Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp202-207)

Hybrid Assistance in KDD Task Definition

RONALDO GOLDSCHMIDT[1,2], EMMANUEL PASSOS[3], MARLEY VELLASCO[3]
[1]DE9: Departamento de Engenharia de Sistemas, IME
[2]NUPAC: Núcleo de Projetos e Pesquisas em Aplicações Computacionais, UniverCidade
[3]ICA: Laboratório de Inteligência Computacional Aplicada, PUC-Rio

Abstract: - Although some research has been dedicated to the development of Knowledge Discovery in Databases (KDD) assistance mechanisms, little effort has been directed to the deployment of tools that assist humans during the KDD task definition stage. In order to satisfy this need for a KDD task definition assistance device, the present work proposes three different approaches: a) the first one is called theoretical approach and is based on concepts from the Theory of Attribute Equivalence in Databases [3] and from Topological Spaces [4]; b) the second employs Artificial Neural Networks [7] to learn mappings between heterogeneous patterns and is called experimental approach; c) the third one combines the abovementioned approaches to implement what is called hybrid approach. These approaches, their models and implementations are described in detail. Experiments with real KDD applications, comparisons and conclusions are reported.

*Keywords***:** Knowledge Discovery in Databases, Data Mining, Artificial Intelligence, Assistance in KDD

1. Introduction

One of the most important stages in KDD (Knowledge Discovery in Databases) applications is called *Task Definition* [9]. Classification, Regression, Clustering, Summarization, Deviation Detection, Time Series Forecasting, Associative Rules and many others are examples of KDD tasks.

An analysis of the specialized literature reveals that few attempts have been made to develop computational mechanisms that assist humans in defining tasks in KDD applications [9]. In METAL [2] and NOEMON [6] projects, the authors have proposed tools to help users to rank and choose classification algorithms based on its past performance. Additionally, once defined a data mining algorithm, IDEA [1] and MiningMart [5] mechanisms present data pre-processing alternatives. It is important to emphasize that all these works demand previous task definition by humans and could be integrated to the approaches currently described in this paper.

Thus, this paper proposes a computational model (called *KDD Task Definition Assistance Mechanism*) whose purpose is to assist humans during the *Task Definition* stage by suggesting alternative KDD tasks for each application. We argue that such mechanism can be a useful tool in, at least, one situation which has motivated this work: we have taught several KDD and Data Mining courses for different people. Students usually do not know to start solving their exercises. They often do not know how to identify possible KDD tasks in a given database. The proposed mechanism can help students overcome this first obstacle by presenting alternatives of KDD feasible tasks. So students can investigate one, some or all presented alternatives. Such mechanism owes its inspiration to the observation of an intentional-level type of similarity that certain databases present between themselves. The intentional level of a database regards the structure or the schema of that database [3]. The observation of this fact is helpful when one is identifying the type of knowledge to be discovered, since data sets with similar structures tend to arouse similar interests even in distinct KDD applications. Three approaches were proposed to implement heterogeneous pattern mapping between databases' structures and viable KDD tasks: a) theoretical – where knowledge is defined by human experts; b) experimental – where knowledge is learned by Artificial Neural Nets; c) hybrid – a combination of (a) and (b). Concepts from the *Theory of Attribute Equivalence in Databases* [3], from *Topological Spaces* [4] and from Neural Networks' Theory [7] were employed in the formalization of the proposed approaches.

This paper is organized as follows: Section 2 formalizes the proposed computational model. It

presents the basic principles used and specifies the adopted functional composition. It also describes the three heterogeneous pattern mapping approaches. Section 3 contains descriptions of the experiments that were carried out and an analysis of the obtained results. In Section 4, a few conclusions and perspectives for further studies are set forth.

2. Proposed Computational Model

2.1. Basic Principles

In [3], authors have described what is known as the *Theory of Attribute Equivalence in Databases* and its application in the integration of database schema. This theory is based on the use of database attribute characteristics, called *metadata*, for the purpose of determining the existence of some type of equivalence between pairs of attributes. The relation called DOMAIN_DISJOINT_ROLE_EQUAL occurs when the *domains* of the attributes are *disjoint* but their *roles* are identical. The *role* played by an attribute within a context may be defined as form mapping [3]:

Role: Attribute X Context → Role Name

This type of mapping aims to express the relation that exists between the attribute and the context in which it is inserted. However, it is important to point out that the DOMAIN_DISJOINT_ROLE_EQUAL relation is not an equivalence relation since the reflexive and transitive properties are not satisfied.

