

StatMine: An Interactive Statistical Data Mining System

Wen-Chi Hou¹, Yanliang Gu¹, Dunren Che^{1*},
Cheng Luo², Zhewei Jiang¹

¹Dept. of Compute Science Southern Illinois University, Carbondale, IL 62901 USA
{hou, dche }@cs.siu.edu

²Department of Mathematics and Computer Science, Coppin State University,
2500 West North Avenue, Baltimore MD 21216, USA

Abstract. In this research, an interactive approach to statistical data mining system is formulated and implemented. The system, called StatMine, provides a set of operators that can be used to examine data as well as extract various types of knowledge from databases. For efficiency, we build summary tables, which contain concise statistical information about groups of tuples, such that most statistical data mining tasks can be performed without scanning the databases. The knowledge exploration operators, coupled with the summary tables, facilitate the interactive search of different types of knowledge. Examples are provided to show how these operators can be used to explore data.

Keywords: statistical data mining, interactivity, summary table.

1 Introduction

Since its advent in early 1990's, there have been many data mining methods developed for various data mining tasks. Decision trees and rules, regression and classification methods, example-based methods, probabilistic graphical dependency model, and relational learning model are some of popular data mining methods. These data mining methods extract various kinds of knowledge,

* Corresponding Author. Email: dche@cs.siu.edu.

such as association rules, classificatory knowledge, sequential patterns, characteristic rules, etc., from databases [1, 2, 3, 5, 6, 7, 13, 15, 16, 20, etc.]. [11] contains a good review of these methods.

Interestness is an important criterion for data mining as patterns discovered are useful only if they are interesting to users [8]. This is a very user-dependent criterion. An appropriate way to address this problem is probably to let users involve themselves in the data mining process and search for patterns of interest interactively. There have already been some interactive data mining systems in the literature, for example, the IDEA system [18], IMACS system [4], DBMiner system [12], etc. In this research, we are to develop an interactive approach to mine for knowledge of interest.

Data analysis is often viewed as a synonym to data mining. While data mining has just been studied in the fields of database and artificial intelligence for a little more than a decade, data analysis has been researched by statisticians for decades. There is a wealth of statistical techniques available for mining or analyzing data. Indeed, statistical methods have been used in data analysis for a long time in various disciplines. Statistical approaches, with solid theoretical foundation and well-tested methods, are well suited for data mining tasks. In this research, we develop a framework to incorporate useful statistical techniques into a data mining system so that ordinary users without sophisticated background in statistics can also benefit from the wealth of statistical techniques.

In general, data mining is directly performed on raw data, incurring tremendous I/O's because potentially large databases may have to be scanned (sometimes more than once). In this research, we propose to build a summary table, which is a concise summary representation of the raw data, so that most data mining tasks can be accomplished using only the summary table, which makes the data mining process more efficient.

While there may be many types of knowledge, we shall focus on the most commonly referenced knowledge – summarization, association, classification, and clustering – in this research. To facilitate interactive and comprehensive data mining, we design an organized and systematic way to conduct data mining. We propose five basic operators: aggregate, compare, related, estimate, and cluster for data examination and knowledge extraction. These operators allow users to investigate data and perform various data mining tasks interactively. The operators provide users with flexibility and tools to conduct efficient data mining. Our approach is different from other interactive systems such as IDEA [18], IMACS [4], and DBminer [12]. IDEA focuses on keeping track of operations performed and maintaining relationships between them while IMACS on knowledge representation and segmenting data. They cannot perform sophisticated data mining tasks as DBminer and ours. DBminer lays its foundation on multi-dimensional databases while ours is based on statistical analysis. We design mining tools from statistical point of views while DBminer uses techniques developed by the data mining/database community. In this research, we provide convenient and powerful tools for examining data and statistical operators for exploring relationships.

The rest of the paper is organized as follows. In section 2, we describe the information to be stored in the summary table. In section 3, the five basic operators are defined. Section 4 shows

how to extract knowledge from summary tables using the operators. Section 5 is the conclusion.

2 Summary Tables

As the first step to designing a statistical data mining system, we attempt to identify the types of statistics that are needed for common data mining tasks. For statistical methods, characteristics of groups (of tuples) are often compared and hypotheses are tested before further investigation. Therefore, statistics used for such purposes form the basis of the information to keep in our summary table. Through careful reviews of statistical methods, we find that only the basic statistical variables, such as mean (M), standard deviation (δ), number of occurrences (N), and sum of product ($\sum XY$), where X and Y are two attributes of interest, need to be stored, while some other useful variables, such as sum of squares ($\sum X^2$) and covariance ($\sum (X - \bar{X})(Y - \bar{Y})$) can be computed using these basic variables, for example, $\sum X^2 = N(\delta^2 + M^2)$. Therefore, we shall only store such information in a table (relation), here we call a summary table.

Consider the relational schema Emp (Name, Dept, Title, Sex, Salary, Experience) as an example. There are generally two types of attributes in a relational model: numerical and categorical attributes. For example, Salary and Experience belong to the former while Dept, Title, Sex, and Rank the latter. In a summary table, tuples are grouped by categorical attributes, and summary information on numerical attributes is kept for each group of tuples. Table 1 is the sample summary table Emp_Sum constructed from Emp.

The summary table is very similar to the fact table in OLAP except that, instead of storing simple aggregate values, we store more sophisticated summary information (e.g., variance δ) and information about pairs of attributes of potential interests (e.g., sum of products of two attributes $\sum XY$). A summary table can also be visualized as a variant of the data cube with grouping attributes corresponding to the dimension attributes and numerical attributes to the measure attributes.

