new things .This extraction of knowledge from a data bases is called data mining. Knowledge discovery in databases (KDD) develops the methods and techniques to make data with sense.

The KDD process has a basic problem of mapping low-level data which are too large to understand as it may be more useful or abstract or compact. The result of the KDD process is the application of specific data-mining methods for pattern discovery and extraction. Knowledge Discovery in Databases (KDD) is the automated discovery of patterns and relationships in large databases. The problem addressed by KDD is to find patterns in these massive datasets. Traditionally data has been analyzed manually, but there are human limits. Large databases offer too much data to analyze in the traditional manner.

Agents can help navigate and model the World-Wide Web is another area growing in importance. Uncertainty in AI includes issues for managing uncertainty, proper inference mechanisms in the presence of uncertainty, and the reasoning about causality, all fundamental to KDD theory and practice.[1] Knowledge representation includes philosophical study of the nature of being, new concepts for representing, storing, and accessing knowledge. Also there are schemes for representing knowledge and allowing the use of prior human knowledge about the underlying process by the KDD System.

What is the KDD Process?

The term Knowledge Discovery in Databases (KDD), refers to the broad process of finding knowledge in data, and the "high-level" application of particular data mining methods. It also concentrates on researchers in artificial intelligence, machine learning, data visualization, databases, statistics, knowledge acquisition for expert systems, and pattern recognition.

The main goal of the KDD process is to extract knowledge from data in the context of large databases. This is done by using data mining methods (algorithms) to extract and identify what is deemed knowledge, according to the specifications of measures using a database along with any required

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

preprocessing, replication, and transformations of that database.

Knowledge representation includes philosophical study of the nature of being , new concepts for representing, storing, and accessing knowledge. Also included are schemes for representing knowledge and allowing the use of prior human knowledge about the underlying process by the KDD System[2]. These potential contributions of AI are but a sampling; many others, including human computer interaction, knowledge-acquisition techniques, and the study of mechanisms for reasoning, have the opportunity to contribute to KDD . In conclusion, we present some definitions of basic notions in the KDD field. Our primary aim was to clarify the relation between knowledge discovery and data mining. We provided an overview of the KDD process and basic data-mining methods. Given the broad spectrum of data-mining methods and algorithms, our overview is inevitably limited in scope: There are many data-mining techniques, particularly specialized methods for particular types of data and domain. Although various algorithms and applications might appear quite different on the surface, it is not uncommon to find that they share many common components. Understanding data mining and model induction at this component level clarifies the task of any data mining algorithm and makes it easier for the user to understand its overall contribution and applicability to the KDD process. This article represents a step toward a common framework that we hope will ultimately provide a unifying vision of the common overall goals and methods used in KDD. We hope this will eventually lead to a better understanding of the variety of approaches in this multidisciplinary field and how they fit together.

Why KDD?

- ✓ Large databases are not usual
  - Data's from business, telephone, government records, medical records, and credit card data
  - Scientific instruments can produce terabytes and petabytes at rates of gigs per hour
  - Storage capabilities are larger and Cheaper
- ✓ Databases growing in field size
- ✓ Databases growing in record size
- ✓ Human limits