The approaches that have been adopted in this paper start out by attempting to characterize the similarity between attributes in relation to the functions they perform in their respective databases. Thus, the concept of attribute role was specialized in a manner such that the context considered is the database structure itself:

Role: Attribute X Database Structure → Role Name

Based on the description above, a relation R is proposed: Given any two attributes A_1, A_2 , $(A_1, A_2) \in R$ if, and only if A_1 and A_2 have the same role in the structures of their own respective databases. It can easily be verified that R is an equivalence relation. Such being the case, it is also proposed that attributes that perform the same role in the data sets to which they belong be regarded as being *functionally equivalent* and that they be classified within a same *functional class*.

The present approaches also propose the use of metadata on each attribute of a data set, and that such metadata be used to define the role of each attribute within the structure of its set. Metadata may be used for the functional classification of database attributes [9].

As an extension of the concepts that have been put forth, this paper also proposes the following definition for structural resemblance between databases:
Definition: Two databases (data sets) S and S' have structural resemblance between themselves if, and only if the set of functional classes of the attributes of S coincides with the set of functional classes of the attributes of S'.

The usefulness of this definition may be understood when it is noticed that the data sets that have structural resemblance between themselves have the potential for performing the same tasks in KDD applications.

2.2. Functional Composition

For the purpose of providing a systemic view of the *KDD Task Definition Assistance Mechanism*, figure 1 illustrates the functional components of this device.

Let $S(A_1, A_2,..., A_n)$ be a data set that has been presented to the *KDD Task Definition Assistance Mechanism*. Each attribute A_i, $i=1,..., n$ is submitted to the functional classification process. This process receives the values of the metadata of A_i and then processes the functional classification knowledge base, thereby defining the functional class to which each A_i belongs.

The functional classes, the metadata and the functional classification knowledge base rules should reflect the experience of a KDD specialist and are to have been specified beforehand [9].

Once the functional classes have been defined for all A_i, $i=1,..., n$, the assistance mechanism represents S as a set of functional classes C_S. Potentially, C_S may be any element of $P(C)$, the power set of the complete set of functional classes C.

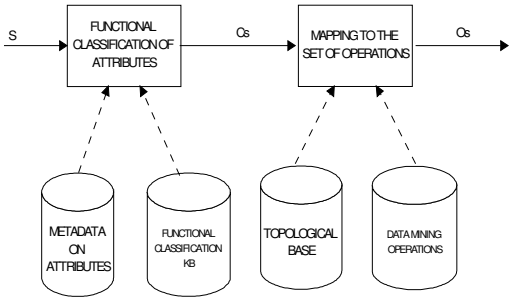

Fig. 1. Systemic View of the *KDD Task*

The mapping of C_S into the set of feasible data mining operations, O_S, is an important process in KDD task definition. Section 2.3 describes three alternatives for the implementation of this process. Within the scope of this paper, the expressions *"data mining operation"* and "*KDD task"* are regarded as synonyms and will be employed interchangeably throughout the text.

2.3. Mapping between Functional Classes and Data Mining Operations

2.3.1. Theoretical Approach

In the theoretical approach, the mapping process receives the set of functional classes C_S generated from the data set S and is subdivided into two stages: (a) Representation of C_S as a combination of previously defined patterns; (b) Mapping of the obtained combination into a set of data mining operations O_S. To this end, this approach involves the abstract treatment of similarity (*distance*) between sets whose elements are functional classes.

Thus, the present approach was formalized with the use of concepts of *Topological Spaces* [7], which generalize the concepts related to distance of *Metric Spaces*.

Let S be a data set, C the set of all the functional classes, C_S the set formed by the functional classes of the attributes of S, and O the set of all data mining operations. The mapping process consists of obtaining O_S, $O_S \subseteq O$, the subset of data mining operations associated with C_S. $(C,P(C))$ and $(O,P(O))$ are topological spaces [9].