Table 1. A sample summary table

College	Dept	Title	Sex	Count	Avg_Salary	Dev_Salary	Avg_Exp	Dev_Exp	Sum_Exp_Salary
Engineering	ME	ASTP	F	5	53400	3400	3.2	2	856400
Engineering	ME	ASTP	M	8	54700	4500	4	2.3	1751000
Engineering	ME	PROF	F	4	73600	8400	17.6	7.3	5683000
Engineering	ME	PROF	M	8	74200	7500	15.8	5.7	9400000
Science	CS	ASTP	F	5	50800	2300	5.5	4.5	1388750
Science	CS	ASTP	M	6	52000	4200	6	3.2	1875000
:	:	:	:	:	:	:	:	:	:

We do not intend to store statistics for all possible combinations of pairs of numerical (or measure) attributes ($\sum XY$), as it could increase the size of the table dramatically. To keep the

summary table small, we include only those attributes that are of general interest. Hopefully, most of the data mining tasks can be accomplished using only the summary table. If queries refer to attributes that are not included in the summary table, the underlying database would have to be searched to derive such information.

It can be easily observed that the summary information in the table can be updated dynamically. That is, for any insertions, deletions, or updates, the summary table can always be updated incrementally without scanning the base relation.

3 A Statistical Data Mining System - Design and Implementation

Our goal is to design a data mining system such that ordinary users, even without statistical knowledge, can conduct meaningful and reliable statistical data mining tasks, benefiting from the wealth of statistical techniques. Instead of examining all the data, which can be very time consuming and wasteful, we adopt an interactive data mining approach to allow users to focus on the data they are interested in.

We have selected useful statistical techniques and integrated them into our data mining framework. We observe that a statistical data mining process usually starts out by examining the data of interest, verifying the existences of differences or similarities, and then extracting the relationships. Therefore, we design a set of operators to facilitate this process. These operators allow users to focus on the data of interests with flexibility, examine and comprehend the data informatively, and explore the data in an organized fashion.

After careful reviews of the fundamentals and techniques of conventional statistics, we define the following five operators: **Aggr(egate)** for obtaining summary information about specific groups of tuples; **Compare** for hypothesis testing on sets of tuples; **Related** and **Estimate** for statistical induction, regression analysis, and classification analysis; and **Cluster** for clustering analysis. While each of these operators can accomplish a (part of) particular data mining task, a combination of these operators can be used to accomplish complex data mining tasks. This flexibility allows us to expand the pool of data mining operations easily in the future.

3.1 User Interface



Visual Basic 5.0 is used to build a friendly user interface (see **Figure 3.1**). We aim to provide users with little statistical background an easy way to conduct data mining tasks. Each operator is carefully implemented to ensure as few database accesses as possible to retrieve data required for the computation. Once the required information is obtained, all computations are done locally. The results of the operators are presented in a way such that users can easily comprehend. The user interface is a menu driven window with shortcuts for all the operators on the toolbar as shown in



Figure 3.1. Within the main window, child windows can be moved, resized, minimized and maximized. Each child window corresponds to one operator.

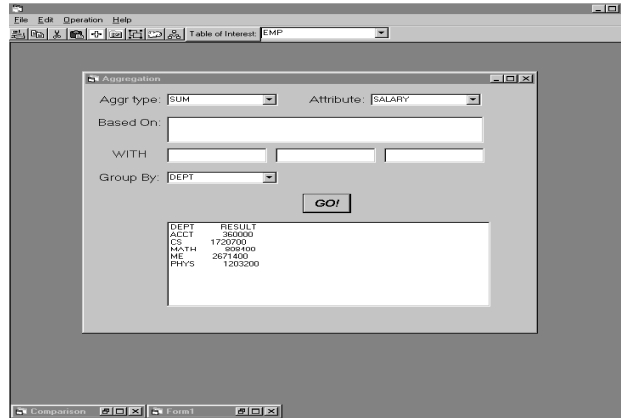


Fig. 1. User interface of the system

3.2 Database Access

The five operators mentioned in the previous sections are implemented on Oracle 8.05 database. Oracle Objects for OLE (OO4O) is used to establish the connection between Visual Basic and Oracle database. OO4O is a product designed to allow easy access to data stored in Oracle databases through any programming or scripting language that supports the Microsoft COM Automation and ActiveX technology. This includes Visual Basic, Visual C++, Visual Basic For Applications (VBA), IIS Active Server Pages (VBScript and JavaScript), and others. The heart of OO4O is the OO4O Automation Server. It is a set of COM automation objects for connecting to Oracle database servers, executing SQL statements and PL/SQL blocks, and accessing the results. Since OO4O is specially designed for Oracle databases, it provides key features for accessing Oracle databases efficiently and easily in a typical client/server environment.

Once a user logs on to our mining system, a connection is established between our data mining client and an Oracle database. The Oracle database can reside on a remote server or a local machine. Two Oracle objects are created during the login process: OraSession and OraDatabase. An OraSession object manages collections of OraDatabase, OraConnection, and OraDynaset objects used within an application. An OraDatabase interface represents a user session to an Oracle database and provides methods for SQL and PL/SQL execution. We can create OraDynaset object to browse and update data created from a SQL SELECT statement afterwards. The following code fragment shows how to create those two objects:

```
Set OraSession = CreateObject("OracleInProcServer.XOraSession")
Set OraDatabase = OraSession.DbOpenDatabase(database, user/passwd, 0&)
```

On the toolbar, a user can select a table of interest. The user is able to perform several data mining tasks on the same table in a section before moving to next table, which can significantly reduce database accesses by overlapping the table selection process of different data mining tasks.

