Flexible Support for Knowledge Discovery

JÁN PARALIČ AND MAREK HUBAL

ABSTRACT: This paper presents basic ideas and results of the GOAL project (International Copernicus research project No. 977091 Geographic Information On-Line Analysis (GIS - Data Warehouse Integration) supported by the European Commission) focusing on its knowledge discovery part. Within this project a KDD (Knowledge Discovery in Databases) package [4] has been designed and implemented. In this paper motivation, architecture, functionality and one two of the implemented DM (data mining) modules of the KDD Package are described in greater detail. KDD Package supports the whole KDD process [8] starting with possibility to connect to various data sources, following with support for data preprocessing, data mining and knowledge visualization. Based on the GOAL pilot applications one of the DM tasks that are supported now, is the prediction DM task (by means of case-based reasoning approach optimizing weights using evolutionary algorithm [2]). KDD Package is designed as an open system with easy integration of new DM, data preprocessing or visualization modules.

KEY WORDS: knowledge discovery in databases, data mining

1 Introduction

The integration and combination of GIS (Geographic Information System) data into and with OLAP (On-Line Analytical Processing) systems poses a number of yet not satisfyingly solved problems in terms of getting the data into the OLAP system [7], representing the data for analysis and extracting knowledge while considering security restrictions. Current approaches do not address these special problems resulting from the targeted application arena of GIS and OLAP systems.

The main objective of the GOAL project is to develop a generic framework both recognized by the research community and applicable in real world applications, which solves the general issues of GIS and DWH (Data Warehouse) interoperability [5], including DWH feeding [6], knowledge extraction, interpretation, and security concepts. The feasibility of this framework is being tested on 2 very different real world applications from the GIS domain using environmental sensor data of a water supply company and cultural data about historical monument visitors [11], allowing a real world evaluation of the framework.

Within this project a KDD (Knowledge Discovery in Databases) package [4] has been designed and implemented at the Department of Cybernetics and Artificial Intelligence, University of Technology in Kosice.
In this paper the KDD Package is presented in greater detail. Starting with motivation for development of such a system in the context of the GOAL project mentioned above (Sect. 2) there is further presented architecture of the KDD Package in Section 3. Sections 4 and 5 provide a detailed view of the most interesting part of the system, namely DM modules supporting prediction task. Finally, Section 6 concludes the paper with a summary of the main ideas presented here.

2 Motivation

Techniques provided by the OLAP systems enable to analyze data from data warehouses in a quite easy way, but they are lacking algorithms for more sophisticated analysis of data provided by the KDD approach. Therefore one of the three major objectives of the GOAL project is to develop a KDD supporting tool which shall be integrated with the generic framework of the GIS - DWS integration system [6]. Moreover, DM algorithms implemented within the KDD Package prototype should effectively support two real pilot applications mentioned in Sect. 1.

In the following KDD process will be briefly introduced as a basic reference for KDD Package design objectives presented afterwards.

2.1 KDD Process

Knowledge discovery in databases (KDD) can be defined as nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. According to [1] it is an interactive and iterative process with several steps. It means that at any stage the user should have possibility to make changes (for instance to choose different algorithm settings, different DM task or preprocess data in another way) and repeat the following steps to achieve better results. Data mining is a part of this process.

In most of sources, the term Data Mining (DM) is often used to name the field of knowledge discovery. This confusing use of terms KDD and DM is due to historical reasons and due to fact that the most of the work is focused on refinement and applicability experiments of machine learning algorithms from artificial intelligence for the data-mining step. Pre-processing is often included in this step as a part of mining algorithm.

Within the KDD process following steps can be recognized [2].

1. Data cleaning to remove noise and inconsistent data
2. Data integration, where multiple data sources may be combined
3. Data selection, where data relevant to the analysis task are retrieved from database (or data warehouse, where data is already cleaned and integrated)
4. Data transformation - data are transformed or consolidated into forms appropriate for mining
5. Data mining [12] as core of the KDD process, where intelligent methods are applied in order to extract data patterns
6. Pattern evaluation - to identify interesting patterns
2.2 Main design objectives of the KDD Package

Software that aims effectively support the KDD process cannot focus just on one of the above-mentioned steps but should cover ideally all of them. Therefore in order to adopt existing data mining algorithms to be used in connection with real data sources (in a database or in a data warehouse), four crucial objectives have been identified for KDD Package design.

1. Fast connection to existing data sources. In our not only databases, but also data warehouses must be taken into account.

2. Flexible and rich set of data preprocessing methods (which involves steps 1 to 4 in KDD process presented in Sect. 2.1 above) must be provided (in form, which is easy to use and easy to understand by the user).

3. The system must be open for easy integration of new data mining algorithms (step 5).

4. Effective support of evaluation of discovered patterns must be provided by means of (preferably) visual knowledge representation (steps 6 and 7).