Assuming that there is a topological base on C that contains the canonical base on C [14]: $\beta = \{B_1, B_2, ..., B_k\}$ and that there exists a partitioning of the elements of β into two subsets G and G' ($G \cup G' = \beta$ and $G \cap G' = \varnothing$), such that: (a) G contains elements of the base that have enough information to define feasible data mining operations; (b) On the other hand, G' contains elements of the base that do not have enough information to define feasible data mining operations. Also assuming that there exists a previously defined function $t: \beta \rightarrow P(O)$ for all the elements of the topological base on C. The following mapping T is proposed $T: \beta \times \beta \rightarrow P(O)$, such that:

$$T(B_i \cup B_j) = \begin{cases} t(B_i) \cup t(B_j), & \text{if } B_i \text{ and } B_j \text{ belong simultaneously to } G \text{ or } G' \\ t(B_i) \cap t(B_j), & \text{if not.} \end{cases}$$

Since $C_S \in P(C)$, one has that: $C_S = \cup B_k$. Thus, $T(C_S) = T(\cup B_k)$ [9]. It is proposed that $T(C_S)$ be used for defining a set of data mining operations that are potentially performable in S, that is, $T(C_S) = O_S$.

Since the representation of C_S as the union of the elements of base, $\cup B_k$, is not unique, a procedure that will ensure the uniqueness of this representation still remains to be defined. This procedure is described in [9].

Once it has been expressed as the union of the elements of the base ($\cup B_k$), C_S may then be mapped into the set of data mining operations by the function T defined above.

2.3.2. Experimental Approach

In this approach, the mapping between functional classes and data mining operations must be learned from previously processed databases. The notorious Back-Propagation Neural Network Model's good performance in learning mappings between heterogeneous patterns strongly influenced the choice of such model to implement the experimental approach.

For the training set, there were selected only databases where KDD processes had been executed. Each database was then represented by two binary patterns: a) the first one was built from database's functional classes (C_S). In each position, the pattern indicated presence (1) or absence (0) of the corresponding functional

class; b) the second pattern was generated from database's executed data mining operations. In an analogous way, for each position, the pattern indicated presence (1) or absence (0) of the corresponding data mining operation.

Once the neural network had been trained, the idea in this approach was to present new databases, represented by their functional classes, to such net and consider its indicated data mining operations as viable ones.

2.3.3. Hybrid Approach

This approach combines the previous ones. The idea was to obtain better results, even under situations where one of the other approaches did not show good performance.

In essence, the hybrid approach considers the union of the data mining operations suggested by the other approaches. Using union, the hybrid approach takes into consideration data mining operations suggested by at least one of the others approaches, thus enhancing the set of identified alternatives.

3. Experiments

3.1. Testing Methodology

For the purpose of evaluating the implementation of the proposed computational model, a study was carried out with a view to collecting the opinions of KDD analysts. These analysts were asked to identify feasible KDD tasks in twenty different situations. Each situation was characterized by the name of a data set and by a brief contextual description of the attributes involved.

The data sets were selected in such a way as to contain diversified examples of real KDD applications and examples of databases that have been used in several studies performed by the scientific community [9].

Basically, the criterion for selecting the KDD analysts was their availability and interest in participating in the process. Thirty analysts were consulted and separated into three knowledge levels: experienced, intermediate, and beginner. Ten analysts were classified in each level. Students were classified in beginner level. Intermediate level contained analysts that have worked in KDD for three years. Analysts with more than three years working with KDD were classified in experienced level.

For each situation presented in the study, the interviewees were asked to mark the KDD tasks that they considered feasible in each database.