3.3 Knowledge Extraction Operators

We intend to discover inter-field and inter-record knowledge. Finding association among attributes is to discover inter-field patterns (that is, mining in a horizontal direction), while finding similarity of records is to discover inter-record patterns (mining in a vertical direction). The classification and clustering problems belong to object similarity problem, while regression and dependence modeling belongs to the problem of association or relationship among attributes. In this section, we shall use the summary table shown in Table 1 as a running example to illustrate the use of the proposed operators.

Before proceeding, we define the following notations. Let A_1, A_2, \dots, A_n be attributes of a relation. A formula F is made of zero or more atoms connected via the logical operators \wedge (and), \vee (or), and \neg (not). An atom has the form $A_i \text{ op } v$, where A_i is an attribute, v is a value, and op is one of the mathematical comparators $\{=, \leq, <, \geq, >, <>\}$. For example, F can be $\text{Dept}='ME' \wedge \text{Rank}='Prof'$.

(I) Aggregate Operator

This operator provides statistical information about sets of tuples. The operator is denoted as $\text{Aggr}(A; F; G)$. In this format, Aggr can be one of the commonly used aggregate operators in statistics: Count, Sum, Avg(average), Prop(proportion), and Dev(standard deviation). A is the attribute on which the operation is conducted. F is the

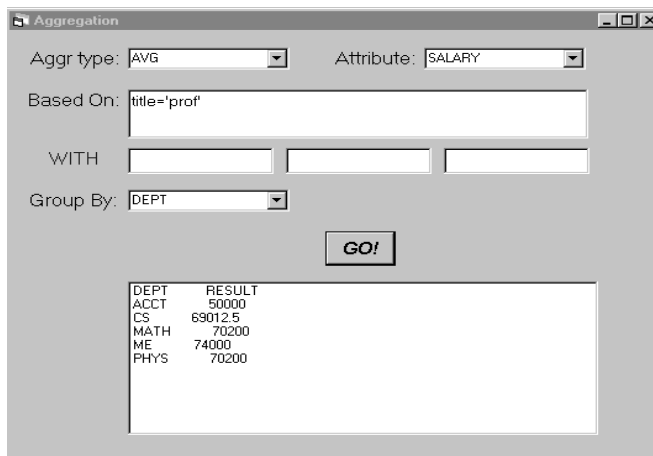


Fig. 2. Avg(salary; title='prof'; dept)

selection formula specifying the data of interest. It can be omitted if the whole data set is to be selected. **G** is the grouping scheme by which users can define their own grouping. Tuples satisfying **F** are grouped based on **G** and then the aggregate operator **Aggr** is performed on the numerical attribute **A** for each group. It can be best demonstrated with the example.

The above query ‘*What are the average salaries of Professors in each department?*’ is accomplished by executing the SQL statement *Select Dept, sum(Count*Avg_salary) /sum(Count) from Emp_sum where Title=‘Prof’ group by Dept* on the summary table. Other aggregate operators are accomplished in the same way except that the functions plugged into the SQL statements are different. In general, the base relations are searched unless the summary table does not contain the required information.

(II) Compare Operator

To compare the characteristics of groups (of tuples), or a characteristic of a group against a constant, the **Compare** operator is defined. The results of the comparisons often determine whether further investigations are necessary. The syntax for this type of operators is **Compare(C; F; G)**, which compares groups of tuples satisfying **F** and grouped by **G** based on the criterion **C**. In this operator format, **C** is the comparison criterion. It has either a format like **Aggr(A)** to specify the characteristic (e.g., Avg(Age)) to be compared between groups, or a format like **Aggr(A) op v** to specify a constant **v** to be compared with the characteristic **Aggr(A)**, where **op** is one of the comparison operators {=, >=, <=, >, <, <>}. Here, **Aggr** is either Avg or Prop because they are the two main characteristics of interest in hypothesis testing. **F** and **G** are defined in the same way as for the Aggregate operator.

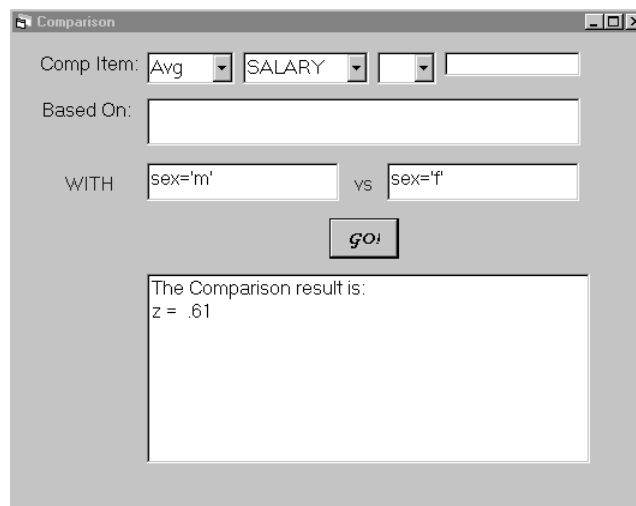


Fig. 3. Compare (avg(salary); ; female: = sex = ‘f’, male: = sex = ‘m’)

The z-statistics is usually computed for hypothesis testing. Based on the result of z-statistics, the corresponding P-value [9] can be obtained from the normal distribution, on which users determine whether there exists a significant difference between the values tested. Depending on the criterion C, z-statistics is calculated with different formulas. Above are examples with various possible criteria.

