Disparity of produced values from Aggregate Functions

Lamia Abed Noor
Computer Science Department
College of Sciences
University of Al-Qadissiya

Disparity of produced values from Aggregate Functions

Abstract

The aggregate functions are used in extraction features from groups of data. These functions are varied in nature of produced features that are represented by aggregated values. Our work in this paper is interested with discrete aggregate functions in order to evaluate the degree of closeness the result aggregated values from the source data. To achieve our aim, we use in this paper two discrete aggregate functions; one is conventional function and other is proposed one.

Keywords: aggregate, relational data base, disparity.

1. Introduction

Transaction files contain huge data that are located in groups of records that are cooperated with other files in relational data base. The aggregate function plays an important role in summarizing the interested attribute in the groups and introduces one value that express set of values. Aggregation functions are categorized according to type of data in attribute. Continuous aggregate functions like MAX,MIN,MEAN are used with continuous values. While discrete values, special aggregate function such as MODE is the conventional function is used. This paper will produce in section two, some background about the aggregate function in relational data mining that care with the power of features that are extracted from groups of records. Section three; we introduce the algorithms of two specified functions through method description. The practical work will be described in section four. At last, we explain the conclusion in section five.

2. Aggregate and Relational Database

The kinds of database are based on the organization of data that spread on one or more tables and access way to the component data in these tables.

The relational model was formally introduced by Dr. E. F. Codd in 1970[1] as a structure of relational databases and evolved since then. Conceptually, the data are organized as entities, relations and attributes. Entities are the principle data objects about which information are to be collected. Entities are usually recognizable concepts such as person, places, things, or event which have relevance to the database. An entity composed of set of attributes. Attributes are atomic components describe the entity of which they are associated. Attributes can be classified as identifiers or descriptors.

An entity can be associated with other entities through what are called relationships. Typically, a relationship is indicated by a verb connecting two or more entities. The relationships are classified according to their connectivity and cardinality. The connectivity describes the mapping of associated entity instance in the relationship. The values of connectivity are "one" or "many". The cardinality is the actual number of related occurrences for each of two entities.

The relational model in relational database introduce have rich informative in the associations, have. The relationships that associates between different tables with identified as one-many or many- many can add more features to be exploited in the dependence network using a suitable function that is called aggregate[2]. However, the aggregate functions are varied; the features are provided will be different. They are used in order to introduce a new value, describes multiple values. Aggregate functions are functions on sets of records. Given a set of records and some instance of an aggregate function, a feature of the set which summarizes the set is computed. Then, aggregates are used to characterize the structural information that is stored in tables and associations between them. Consider two tables P and Q, linked by a one-to-many association A. For each record in P, there are multiple records in Q, and A defines a grouping over Q. An aggregate function can now be used to describe each group by a single value. Because a single value is specified for every record in P, we can think of the aggregate as a virtual attribute of P[3]. As result, P will have additional attribute and can be searched for existences of its interaction with other attributes in P.

3. Method Description

Through the tracing of aggregation function, we found that more information may be loosed using unsuitable functions especially with the discrete attribute that have unrelated states values in contrast to the information exists in continuous attribute; where it can be abstracted successfully with aggregation function because each state in the set of objects can cooperate in producing the aggregated value for example, average is aggregate function is used for continuous values, it is obvious all the values in bag participate in producing the average value. While, the aggregate function that deal with discrete attribute like high frequency (mode) is concerned with the frequencies of the states inside the set of related objects and produces the aggregated value which represents the highest frequency in the set of related objects; this can forbid other states to be represented through the aggregate values in spite of the possibility of hidden states importance from the significantly. We refer to this aggregate function as conventional function and its algorithm is listed in Figure(1.a).

Input: Database (D) organized in Transaction table(X) consists of groups of records (Di) with specific attribute(X.A) to be aggregated

Output: Aggregate value for each group of instances of X.A

(a) **Algorithm 1 (Conventional Function)**

1- For i=1 to m (m number of groups) execute step(2-4)

2- Compute frequency(j), j $\in$ {states of (X.A) in X}:

3-Search for the largest frequency from set of frequencies of states of (X.A)

4- Let freq(k) is the largets frequency

So k is the state value is the aggregated value to represent Di for attribute X.A

(b) Algorithm 2 (Proposed Function)

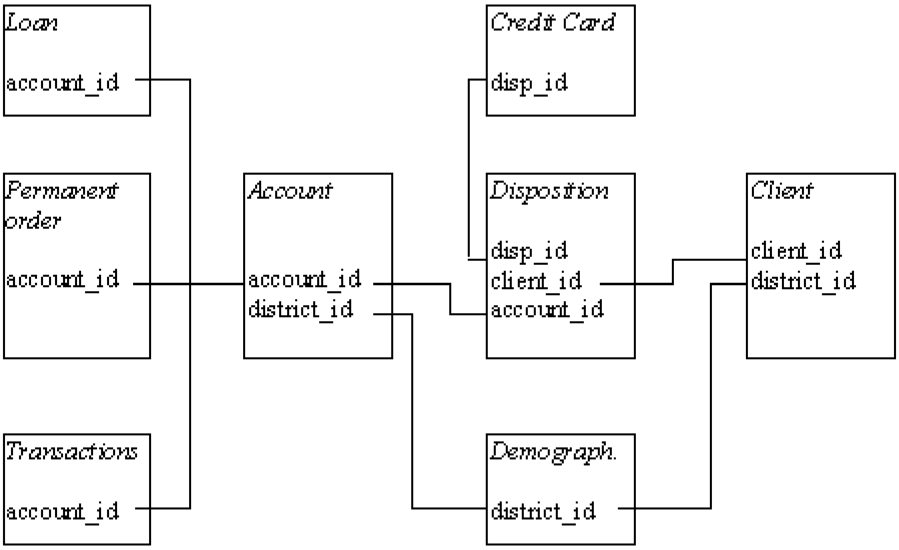
1- Compute frequency(j) of each state j $\in$ {states of X.A}

Figure(1) algorithms .(a) algorithm for Conventional aggregate function. (b) algorithm for Proposed aggregate function

The other proposed aggregate function exploits the significant frequency of states that are more important in the scarce and attempt to rise out in aggregate step. The steps of algorithm to perform the propose function are listed in Figure (1.b).

