A SIMPLE APPROACH FOR DATA MINING IN DELPHI

***Hewen Tang*[1*, 2] *and Yongsheng Cao*[1]**

[*1]*Institute of Crop Sciences, CAAS, Beijing, China 100081*
*Email:*Tanghw@cgris.net
[2]*Department of Computer Science of Yuxi Teacher's College, Yuxi, Yunnan Province, China 653100*

ABSTRACT

Data mining is a technique used to uncover previously unknown and potentially useful knowledge from large datasets. It turns data into actionable information for better decision making. Delphi incorporates a collection of high-performance data-analysis components known as the Decision Cube. The Decision Cube offers an easy path to powerful data analysis in an application. This paper describes a simple approach for data mining using the Decision Cube components of Delphi. It requires only some drag-and-dropping and property-setting, with no need for manual coding. It does not require special knowledge of data mining. At the end, an example is offered.

Keywords: Data mining, Delphi, Decision Cube, Drag-and-dropping, Property-setting

1 INTRODUCTION

With the development of society and enlargement of human activity, more and more data are produced. Simultaneously, progress in digital data acquisition and storage technology has resulted in the growth of huge databases (Hand et al., 2001). Databases today can range in size into terabytes - more than 10^{12} bytes of data. Hidden information of strategic importance lies within these masses of data (Two Crow's Corporation, 1999). However, when there are so many data and such huge databases, how can we get meaningful and useful knowledge from these large datasets? With ordinary tools or methods, it is difficult or even impossible. In this situation, data mining has been proposed, as a new technology and discipline providing a new and effective method to process data.

2 BACKGROUND

2.1 Data Mining

Data mining is a result of the evolution of databases and has become an important branch of database technology. Data mining, also popularly referred to as knowledge discovery in databases (KDD), is the automated or convenient extraction of patterns representing knowledge implicitly stored in large databases, data warehouses, and other massive information repositories (Han & Kamber, 2001). In short, data mining obtains useful and interesting information from very large datasets. It has many names: knowledge extraction, data archaeology, data dredging, and data/pattern analysis. Data mining has become increasingly common in many fields, such as banking, insurance, medicine, retailing, biology, and agriculture. In this paper we briefly describe data mining using a common data mining tool – Delphi.

2.2 Delphi

Delphi is an object-oriented, visual programming environment that develops applications for deployment on Windows and Linux (Borland, 2001). Delphi is usually considered to be a rapid application development (RAD)

tool. It has a very large visual component library (VCL), which provides a large number of components. This is very helpful for developing highly efficient applications with a minimum of manual coding.

Delphi holds many database components, such as Data Access, Data Controls, and Decision Cube, which make database-application development very easy and highly efficient. In this paper, we use only the Data Cube components. This group of components includes TDecisionCube, TDecisionQuery, TDecisionSource, TDecisionPivot, TDecisionGrid and TDecisionGraph, which introduce an easy way to provide powerful data analysis capabilities to an application, without having to write very much code.

3 PROPOSED APPROACH

We implemented a simple data mining task in Delphi using Decision Cube. This approach needed only drag-and-dropping and property-setting without any manual coding. The analysis process included two steps. First, we arranged all the components of Decision Cube in a form and set property values. Second, we set some properties values, as follows.

3.1 Select dataset source through TDecisionQuery

We selected the dataset source though the decision query Editor Box (Figure 2), reached the TDecisionQuery component. In this box, we selected the database and its table, chose the necessary fields as dimensions to be analyzed, and chose the calculation form (sum, average, or count). The selected fields and forms were displayed in “Dimension” box and “Summaries” box, respectively. After this setting, the SQL text was automatically formed and displayed in the SQL Query box (Figure 3). The text also could be edited through clicking “Edit Query” button.

3.2 Analyze dataset

We used the TDecisionCube component to receive a dataset from TDecisionQuery and analyze it, and then we transported the analyzed data to TDecisionGrid and TDecisionGraph through TDecisionSource.

After the necessary properties of TDecisionQuery were set, TDecisionCube and the Decision Cube Editor Box appeared (Figure 4). In this box, we set properties of the selected fields.