The above query ‘*Is the average salary of male employees significantly different from that of female employees?*’ compares the mean values. First, we pose a SQL query to the summary table to get the Count, Average, and Dev of the Salary for female staffs. Then, we pose another SQL query to get the same statistics for male staffs. Two summary table accesses are needed for this query. Finally, the z-value is calculated. The user can use the z-value to determine the significance. Alternatively, we could select a default significance level so that, instead of given the z-value, a Yes/No (or high/medium/low, etc) answer is displayed (for users without background in statistics). However, due to the possible different requirements of users and flexibility of choosing different significance levels, we are inclined to display the z-value and let users (with some statistical background) do the explanation by themselves.

The z-statistics [19] is calculated:

$$z = (\bar{X}_1 - \bar{X}_2) \sqrt{\frac{n_1 n_2}{n_2 \delta_1^2 + n_1 \delta_2^2}} \quad (1)$$

where \bar{X}_1 , \bar{X}_2 , n_1 , n_2 , δ_1 , δ_2 are means, counts, and standard deviations of X for groups 1 and 2, respectively. They can be easily retrieved from the summary table. Based on the z-value, the user can then determine whether the test is significant.

The query ‘*Is the average salary of professors in the CS department significantly higher than \$70,000?*’ demonstrates another criterion by comparing between the mean and a constant.

For this query, the Count and the Average and Dev of the salary of professors in CS department can be retrieved using one single SQL statement over the summary table. The z-statistics is then calculated:

$$z = \frac{\bar{X} - v}{\delta / \sqrt{n}} \quad (2)$$

where \bar{X} , n , δ are the mean, count, and standard deviation of X , respectively, and v is the constant value to be compared with.

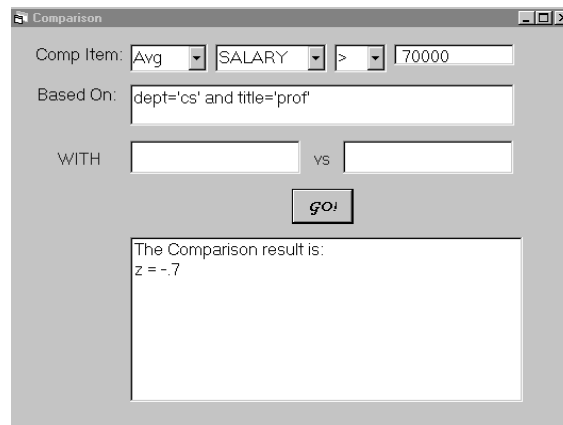


Fig. 4. Compare (avg(salary)>70,000; dept = 'cs' and title='Prof';)

The third example, 'Are the percentages of junior and senior professors significantly different in CS department?', compares the proportions. Note that we have defined on the fly a group, called Senior, as the union of associate or full professors in the operator. The formula for the z-value is:

$$z = \frac{p_1 - p_2}{\sqrt{pq\left(\frac{n_1 + n_2}{n_1 n_2}\right)}} \quad \text{with } p = \frac{n_1 p_1 + n_2 p_2}{n_1 + n_2} \quad \text{and } q = 1 - p \quad (3)$$

where p_1, p_2, n_1, n_2 are proportions and counts of groups 1 and 2, respectively. Since the proportion value is calculated from the count values, so only the Count values from the summary table is required to calculate z .

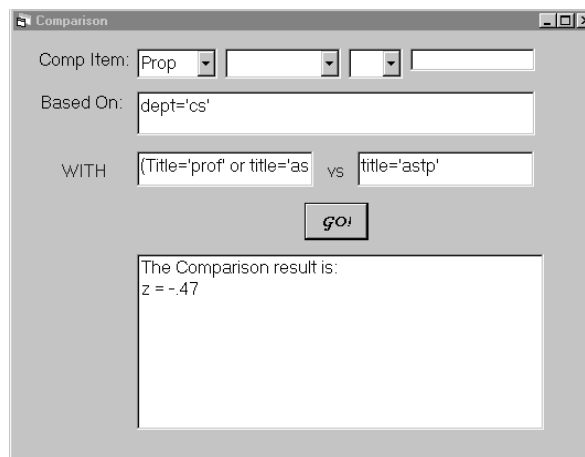


Fig. 5. Compare (prop; dept='cs'; senior: = Title = 'prof' or title = 'asop', junior: = title = 'astp')

(III) Related Operator

This operator is to discover the existence of correlations between a given attribute and a set of other attributes. It is denoted as **Related (A; X; F; G)**, where **A** is a given attribute, **X** is a set of attributes (**X₁, X₂, ...X_n**), which we need to determine whether any of them are related to **A** or not. The operator looks for existence of correlations between attribute **A** and individual attributes of **X** for tuples satisfying **F** grouped by **G**. The operator returns the significance of the hypothesis testing, which can be used to determine whether **A** and **X** are related or not. Depending on the attribute types (numerical or categorical) of **A** and **X**, different statistical techniques are applied.

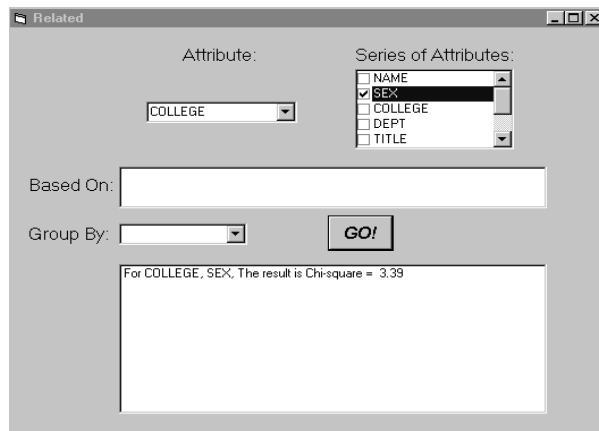


Fig. 6. Related (sex; college; ;)