3 Architecture

Based on objectives presented in Sect. 2.2 above, KDD package has been design [10] and implemented in such a way, that it can be used not only within the GOAL generic framework, but also as a stand-alone application supporting the whole KDD process.

KDD Package has modular structure (see Figure 1), where common parts of the system can be used by each of the specialized DM modules.

Data access module serves for accessing database sources of various types (which can be a text file, DBase, Paradox or MS Excel table, any local or remote SQL database using a visual SQL wizard or data warehouse through MS Excel interface). Data is connected (basic statistical measures are provided at the same time. If user decides to process actually connected data, it will be loaded) and forms a so called view, which can be previewed in three levels of granularity - list of operations performed on actual view, list of attributes with statistics (so called quick view) or a two dimensional table of data (see Figure 2).

Module for data pre-processing enables a user to visualize, browse, modify, transform, sample etc. connected data. Pre-processing techniques are divided into those, which operate on rows, and those, which operate on columns of a view (see Figure 2).
All possible operations on data are defined by means of plug-in modules. This makes it possible to add a new transformation (sampling, discretization, etc.) operation on data any time very easily.

For each KDD task a different DM algorithm as well as knowledge visualization component is suitable. Therefore each new DM algorithm and its respective knowledge visualization component can be implemented separately and added into the KDD package in form of a separate plug-in module. It does not necessarily mean that each DM algorithm must have its own knowledge visualization component.

Usually in order to add a new DM task functionality into the KDD Package the following steps need to be done.

1. Implementation (or just re-use of an existing implementation) of a data-mining algorithm in form of a plug-in module

2. If necessary, implementation of new transformation or other pre-processing functions in form of plug-in module

3. If necessary, implementation of a new knowledge visualization component in form of a plug-in module
Process of adding a new plug-in module may be controlled directly from the running KDD package application.

Based on character of the real data from the GOAL project pilot applications, classification and prediction DM tasks functionality have been implemented by means of various algorithms and their combination (see Sect. 4 and 5). KDD Package with its DM modules is being tested on the real data from two pilot applications as well as on the data from UCI repository.

4 Prediction

This DM task will be used in other pilot application (prediction of water consumption based on environmental sensor data of a water supply company). The following algorithms have been implemented.

1. Linear regression

2. MP5’ producing model or regression trees [12], see also Figure 3.

3. Case-based reasoning (CBR) approach [9], where weights of the nearest cases taken into account for calculation of predicted value are optimized by means of evolutionary algorithm.

In the following our CBR approach will be briefly presented. A new problem is solved in CBR by finding a similar past case, and reusing it in the new problem situation. In case of prediction we use $k$ nearest past cases to predict the target attribute of a new case. In order to find the $k$ nearest cases, we use the following distance measure between two cases $A$ and $B$: 
\[ D_{A,B} = \sum_{i=1}^{n} W_{A_i} |atr_A^i - atr_B^i| \]  

(1)

where \( A \) is a past case, \( B \) a new case, \( W_{A_i} \) is weight of attribute \( A \), \( atr_A^i \) is the value of \( i \)-th attribute of the past case \( A \), \( atr_B^i \) is the value of \( i \)-th attribute of the new case \( B \) and \( n \) is the number attributes (without the target attribute to be predicted). The attributes are normalized before calculating distances. If any of the attribute values is unknown, value of the distance is set to 1. For nominal attributes if both attribute values are similar, their distance is 0, otherwise the distance is set to 1.

Value of the target attribute of a new case \( B \) is calculated as a weighted sum of corrected values of target attribute of all \( k \) nearest past cases.

\[ C = \sum_{i=1}^{k} W_{N_i} Corr_i C_i \]  

(2)

where \( W_{N_i} \) is the weight of the \( i \)-th closest neighbor case, \( C_i \) is its value of target attribute, \( Corr_i \) is correction used for the \( i \)-th closest neighbor case and \( k \) is the number of neighbor cases taken into account for prediction. Correction used for the \( i \)-th closest neighbor case is calculated as follows.

\[ Corr_i = 1 + \sum_{j=1}^{n} 1 - \frac{atr_A^j atr_B^j}{W_{C_i}} \]  

(3)

where \( W_{C_i} \) is weight of the \( i \)-th attribute after correction.

For set of all training cases minimal, maximal and average error is calculated. Quality of prediction is given by used weights (\( W_{A_i} \), \( W_{N_i} \), \( W_{C_i} \)) and number of past neighbor cases \( (k) \) taken into account for prediction. User may set up all these parameters. Weights are further optimized in order to achieve minimal prediction error by means of evolutionary algorithm.

Each individual in population is represented as a one-dimensional vector of real values of all weights \( W_{A_i}, W_{N_i}, W_{C_i} \) for \( i = 1, \ldots, n \). There is possible to choose between mean deviation and mean squared error for fitness function that defines quality of each individual.