The same twenty data sets were presented to the three approaches of *KDD Task Definition Assistance Mechanism* (theoretical, experimental and hybrid) that, based on the models shown in the previous section, indicated sets of data mining operations. With the answers provided by the KDD analysts and by the assistance mechanism, some comparative measures were calculated for each situation. These measures are described in Table 1. *Card(X)* indicates the number of elements that belong to set *X*. The E_k, *A, I* and *U* sets contain, respectively: a) the data mining operations suggested by the *k-th* KDD analyst; b) the data mining operations suggested by the assistance mechanism; c) the common data mining operations suggested by all the KDD analysts: $I=\cap E_j$. The objective of set *I* is to represent the common sense presented by the analysts that were consulted; d) the data mining operations suggested by at least one of the analysts: $U=\cup E_j$.

Table 1. Comparative Measures

$\frac{Card(E_j - A)}{Card(E_j)}$	Proportion of operations suggested by the *j-th* KDD analyst and that were not identified by the Assistant.
$\frac{Card(A-E_j)}{Card(A)}$	Proportion of operations suggested by the Assistant and that were not identified by the *j-th* KDD analyst.
$\frac{Card(I - A)}{Card(I)}$	Proportion of common operations suggested by a group of analyst and that were not identified by the Assistant.
$\frac{Card(A - I)}{Card(A)}$	Proportion of operations suggested by the Assistant and were not identified by the analysts in a specific group. *The closer this measure is to one, the better can be the assistance provided by the proposed mechanism.*
$\frac{Card(A - U)}{Card(A)}$	Proportion of operations suggested by the Assistant and that were not identified by any of the analysts in a group.

The theoretical approach based assistance mechanism was configured by experienced KDD analysts who had not participated in the

mentioned study. Similarly, the experimental approach based assistance mechanism was trained with sixty databases different from the ones used in the mentioned study. In all experiments, the back-propagation training parameters *learning rate*, *momentum*, *number of epochs* and *error tolerance* were set to *0.45*; *0.75*; *15.000* and *0.05*, respectively. Due to the limited number of patterns (sixty databases), network's topology was *12 (Card(C)) – 2 – 23 (Card(O))* in all experiments.

3.2. Results

Table 2 consolidates the obtained results based on the answers presented by the KDD analysts and by the proposed assistance mechanism (Theoretical, Experimental and Hybrid approaches). It presents the average measures for all twenty data sets. The results have been summarized in three groups. Each group represents one level of KDD analysts. The answers provided by each group were compared to the answers supplied by the assistance mechanism.

Table 2. Comparison of answers: Analysts Vs. Assistance Mechanism

measures	approach	Analyst Group		
		Group I Beginners	Group II Intermediate	Group III Experienced
$\frac{Card(\cap Ej\text{-}A)}{Card(\cap Ej)}$	T	0.07	0.15	0.11
	E	0.19	0.12	0.10
	H	0.17	0.10	0.08
$\frac{Card(A\text{-}\cap Ej)}{Card(A)}$	T	0.45	0.40	0.30
	E	0.63	0.52	0.32
	H	0.63	0.52	0.32
$\frac{Card(A\text{-}\cup Ej)}{Card(A)}$	T	0.34	0.17	0.00
	E	0.39	0.19	0.00
	H	0.40	0.20	0.00

The following considerations can be observed within each mapping approach:

- As was expected, the proposed mechanism does not exhaust the universe of KDD tasks that may be formulated by human beings. Nevertheless, it may be observed that the less experienced and knowledgeable the group of analysts, the greater is the quantity of data mining operations that are identified by the assistance mechanism, but are not perceived by the analysts ($Card(A\text{-}\cap E_j)/Card(A)$). It shows how useful such assistance could be if used in KDD teaching courses or even in real applications carried out by less experienced analysts. The assistant presents options of KDD tasks not perceived by analysts, indicating potential directions that could be followed within KDD applications.

- It is also important to point out that in *every* situation, *each one* of the tasks that were proposed by the assistant was validated by *at least one* analyst from Group III ($Card(A\text{-}\cup E_j)/Card(A) = 0$). This indicates that not only the assistance mechanism, but also the knowledge incorporated into it, showed good agreement with the KDD task definition process. Such measurement validates knowledge incorporated into the assistant mechanism because experienced analysts agreed with KDD tasks suggested by the assistant in all situations.