The above query ‘*Is there any relationship between Sex and College?*’ provides an example where both attributes are categorical. The Count values are needed to compute the Chi-square value. We use “Sex, College” as the grouping scheme to get the numbers of female and male staffs in each college. The SQL statement: *Select College, Sex, sum(Count) from emp_sum group by College, Sex order by College, Sex* is posted against the summary table. Then, we plug the resulting count (sum(Count)) value into the contingency table and compute the Chi-square statistics. The Chi-Square value is computed based on the contingency table with the following formula:

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{4}$$

where O_{ij} is the observed count for cell ij , and E_{ij} is the expected count for cell ij , which is calculated as

$$E_{ij} = n_i n_j / n \tag{5}$$

where n_i and n_j are the total count of row i and column j respectively, and n is the total count of the table.

To construct the contingency table we first use the SQL statement: *Select College, Sex, sum(Count) from emp_sum group by College, Sex order by College, Sex*, and then map the Count (sum(Count)) value into the contingency table.

Table 2. Contingency table for college and sex

	Business	Engineer	Science	Total
F	6	16	33	55
M	3	26	30	59
Total	9	42	63	114

For the case where both attributes are numerical, such as ‘*Is there any relationship between salary and age?*’, we use the t-statistics [19] to examine the significance of a linear relationship. The formula is:

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}} \quad (6)$$

where n is the number of tuples in question, and r is correlation coefficient calculated with the following formula:

$$r = \frac{n\sum AX - (\sum A)(\sum X)}{\sqrt{(n\sum A^2 - (\sum A)^2)(n\sum X^2 - (\sum X)^2)}} \quad (7)$$

where $\sum A^2$, $\sum X^2$, $\sum AX$ all can be derived indirectly from the summary table.

Note that all the values for those terms in equation (7) can be derived either by following their definitions or retrieving from the summary table directly. For simplicity, we show only the linear relationship with single attribute X above. For more complex relationships, please refer to Ratkowsky [17] for the detailed discussion.

With the query ‘*Is there any relationship between salary and title?*’, we have two different types of attributes (one categorical and the other numerical), and the query is posted as **Related** (Salary; Title; ;).

For this query, F -test is conducted to determine the correlation of the two attributes. The formula to calculate F is:

$$F = \frac{SSB/(I-1)}{SSE/(\sum(T_i-1))} \quad (8)$$

In the above formula, I is the number of groups based on **A**. T_i is the number of tuples in group I . SSB is the sum of squares between groups, which is calculated as

$$SSB = \sum n_i (\bar{X}_i - \bar{X})^2 \quad (9)$$

where n_i is the count value in group I , \bar{X} is the overall average of tuples in question, and \bar{X}_i is the average of X in group I . SSE is the error sum of squares within a group, computed as

$$SSE = \sum (n_i - 1) \delta_i^2 \tag{10}$$

where δ_i is the standard deviation of X in Group I . Only Average, Count, and Dev of X for each group based on A are required to calculate the F statistics. We combine the above two queries, ‘Is there any relationship between salary and age?’ and ‘Is there any relationship between salary and title?’ as one query, as displayed in Figure 7.

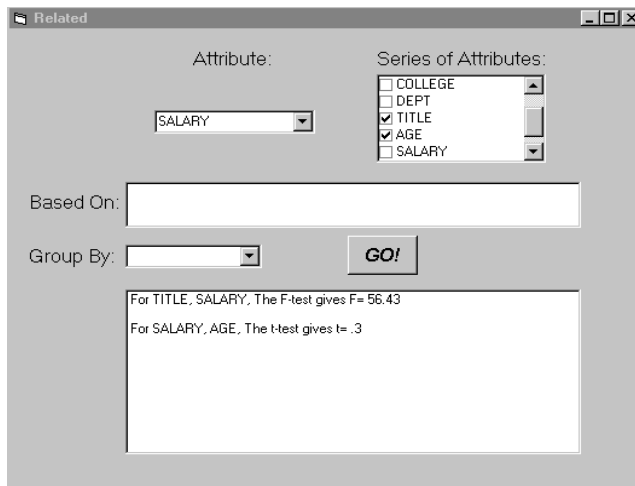


Fig. 7. Related (salary; (age,title); ;)

(IV) Estimate Operator

This operator is designed to estimate the value of an attribute, given the values of another set of attributes, assuming that the former and latter variables are related. The syntax is **Estimate (A; V; F; G)**, where **A** is the attribute whose value is to be estimated, **V** is a set of expressions in the form of “attribute-name = constant” or “attribute-name”, assuming the set of attributes **X**(X_1, X_2, \dots, X_k) are related to **A**; **F** and **G** have the same meaning as in other operators. If **V** is in the form of a set of “attribute-name”, the operator displays the mathematical relationship. Depending on the data type of **A**, different techniques may be applied. We shall call the attribute **A** the dependent attribute and **X** independent attributes.