An Outline of the Steps of the KDD Process

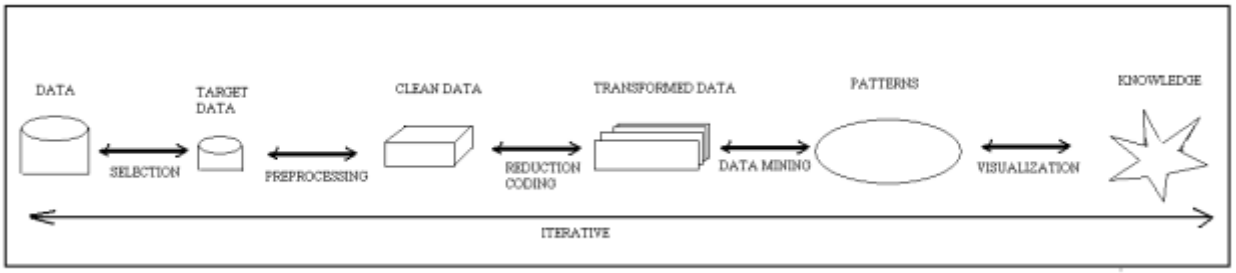

Figure : 1 Steps of the KDD Process

Figure 1 states that initially all the datas are recognized in several locations, and they are brought to one location called as Data Warehouses or Data Marts. Data transformation may be performed on the raw data before it is placed in the data warehouse. Data transformation resolves inconsistencies between the data of one location from that of another. For example, inconsistencies may be differences in data type for the same information and different field names for the same data field. Data warehouses hold both detailed and summary data. Detailed data is used for pattern analysis, where summarized data may hold the results of previous analyses[3]. Data warehouses also contain historical data whereas operational data is usually current. Efficient organization of the data warehouse is essential for efficient data mining. Once the data is organized, a selection process occurs where some subset of this data becomes the target data upon which further analysis is performed. It is important when creating this target data that the data analyst understand the area, the end users needs, and what the data mining task might be.

When data is collected in an informal manner. Data entry mistakes can occur and the data may have missing or unknown entries. During the data cleaning and preprocessing stage noise is removed from the data. Outliers and deviation in the data can cause special problems for the data analyst during the data cleaning process. The goal is to find rare patterns in the data, the outliers and deviation may be representations of these rare patterns. Care must be

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

taken not to remove these types of outliers and deviation. This step may be time consuming. Data Reduction and Coding step employs transformation techniques that are used to reduce the number of variables in the data by finding useful features with which to represent the data.

Data Mining is one of the step in the KDD process. This transforms the data for data mining step. The patterns of interest is performed in this step. The analysis is being performed and the search for pattern is done within the context of the data mining task and the representational model. It is important at this stage to decide on the appropriate data mining algorithm, for the data mining task . The data mining task itself can be a classification task, such as cluster analysis, rule formation, or linear regression analysis. The data mining steps may not generate data's which are new or interesting. So we have to remove the repeated and irrelevant patterns that are taken from the set of useful patterns. Once a set of good patterns have been discovered, they then have to be reported to the end user. This can be done textually, by way of reports or using visualizations such as spreadsheets, diagrams and graphs etc. The result is interpreted as knowledge by the interpretation step. Interpretation may require resolving possible conflicts with previously discovered knowledge since new knowledge may even be in conflict with knowledge that was believed before the process began. Knowledge is documented and reported to interested parties when it is done to user's satisfaction. This again may involve visualization[4]. It is important to stress that the KDD process is not linear. Results from one phase in the process may be fed into different phases. Current KDD systems have a highly interactive human component. Humans are involved with many if not each step in the KDD process. Hence, the KDD process is highly interactive and iterative.

The overall process of finding and interpreting patterns from data involves the repeated application of the following steps:

1. Developing an understanding of
    - The area of application
    - The prior knowledge which is relevant
    - The end-user goals
2. Creating a target data set: selecting a data set, or data samples , focusing on a subset of variables on which discovery is to be performed.
3. Data cleaning and preprocessing.
    - Removing of noise and human errors.
    - Modeling and accounting for noise in the collected information.
    - Strategies for handling missing data fields.
    - Accounting for time sequence information and known changes.
4. Data reduction and projection.
    - Depending on the goal of the task finding usseful features to represent the data.
    - To find invariant representations for the data using dimensionality, transformation or reduction methods to reduce the effective number of variables to be considered.
5. Choosing the data mining task.
    - Deciding whether the goal of the KDD process is classification, regression, clustering, etc.
6. Choosing the data mining algorithm(s).
    - Deciding which models and parameters may be appropriate.
    - Selecting method(s) to be used for searching for patterns in the data.
    - Matching a particular data mining method with the overall criteria of the KDD process.
7. Data mining.
    - Searching for patterns of interest in a particular representational form or a set of such representations as classification rules or trees, regression, clustering.
8. Interpreting mined patterns.
9. Consolidating discovered knowledge.

The terms knowledge discovery and data mining are distinct.

Definitions Related to the KDD Process

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

Data	A set of facts, F
Pattern	An expression E in a language L describing facts in a subset FE of F.
Process	KDD is a multi-step process involving data preparations, pattern searching , knowledge evaluation , and refinement with iteration after modification
Valid	Discovered patterns should be true on new data with some degree of certainty
Novel	Patterns must be novel.
Useful	Patterns should lead to useful actions.
Understandable	Patterns should make way to understand the data .