3.3 Visualize result

Visualizing the analysis results is the final process. We utilized the TDecisionSource component to transport result data to TDecisionPivot, TDecisionGraph, and TDecisionGrid by setting these components’ DecisionSource property as the name of their TDecisionSource component. We used TDecisionPivot to choosehow to display result data, TDecisionGrid to display the results in a table, TDecisionGraph to display result data in a graph. We chose the type of graph and change other graphing properties through the Editing DecisionGraph box (Figure 5), which appear from choices in TDecisionGraph component.

Finally, we set TDecisionQuery component’s Active property as true.

4 EXAMPLE

In this example, we chose the DBDEMOS database, affiliated with Delphi, as our data source. The task was to analyze the relationship between salary and hiredate.

4.1 Arrange components in a form

We placed components of Decision Cube in a form and arranged them by setting their “Align” properties. At the

same time, we set some dataset properties to connect the data sources among components. The changed properties are shown in Table 1, and values of the other properties are set by default. The final form is Figure 1.

Table 1. Value of property

control	property	value of property
TDecisionCube	DataSet	DecisionQuery1
TDecisionSource	DecesionCube	DecesionCube1
TDecisionPivot	Align	alTop
	DecisionSource	DecisionSource1
TDecisionGrid	Align	alTop
	DecisionSource	DecisionSource1
TDecisionGraph	Align	alTop
	DecisionSource	DecisionSource1

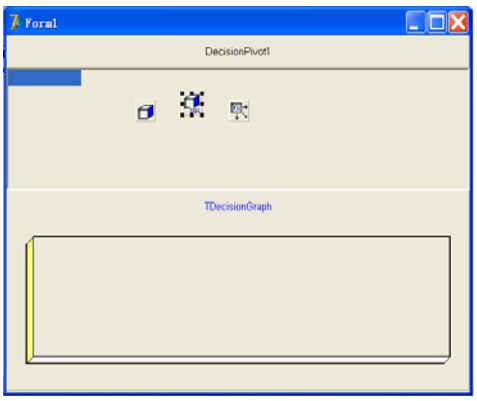

Figure 1. Components arrangement

4.2 Set value of important property in TDecisionQuery

In the Decision Query Editor Box, we selected DBDEMOS as the database and employee.db as the table. The selected fields and calculation form are shown in Figure 2. The SQL Query text was automatically formed as Figure 3.

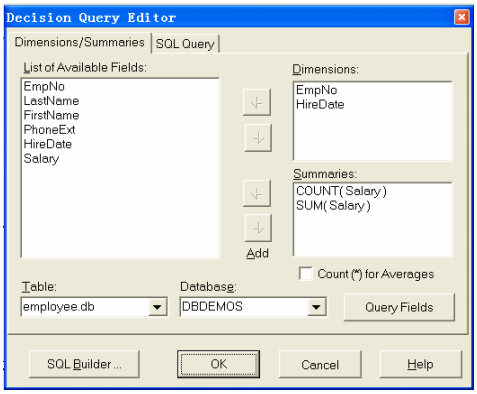

Figure 2. Decision Query Editor Box setting

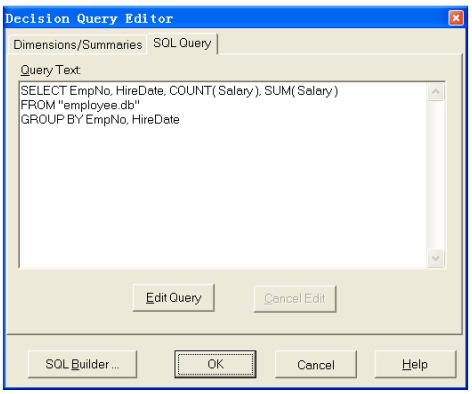

Figure 3. SQL Query text

4.3 Set value of property in TDecisionCube

We set selected fields' properties in the Decision Cube Editor Box (Figure 4). In this box, we could set the dimension's display name, type, and so on. In this example, we set all to default.