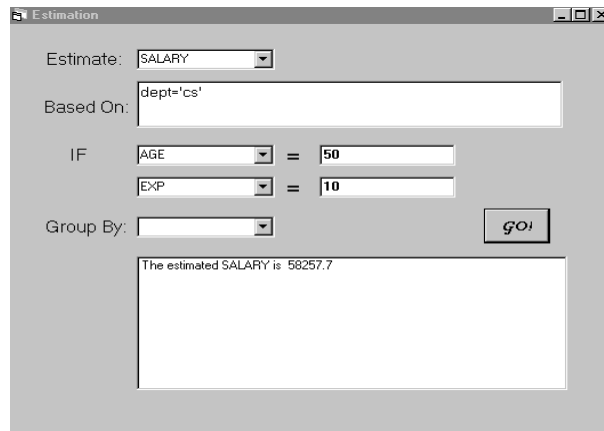


Fig. 8. Estimate(salary; age=50, exp=10; dept='cs')

When the dependent attribute is numerical, as in the query ‘*What is the salary of a 50-year-old staff with 10 years of experience?*’, we need to first get Sum_Salary_Age, Sum_Salary_Exp, Sum_Age_Exp, Avg_Salary, Avg_Exp, Avg_Age, and Count in one disk access. Estimated salary is then calculated according to the regression model [19]. The **A** value is estimated as

$$A = b_0 + b_1X_1 + b_2X_2 + \dots + b_iX_i + \dots + b_kX_k$$

where X_i ($0 \leq i \leq k$) are the attribute values specified in **V**, and b_i is calculated as

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{bmatrix} = [S]^{-1} \begin{bmatrix} S_{a1} \\ S_{a2} \\ \dots \\ S_{ak} \end{bmatrix} \tag{11}$$

where $[S]^{-1}$ is the inverse of matrix $[S]$ with element

$$S_{ij} = \sum_{l=1}^n (X_{il} - \bar{X}_i)(X_{jl} - \bar{X}_j) = \sum_{l=1}^n X_{il}X_{jl} - n\bar{X}_i\bar{X}_j \quad (i, j = 1, 2, \dots, k) \tag{12}$$

$$S_{ii} = (n-1)\delta_i^2 \quad (\text{when } i=j) \tag{13}$$

and

$$S_{ai} = \sum_{l=1}^n (X_{il} - \bar{X}_i)(A_l - \bar{A}) = \sum_{l=1}^n X_{il}A_l - n\bar{X}_i\bar{A} \quad (i, j = 1, 2, \dots, k) \tag{14}$$

$$b_0 = \bar{A} - b_1\bar{X}_1 - b_2\bar{X}_2 - \dots - b_k\bar{X}_k \tag{15}$$

where n is the total number of tuples involved. All the information needed for the computation can be retrieved from the summary table directly.

For the case where the dependent attribute is categorical, we present the query ‘Estimate the Title of a staff with a salary of \$58000 in CS department.’

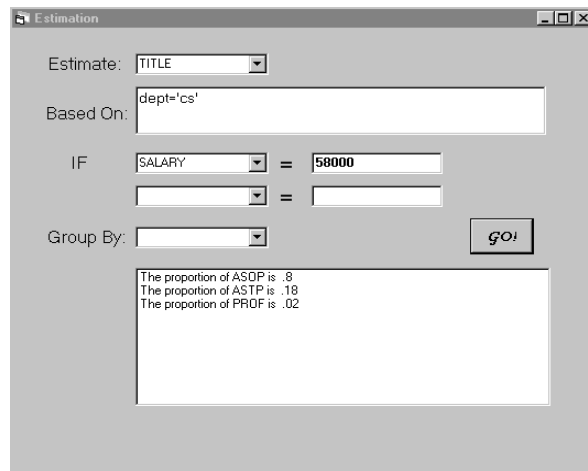


Fig. 9. Estimate (title; salary = 58000; dept = ‘cs’)

This type of queries is also known as classification. Classification analysis [10, 14] is applied to determine the conditional probability that an object is assigned to one of a number of predetermined groups. Let $P(A_i|V)$ be the conditional probability that a tuple belongs to group A_i with a given V . Baye’s rule can be employed to calculate $P(A_i|V)$:

$$P(A_i | V) = \frac{P(V | A_i)P(A_i)}{\sum_{j=1}^n P(V | A_j)P(A_j)} \tag{16}$$

where $P(A_i)$ is the proportion of group A_i in the population, and $P(V|A_i)$, the conditional probability to get a particular set of measurements defined by V given that the object comes from group i , is defined as

$$P(V | A_i) = \frac{1}{(2\pi)^{n/2} |C_i|^{1/2}} \exp[-\frac{1}{2}(V - \mu_i)C_i^{-1}(V - \mu_i)] \tag{17}$$

where C_i is the covariance matrix of i^{th} group, which is

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix} \quad \text{with } \sigma_{pq} = \sigma_{qp} \text{ if } p \neq q, \text{ and} \tag{18}$$

$$\sigma_{ij} = \sqrt{\frac{\sum (X_i - \bar{X}_i)(X_j - \bar{X}_j)}{n-1}} = \sqrt{\frac{\sum X_i X_j - n \bar{X}_i \bar{X}_j}{n-1}} \quad (19)$$

where μ_i is the mean of \mathbf{X} for group A_i ; $V\text{-}\mu_i$ is $1 \times n$ matrix and n is the number of attributes in \mathbf{X} ; $(V\text{-}\mu_i)'$ is the transpose of matrix $V\text{-}\mu_i$.

(V) Cluster Operator

We demonstrate this operator with the query 'Partition the department into three groups based on the average salaries of their female employees.'