Software agents

Software agents are independent programs. At the same time as there are a number of different

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

types of software agent, the unifying feature is that they require less human interaction than a conventional program with a user interface. For example, multi-agent systems are nothing but distributed artificial intelligence programs that operate without a central controller. Network agents are programs that carry out users' wishes on remote computers[5]. User-Interface agents are programs that seek to relieve users of everyday tasks, such as electronic-mail filing, or scanning news items. The autonomy is generally the result of providing the agent with knowledge about its environment, as well as the actual task.

Network agents have knowledge about user's preferences, routing costs, etc. User Interface agents have knowledge about a user's interests and habits, and often they have the capability to learn new knowledge for themselves. This report draws its ideas from the area of research concerned with multi-agent systems. The research about multi-agent systems aims to produce co-operative agents that interact with other agents. There are a number of new technologies designed to implement software agent systems.[6] Agent-Oriented Programming is a methodology for specifying agent behavior. The Knowledge Sharing Effort is a set of standards designed to promote open inter-agent communication. Tele script provides a language and environment for network agents. For more details on the different kinds of agents, and the new technologies being used to build software agents, see . The remainder of this section examines the intersection between machine learning and software agents, as we intend to build agents that employ learning algorithms of the type mentioned above.

One of the key features that agents ought to possess is the ability to adapt and learn, both about other agents and their environment. An agent may either apply learning to its own internal mechanism, or to the task that is designed to do. However, an agent's primary task could be to apply a learning algorithm. These we call learning agents. This report is concerned with this use of learning. It is actually quite difficult to characterize learning agents - as learning usually has some associated performance task. This can be an application which drives the learning component. The alternative is that another agent (human or software) provides a learning goal. However in both cases the major feature is that learning takes place.

KDD agent

The KDD process is defined as a structured source of intellectual agent called KDD agent.[7]

Classification agents of data source

An important tool in understanding complex systems is the analysis of large data sets which is in producing decisions using source-based data. They are subjects of training performed by agents.

Server of training and testing methods

This component comprises a multiple of classes Of software which implements particular KDD methods and quality metrics, etc.

Meta-Learning agent

It computes the training and testing meta-data sample and also manages design of meta-model of decision making and Manages the distributed design of DDM MAS application

Meta-level KDD agent

The meta resources bounded rational agent and trains and tests the level classification agent and assesses its quality .The sequence of domain and control actions without consuming too many resources in the process.

Decision making management agent

It coordinates operation of Agent-classifier of meta-level and Meta-level KDD agent both in training and decision combining modes of their performance.

Server of KDD methods

This component comprises a multitude of classes implementing particular KDD methods, metrics, etc.

Decision combining management agent

It coordinates operation of Agent-classifier of metalevel and Meta-level KDD agent.

Agent-classifier of meta-level

It is subject of training and testing performed by Meta-level KDD agent of DDM MAS.

Potential role of AI in KDD

AI fields can potentially contribute various aspects of the KDD process. Some examples of these areas are: Natural language presents significant opportunities for mining in free-form text, especially for automated annotation and indexing prior to classification of text .

Limited translating capabilities can help substantially in the task of deciding what an article refers. Hence, the spectrum from simple natural language processing all the way to language understanding can help significantly[8]. Also, natural language processing can contribute significantly as an effective interface for stating hints to mining algorithms and visualizing and explaining knowledge derived by a KDD system. Planning consists of a complicated data analysis process. It involves conducting complicated data-access and data-transformation operations; applying preprocessing routines and, in some cases, paying attention to resource and data-access constraints. Typically, data

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

processing steps are expressed in terms of desired post conditions and preconditions for the application of certain routines, which lends itself easily to representation as a planning problem. In addition, planning ability can play an important role in automated agents to collect data samples or conduct a search to obtain needed data sets.

Intelligent agents[9] can be fired off to collect necessary information from a variety of Sources. In addition, information agents can be activated remotely over the network or can trigger on the occurrence of a certain event and start an analysis operation. Finally, agents can help navigate and model the World-Wide Web another area growing in importance. Uncertainty in AI includes issues for managing uncertainty, proper inference mechanisms in the presence of uncertainty, and the reasoning about causality, all fundamental to KDD theory and practice.