- In relation to the commonsensical measurements, it may be noticed that there were several data mining operations that were not perceived by all the analysts in each group ($Card(A\text{-}\cap E_j)/Card(A) \neq 0$). It may also be observed that, in certain situations, all the analysts in Groups I and II failed to indicate some viable data mining operations ($Card(A\text{-}\cup E_j)/Card(A) \neq 0$). It illustrates that even for the most experienced analysts, the proposed assistance mechanism may represent a useful tool that helps to reduce the possibility of forgetting feasible and interesting KDD task alternatives.

A comparative analysis of the three mapping approaches shows that the hybrid approach outperforms the other two. This fact can be verified by observing the average measures under each approach. As the hybrid approach considers the union of operations identified by the other two approaches, its set of data mining options is always more complete, or, in the

worst case, equal to the others. Therefore, every measure in hybrid approach is always better or equal to its corresponding in the other approaches.

4. Conclusions and Future Work

The purpose of this paper was to propose a computational model that could assist humans in defining which tasks should be performed in KDD applications. Unlike related works [9], the present approach does not use previous cases in order to suggest tasks for a new application. Concepts from the *Theory of Attribute Equivalence in Databases* [3], from *Topological Spaces* [4] and from Neural Networks [7] were employed in the formalization of the proposed model. Details about the model and its implementation were presented. A methodology for evaluating the results was proposed and applied. The analysis of the tests that were performed and of the obtained results illustrates the potentialities of the proposed model. It must also be mentioned that the proposed *KDD Task Definition Assistant Mechanism* has proven to be a useful practical tool in KDD and Data Mining courses taught by the authors. It has helped students from both scientific and industrial communities to learn how to identify potential KDD tasks in new applications.

A device for inducing the topological base and the other elements that are necessary for the execution of the theoretical approach is currently being developed. This device is a desirable tool mainly due to two reasons: (a) it would help KDD specialists to configure the knowledge used by the assistance mechanism; (b) it is not common to find out KDD specialists that are familiar with the abovementioned concepts of Topological Spaces.

Another work that is currently being developed by the authors involves the design of a planning assistant that helps human analysts in selecting algorithms to perform KDD tasks. The approaches used in METAL [2], NOEMON [6], IDEA [1], MiningMart [5] and IKDD [8] projects have provided important insights to such work. This assistant is supposed to be integrated to the mechanism described in this paper, providing a useful and more complete assistance in KDD applications.

Acknowledgements:
This work was supported by FAPERJ, Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro, Brazil.

References:

[1] Bernstein, A., Provost, F., An Intelligent Assistant for the Knowledge Discovery Process, *IJCAI-01 Workshop on Wrappers for Performance Enhancement in KDD*, (2001).

[2] Brazdil P., Soares C., A Comparison of Ranking Methods for Classification Algorithm Selection, *Proceedings of the 11th European Conference on Machine Learning*, (2000).

[3] Larson, J. A., Navathe, S. B., Elmasri, R., A Theory of Attribute Equivalence in Databases with Application to Schema Integration, *IEEE Transactions on Software Engineering*, Vol. 15, No. 4, April (1989).

[4] Lipschultz, S., Topologia Geral, *Coleção Schaum*, McGraw-Hill, São Paulo (1979).

[5] Morik, K., The Representation Race - Preprocessing for Handling Time Phenomena, *Proceedings of the European Conference on Machine Learning 2000 (ECML 2000)*, LNAI Vol. 1810, Springer Verlag, Berlin (2000).

[6] Spiliopoulou, M., Kalousis, A., Faulstich, L., Theoharis, T., NOEMON: An Intelligent Assistant for Classifier Selection. *In FGML98*, 98/11, Dept. of Computer Science, TU Berlin, pp. 90-97, (1998).

[7] Haykin, S. Neural Networks: a Comprehensive Foundation. Prentice Hall, (1999).

[8] Goldschmidt, R., Passos E., Godoy, R., Paolino, A., Schettini, F. *Assistance in Selecting KDD Algorithms*, International Conf. on Computing, Communications and Control Technologies, Texas, USA, (2004).

[9] Goldschmidt, R., Passos E., Vellasco, M., *Task Definition Assistance in KDD Applications*, in Proc. of XXIX Conferência Latino Americana em Informática, Bolívia, (2003).