The user gives population size and may also specify initial weights, which will be used for the first (best) individual in initial generation. Termination condition can be specified by means of maximal error (minimal fitness) of the best individual, by means of the number of generations, or both.

Each new generation is calculated using two genetic operators - mutation and crossover. There are two types of mutation, one that should produce individuals significantly different from the parents (random change of weights from predefined intervals) and the other one mainly aimed for fine changes in individuals (random change of weights in a very small interval). Crossover means that less then half of the weights are changed between two parent individuals. Particular genetic operators are applied to each parent individual with specific probabilities, which may be set up by the user.

New generation is selected from all individuals produced within a generation (all parent as well as all individuals newly generate using genetic operators) as follows. All individuals are sorted by their fitness and 80% of the best plus 20% of the worst is taken into the new generation of individuals.
5 Experiments

The goal of our experiments was to apply the CBR prediction algorithm to predict drinking water consumption.

At first, using SQL scripts we transformed data from data warehouse to a single table. For prediction data from one reservoir has been used. In the next step we summed up all measurements achieved in one day in order to obtain values of “one-day” attributes. As last step, we computed average values of attributes for some past days.

In this way data from time period since January to November 1999 and from January to June 2000 has been processed. Data from year 1999 has been divided into two sets and data from year 2000 has been used as testing set.

5.1 Common data description

Number of cases in case memory: 212

Number of cases in training set: 114

Number of cases in testing set: 182

Number of attributes: 26
Missing values: No

**Predicted attribute:** CONSUMPTION

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MONTH</td>
<td>Month of year. Values 1–11.</td>
</tr>
<tr>
<td>4. QUARTER</td>
<td>Quarter of the year. Values 1–4.</td>
</tr>
<tr>
<td>5. INFLOW</td>
<td>Water amount in m$^3$ added to tank during a day.</td>
</tr>
<tr>
<td>6. INFLOW_1</td>
<td>Water amount in m$^3$ added to tank during previous day.</td>
</tr>
<tr>
<td>7. INFLOW_2</td>
<td>Water amount in m$^3$ added to tank during previous two days.</td>
</tr>
<tr>
<td>8. OUTFLOW</td>
<td>Water amount in m$^3$ out flowed from tank during a day.</td>
</tr>
<tr>
<td>9. OUTFLOW_1</td>
<td>Water amount in m$^3$ out flowed from tank during previous day.</td>
</tr>
<tr>
<td>10. OUTFLOW_2</td>
<td>Water amount in m$^3$ out flowed from tank during previous two days.</td>
</tr>
<tr>
<td>11. BRIGHTNESS</td>
<td>Amount of sunshine in hours within a day.</td>
</tr>
<tr>
<td>12. BRIGH_1</td>
<td>Amount of sunshine in hours within previous day.</td>
</tr>
<tr>
<td>13. BRIGH_5</td>
<td>Amount of sunshine in hours within previous 5 days.</td>
</tr>
<tr>
<td>14. BRIGH_10</td>
<td>Amount of sunshine in hours within previous 10 days.</td>
</tr>
<tr>
<td>15. BRIGH_15</td>
<td>Amount of sunshine in hours within previous 15 days.</td>
</tr>
<tr>
<td>16. TEMPERATUR</td>
<td>Average temperature in °C during day.</td>
</tr>
<tr>
<td>17. TEMP_1</td>
<td>Average temperature in °C within previous day.</td>
</tr>
<tr>
<td>18. TEMP_5</td>
<td>Average temperature in °C within previous 5 days.</td>
</tr>
<tr>
<td>19. TEMP_10</td>
<td>Average temperature in °C within previous 10 days.</td>
</tr>
<tr>
<td>20. TEMP_15</td>
<td>Average temperature in °C within previous 15 days.</td>
</tr>
<tr>
<td>21. RAIN</td>
<td>Amount of rainfall in mm during a day.</td>
</tr>
<tr>
<td>22. RAIN_1</td>
<td>Amount of rainfall in mm during previous day.</td>
</tr>
<tr>
<td>23. RAIN_5</td>
<td>Amount of rainfall in mm during previous 5 days.</td>
</tr>
<tr>
<td>24. CONSUMTION</td>
<td>Water amount in m$^3$ stripped from tank by customers for a day. Predicted value.</td>
</tr>
<tr>
<td>25. CONS_1</td>
<td>Water amount in m$^3$ consumed from tank by customers during previous day.</td>
</tr>
<tr>
<td>26. CONS_2</td>
<td>Water amount in m$^3$ consumed from tank by customers during previous two days.</td>
</tr>
</tbody>
</table>

Table 1: Detailed description of all attributes provided for the water consumption prediction task.