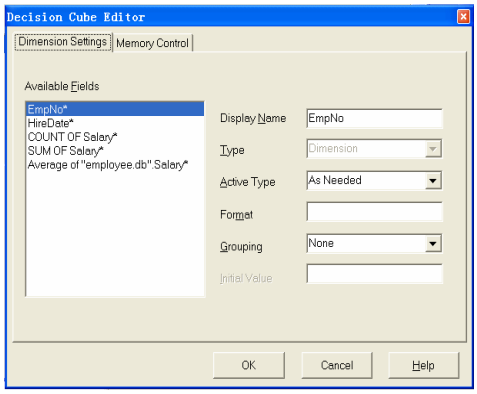

Figure 4. Decision Cube Editor Box

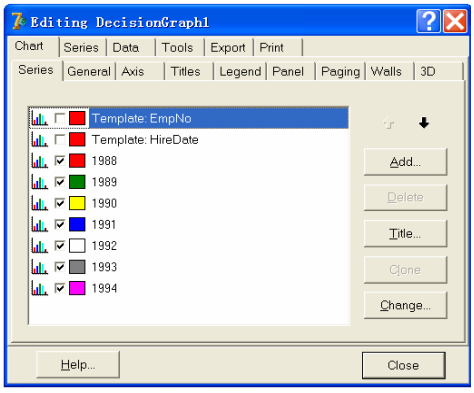

Figure 5. Editing DecisionGraph box

4.4 Set value of property in TDecisionGraph

We set graphing properties in the Editing DecisionGraph box (Figure5). Some values were also set as in Figure 5. Most values were set to default.

To change the graph type, the *change* button in Figure 5 can be used, and a new box displays as in Figure 6. In this example, we selected bar graph and canceled 3D effect.

4.5 Results

After finishing those settings, we set the TDecisionQuery's Active property to *true*. Then we ran the program and got the result shown in Figure 7.

From Figure 7, we see that the employee hired in 1988 receives the highest average salary. On the other hand, the average salary has no definite relationship with hiredate.

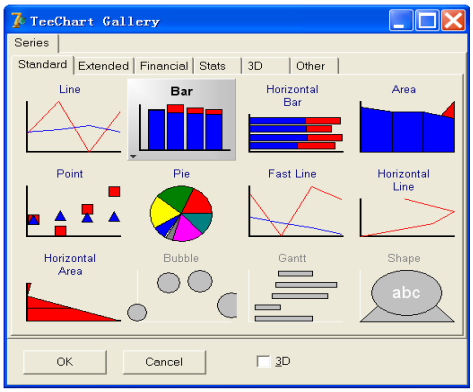

Figure 6. Chart Gallery

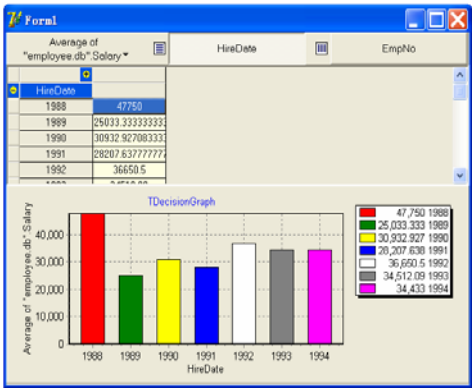

Figure 7. Graph of average of salary of hire date

5 DISCUSSION

This paper presents only a very simple example of data mining in Delphi. Actually, in Delphi, more complicated and efficient data mining applications can be developed. Through changing the values of properties, we can get more information and different kinds of graphs.

The data source can originate in various databases and Excel files through BDE or ODBC or ADO. Data mining results can also be represented in many forms, such as charts, formulas, and text.

6 REFERENCES

Borland Software Corporation (2001) *Delphi 7 Developers' Guide.* Borland Software Corporation

Han, J. & and Kamber, M. (2001) *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, Inc.

Hand, D., Mannila, H. & Smyth, P. (2001) *Principles of Data Mining*. The MIT Press.

Two Crows Corporation. (1999) *Introduction to Data Mining and Knowledge Discovery (Third Edition*). Two Crows Corporation.