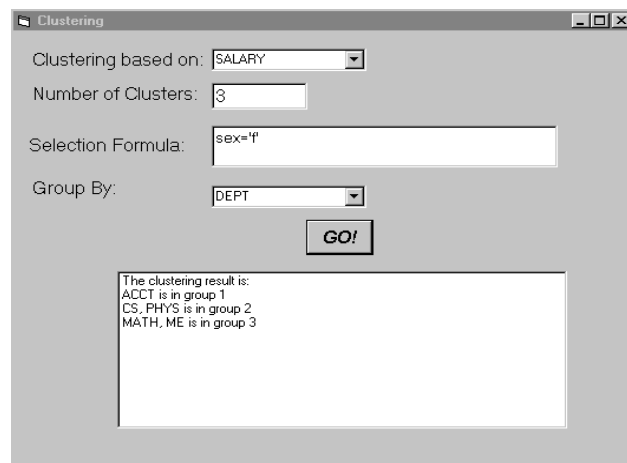


Fig. 10. Cluster(3; salary; sex='F'; dept)

This operator groups objects with common features or similarities together. The syntax is **Cluster(L; X; F; G)**, where **L** is the number of clusters, **X** are attributes that the clustering is based on, **F** is the selection formula, and **G** is the grouping scheme. It partitions objects in **G** satisfying **F** into **L** clusters based on the values of attribute **X**.

There are a variety of mathematical methods for cluster analysis. Ward's minimum variance clustering method [7] is one of the most frequently used. It goes through a series of clustering steps that begin with N clusters, each containing one object, and it ends with one cluster containing all objects. At each step it makes whichever merge of two clusters that will result in the smallest increase in the value of an index E , called the sum-of-square index or variance. E is calculated by the following formula:

$$E = \sum_{c=1}^C \sum_{i=1}^{N_c} (X_{ci} - \bar{X}_C)^2 \quad (20)$$

where X_{ci} is the \mathbf{X} value of the i^{th} object in the c^{th} cluster, \bar{X}_C is the mean of X in the c^{th} cluster, C

is the number of clusters after a merge is made, and N_c is the number of objects in the c^{th} cluster.

The above formula can be rewritten as

$$E = \sum_{c=1}^C (N_c - 1) \delta_c^2 \quad \text{with} \quad \delta_c^2 = \frac{\sum_{i=1}^{N_c} X_{ci}^2 - N_c \overline{X_c}^2}{(N_c - 1)} \quad (21)$$

where δ_c is the standard deviation for c^{th} cluster.

If two clusters A and B are combined, the variance E_{ab} of the combined cluster AB is

$$E_{ab} = (N_a - 1) \delta_a^2 + N_a \overline{X_a}^2 + (N_b - 1) \delta_b^2 + N_b \overline{X_b}^2 - \frac{(N_a \overline{X_a} + N_b \overline{X_b})^2}{N_a + N_b} \quad (22)$$

Here, N_a , N_b , $\overline{X_a}$, $\overline{X_b}$, δ_a , and δ_b are the counts, averages, and deviations of X of cluster A and B, respectively. The deviation δ_{ab} of cluster AB is

$$\delta_{ab} = \sqrt{E_{ab} / (N_a + N_b - 1)} \quad (23)$$

and average X_{ab} of cluster AB is

$$X_{ab} = (N_a \overline{X_a} + N_b \overline{X_b}) / (N_a + N_b) \quad (24)$$

From the above equations, we can see that only the count, average and deviation of X of each group are required to evaluate E following Ward's method.

4 An Interactive Data Mining Example

In this section, we use an example to illustrate the use of the proposed data mining operators, summarized in Table 3, to facilitate interactive data mining.

Table 3. Functions of data mining operators

KNOWLEDGE	OPERATOR
Statistical properties of a group of objects	Aggr
Comparisons of statistical properties	Compare
Correlation between attributes	Related
Relationship, prediction, classification	Estimate
Clustering	Cluster

In the following, we illustrate the use of the operators with a small database. Note that the size of the sample database should not be an issue of concern as we only try to show how these operators can be used to investigate data in a flexible manner. Since the statistical methods underlying our operators are all well discussed in the literature and well received in practice, the reliability of

the test results and knowledge derived should be trustworthy statistically. As for the speed of the mining process, as discussed in the previous section, it would require one to two summary table accesses for each operator if the summary table contains the requested information. Otherwise, the database would have to be searched to compute the requested statistics.

4.1 An Interactive Example

The knowledge to be extracted is highly dependent on the user-specific tasks and interest. To extract useful information, interactions between users and the system are necessary. The operators defined in the previous section can facilitate this interactive process. Note that users can enjoy the flexibility provided by the system to group tuples and investigate relationships among attributes as they wish. During the process, a series of queries may be posed to search for patterns of interest. The results of one query may lead to another query, or single out ones that are not of interest. For example, one may first compute the statistical properties of groups of tuples using **Aggr** operator, and then use **Compare** to examine if there are significant differences in these properties statistically. If significant differences are found, one can use **Related** operator to find attributes that could have contributed to these differences. If attributes are correlated, then we can find out the relationship or use known attribute values to **Estimate** an unknown attribute value.

Example. The survival rates of animal semen after storage were measured at various combinations of concentrations of three materials as shown in Table 4 [19]. One is concerned with the correlation between Y (% survival) and X_1 , X_2 , and X_3 (the concentrations of three materials). If they are correlated, what is the relationship?

Table 4. A database of animal semen survival percentages

No	Y (% survival)	X_1 (weight %)	X_2 (weight %)	X_3 (weight %)
1	25.5	1.74	5.30	10.80
2	31.2	6.32	5.42	9.40
3	25.9	6.22	8.41	7.20
4	38.4	10.52	4.63	8.50
5	18.4	1.19	11.60	9.40
6	26.7	1.22	5.85	9.90
7	26.4	4.10	6.62	8.00
8	25.9	6.32	8.72	9.10
9	32.0	4.08	4.42	8.70
10	25.2	4.15	7.60	9.20
11	39.7	10.15	4.83	9.40
12	35.7	1.72	3.12	7.60
13	26.5	1.70	5.30	8.20

We can use **related**(Y, X_i), $1 \leq i \leq 3$, to check if Y and X_i are correlated. Assume the common significance level of 0.1 is postulated. The related operator reveals that by conducting t-tests (Eq.(6)), X_1 and X_2 are strongly correlated to Y while X_3 is not because its t-test value $|t_3| = 0.556 < t_{0.005}(9) = 3.25$.