The following attributes have been chosen in pre-selection at the beginning: MONTH, DAY, DAY_OF_WEEK, QUARTER, INFLOW_1, INFLOW_2, OUTFLOW_1, OUTFLOW_2, BRIGH_1, BRIGH_5, BRIGH_10, BRIGH_15, TEMP_1, TEMP_5, TEMP_10, TEMP_15, RAIN_1, RAIN_5, CONS_1, CONS_2.

Attribute CONSUMPTION was assigned as predicted attribute. For attribute normalization method that uses statistical variance has been used.

We have chosen attributes according to their correlation coefficients (MONTH, OUTFLOW_2, CONS_2, INFLOW_2). Results of our experiments (average relative error) are presented in Table 2 and Figure 5.
<table>
<thead>
<tr>
<th>NN</th>
<th>M1</th>
<th>M2</th>
<th>Average</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>22,5245</td>
<td>30,03224</td>
<td>26,27837</td>
<td>22,5245</td>
</tr>
<tr>
<td>4</td>
<td>21,80371</td>
<td>30,98994</td>
<td>26,39682</td>
<td>21,80371</td>
</tr>
<tr>
<td>5</td>
<td>13,16785</td>
<td>24,37796</td>
<td>18,7729</td>
<td>13,16785</td>
</tr>
<tr>
<td>6</td>
<td>14,61257</td>
<td>22,14871</td>
<td>18,38064</td>
<td>14,61257</td>
</tr>
<tr>
<td>7</td>
<td>43,1062</td>
<td>34,58964</td>
<td>38,84792</td>
<td>34,58964</td>
</tr>
</tbody>
</table>

NN - number of neighbors, M1 - First measurement, M2 - Second measurement

Table 2: Average relative error for given number of neighbors measured on testing set (used attributes MONTH, OUTFLOW_2, CONS_2, INFLOW_2).

Figure 5: Dependence of average relative error on number of neighbors (attributes MONTH, OUTFLOW_2, CONS_2, INFLOW_2 have been used).

Next, we further reduced the number of used attributes to OUTFLOW_2, INFLOW_2, TEMP_15. We removed attribute MONTH and replaced attribute of consumption CONS_2 with the most correlated one from the weather attributes - TEMP_15. The results achieved are presented in Table 3 and Figure 6.

The lowest prediction error that we obtained on this set of attributes was 10.86%, which is less than in previous case.

6 Summary

In this paper a tool supporting the whole KDD process called KDD Package has been presented. This tool has been designed and implemented within the GOAL project aiming both effectively support decision making in the GIS-DWS integrated framework as well as provide a stand alone application for any future KDD application.
<table>
<thead>
<tr>
<th>NN</th>
<th>M1</th>
<th>M2</th>
<th>Average</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12,51673</td>
<td>12,73552</td>
<td>12,62613</td>
<td>12,51673</td>
</tr>
<tr>
<td>4</td>
<td>12,62787</td>
<td>12,05693</td>
<td>12,3424</td>
<td>12,05693</td>
</tr>
<tr>
<td>5</td>
<td>11,05696</td>
<td>11,16432</td>
<td>11,11064</td>
<td>11,05696</td>
</tr>
<tr>
<td>6</td>
<td>10,85935</td>
<td>11,02842</td>
<td>10,94389</td>
<td>10,85935</td>
</tr>
</tbody>
</table>

NN - number of neighbors, M1 - First measurement, M2 - Second measurement

Table 3: Average relative error for given number of neighbors measured on testing set (used attributes OUTFLOW_2, INFLOW_2, TEMP_15).

Figure 6: Dependence of average relative error on number of neighbors (attributes OUTFLOW_2, INFLOW_2, TEMP_15 have been used).

KDD Package supports the whole KDD process starting with possibility to connect to various data sources like a text file, DBase, Paradox or MS Excel table, any local or remote SQL database or a data warehouse through MS Excel interface. Data pre-processing is supported by means of various row and column operations. Data mining and knowledge visualization provide core steps. Based on the GOAL pilot applications two DM tasks are supported.

Prediction DM task is supported by means of linear regression, M5' and case-based reasoning. All weights used in case-based reasoning approach (attributes weights, neighbors' weights and correction attribute weights) are optimized using evolutionary algorithm.

KDD Package is designed as an open system with the possibility to easy integrate any new DM task functionality, data preprocessing or knowledge visualization modules.
References


Ján Paralič
Department of Cybernetics and Artificial Intelligence
Technical University of Kosice
Letna 9, 04200 Kosice, Slovakia
e-mail: Jan.Paralic@tuke.sk

Marek Hubal
Department of Cybernetics and Artificial Intelligence
Technical University of Kosice
Letna 9, 04200 Kosice, Slovakia
e-mail: hubal@centrum.sk