Coordinating agents at process execution time

In this section, we describe how the activities of these agents are coordinated when a process is instantiated and executed. The vehicle for agent coordination during process execution is an agenda management system (AMS)[10]. An agenda management system is a software system that is based on the metaphor of using agendas, or to-do lists, to coordinate the activities of various human and automated agents. In such a system, task execution assignments are made by placing agenda items on an agenda that is monitored by one or more execution agents. Different types of agenda items may be used to represent different kinds of tasks that an agent is asked to perform. Our agenda management system is composed of a substrate that provides global access to AMS data, a set of root object types (agendas, agenda items, etc.), application specific object types that extend the root types, and application-specific agent interfaces

An agent typically monitors one or more agendas to receive tasks to perform. Multiple agendas are used because an agent may frequently be involved in several disjoint processes or acting in roles that are logically disjoint. When an item is posted to an agenda that an agent is monitoring, the agent is notified that the agenda has changed. In the case of a human agent, for example, this could result in a new item appearing in the person's agenda view window. The agent is then responsible for interpreting the item and performing the appropriate task.

Agents may also monitor items individually; this gives them the ability to post an item to an agenda and to observe the item so they can react to changes in the item's status. By examining the state of an agenda item corresponding to a step of the process program, the interpreter can execute the process. When a new step is to be executed, the interpreter identifies the appropriate execution agent creates an appropriately typed agenda item for that step, and posts it on the agent's agenda. As the agent executes the step, its updated status is reflected in the agenda item's status attribute value, which is monitored by the interpreter. As the status changes, the interpreter accordingly creates and posts sub steps, returns output parameters on successful completion of the step, propagates exceptions on unsuccessful completion, and so on. Thus, an AMS provides language-independent facilities that allow coordination to take place, while the interpreter encodes key coordination semantics.

Conclusions

Knowledge discovery research is developing and exploiting a diverse and expanding set of data manipulation and analysis techniques. Not all analysts, or even all organizations, can have a thorough knowledge of how to correctly and effectively combine and deploy these techniques. Process programming provides an effective means for specifying the coordinated use of KDD techniques by agents in potentially complex KDD processes. KDD applications produce reliable and repeatable results, which is necessary for the effective use of data mining across a wide range of organizations.

References

1. KDD 2006 workshop on Theory and Practice of Temporal Data Mining (TPTDM 2006) ,Tao Li,Changshing Perng.
2. Data Mining: Concepts and Techniques. J. Han and M. Kamber. Morgan Kaufmann, 2000.
3. Diggle, Peter J., "Spatial Analysis of Spatial Point Patterns", 2nd Edition, Arnold Publishers, 2003.
4. Domanski, B., A Neural Net Primer, Journal of Computer Resource Management, Issue 100, Fall 2000.
5. Gianluca Moro, Sonia Bergamaschi, Karl Aberer Agents and Peer-to-Peer ComputingThird International Workshop, AP2PC 2004, New York, NY, USA, July 19, 2004.
6. D. Isern, D.Sanchez, A.Moreno, "Organizational structures supported by agent-oriented methodologies", The journal of Systems and Software, vol. 84, n. 2, Oxford, UK: Elsevier, 2011, pp. 169-184. [2]

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

7. F. Bellifemine, G. Cairo, and D. Greenwood. Developing Multi-Agent Systems with Jade. Wiley Series in Agent Technology, ISBN: 9780470057476, 2007.
8. R. H. Bordini, L. Braubach, M. Dastani, A. E. F. Seghrouchni, J. J. G.Sanz, G. OHare J. Leite, A. Pokahr, and A. Ricci. A survey of programminglanguages and platforms for multi-agent systems. In Proceedings ofthe IEEE International Conference on Cognitive Informatics, 2006.
9. L. Cao, C. Luo, and C. Zhang. Agent-Mining Interaction: An Emerging Area. AIS-ADM07, LNAI 4476, Springer - Verlag, Berlin, Germany, 2007.
10. K. A. Albashiri, F. P. Coenen, and P. Leng. Agent Based Frequent Set Meta Mining: Introducing EMADS. Artificial Intelligence in Theory and Practice II, IFIP'2008, Springer, London, UK, 2008.