Once the existence of correlations between Y and $\{X_1, X_2\}$ is confirmed by the related operator, one can proceed to explore the relationship between the set of related attributes (i.e., Y, X_1 , and X_2) by applying the **estimate**($Y; X_1, X_2;$) operator. The estimate operator applies the regression or classification analysis to the set of variables Y, X_1 , and X_2 and yields the result: $Y = 36.094 + 1.031 X_1 - 1.870 X_2$. Once the relationship is found, one can use it to predict the Y, X_1 , or X_2 value based on the given values. For example, **estimate**($Y; X_1 = 1.5, X_2 = 7.6;$) would yield an estimated value of 23.43% for the survival rate Y based on the materials 1 and 2 of concentrations 1.5% and 7.6%, respectively.

5 Conclusions

In this study, an interactive approach to statistical data mining is developed. The core of the approach is a set of operators that can be used to investigate the data and extract various types of knowledge from a large database. We design these operators from a statistical point of view and use statistical analysis as the foundation. The major innovations of the presented method are:

- Integration of a variety of knowledge exploration operators for users to extract certain knowledge.
- Utilization of a summary table, instead of the database itself, as the target of knowledge extraction.
- Capability to provide the following four types of knowledge:
 - Statistical properties;
 - Correlation between attributes;
 - Linear and simple nonlinear mathematical relationships and prediction;
 - Classification of the objects in terms of given attributes.

The advantages of using statistical methods for data analysis are their solid theoretical foundation and wide recognition in various fields. The use of the summary table improves the querying efficiency while the knowledge generation operators facilitate interactive search of knowledge. This research provides users with an integrated tool for conducting knowledge discovery using statistical techniques.

There is still a lot of work that can be done in the future. A major direction is to expand the current set of operators to extract more types of knowledge and include more statistical methods for exploring data. One can also investigate other clustering methods as the statistical clustering method present in the paper may be slow when the data set is large.

References



1. R. Agrawal, T. Imielinski, and A. Swami: Mining Association Rules between Sets of Items in Large Databases, in: *Proc. ACM SIGMOD Conference*, 207-216, Washington, D.C. (1993)
2. R. Agrawal and R. Srikant: Mining Sequential Patterns, in: *Proc. ICDE Conference*, Taipei, Taiwan. (1995)
3. D. J. Berndt and J. Clifford: Finding Patterns in Time Series: A Dynamic Programming Approach, in: U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, ed., *Advances in Knowledge Discovery and Data Mining*, 229-248, AAA Press / The MIT Press, Menlo Park, California. (1996)
4. R. Brachman et al.: Integrated Support for Data Archaeology, *International Journal of Intelligent and Cooperative Information System* 2(2) (1993) 159-185.
5. L. Breiman, J. Friedman, R. Olshen, and C. Stone: *Classification and Regression Trees*, Wadsworth, Belmont, CA. (1984)
6. K. Chan and A. Wong: A Statistical Technique for Extracting Classificatory Knowledge from Databases, in: G. Piatetsky-Shapiro, ed., *Knowledge Discovery in Databases*, 107-123, AAAI/MIT Press. (1991)
7. B. Everitt: *Cluster Analysis*, Second Edition, Halsted Press, John Wiley & Sons, Inc., New York. (1980)
8. W. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus: Knowledge Discovery in Database: An Overview, In: W.J. Frawley and G. Piatetsky-Shapiro, ed., *Knowledge Discovery in Database*, 1-27, AAAI/MIT Press, Cambridge, Mass. (1991)
9. G. Glass and J.C. Stanley: *Statistical Methods in Education and Psychology*, Prentice-Hall, Inc. (1970)
10. A. Gordon: *Classification*, Chapman and Hall, London & New York. (1981)
11. J. Han and M. Kamber: *Data Mining: Concepts and Techniques*, Morgan Kaufmann. (2001)
12. J. Han et al, DBMiner: Interactive Mining of Multiple-Level Knowledge in Large Relational Databases, in: *Proc. SIGMOD '96*, 50-59, Montreal, Canada.
13. J. Han, J. Pei, and Y. Yin: Mining Frequent Patterns without Candidate Generation, in: *Proc. ACM SIGMOD '00*, 1-12, Dallas Texas. (2000)
14. D. Hand: *Discrimination and Classification*. John Wiley and Sons, Chichester, U.K. (1981)
15. A. Jain, M. Murty, and P. Flynn: Data Clustering: a survey, in: *ACM Computing Survey* 31 (1999) 264-323.
16. J. Quinlan: *C4.5: Programs for Machine Learning*, Morgan Kaufman. (1993)
17. D.A. Ratkowsky: *Nonlinear Regression Modeling*, Marcel Dekker, Inc., New York and Basel. (1983)
18. P. Selfridge, D. Srivastava, and L.O. Wilson: IDEA: Interactive Data Exploration and Analysis, in: *Proc. SIGMOD '96*, 24-34, Montreal, Canada. (1996)
19. R. Walpole and R.H. Myers: *Probability and Statistics for Engineers and Scientists*, Fourth Edition, Macmillan Publishing Company. (1989)
20. T. Zhang, R. Ramakrishnan, and M. Linvy: Birch: an Efficient Data Clustering Method for Very Large Databases, in: *Proc. ACM SIGMOD '96*, 103-114, Montreal. (1996)