4. Practical Work

Two algorithms in Figure(1) are executed on experimented data that are obtained from web site[4] that are contains relational data base related with financial data and shown in Figure (2).



Figure(2) Experimented Data

Multiple tables are related in relational data base in Figure(2); the interested table is “Transactions” that is related with “Account” table with cardinality “many” to “one”. The records in “ Transactions” are groubed according to number of records in “Account”. The number of records in “Transactions” is about (1050000 records) and the number of records in “Account is about (4500 records), therefore the records in “Transactions” are grouped into (4500) groups. We aim to conclude for specific attribute in “Transactions” single value to face the number of records in “Account”. Five attributes in “Transactions” are to be aggregated; “Date”, “Type”, “Operation”, “Symbol_k”, and “Bank_to”. We apply two algorithm in Figure(2) and then compute the Disparity of resulted value with the original data source in “Transaction” as :

$$Disparity = \sum_{k} frequency(aggregate\_value) - frequency(\exp ected\_value)$$

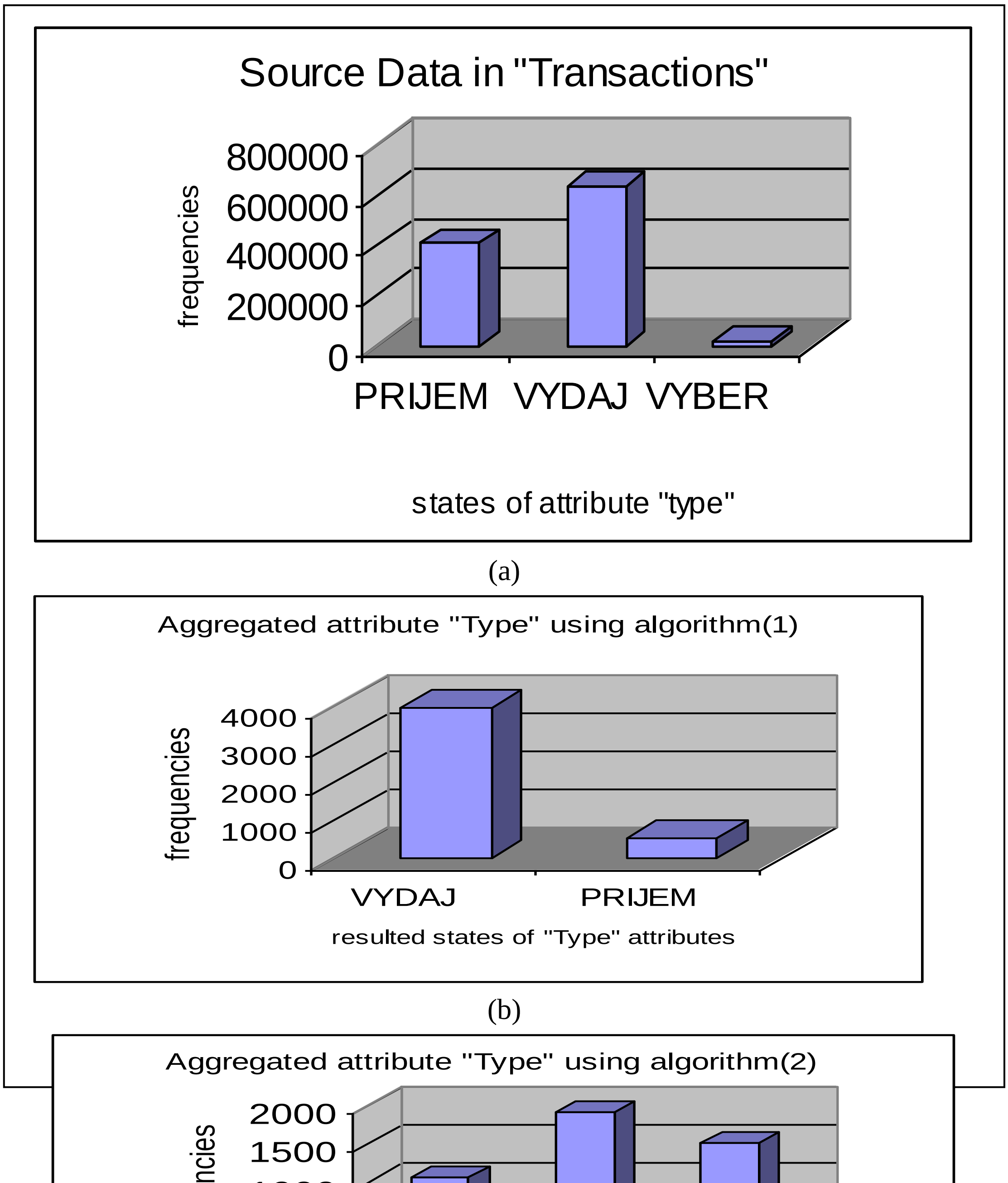
Wher k is the number of states of aggregated attribute

The expected value is computed from frequency of states value in original table as:

expected_value(i)=(frequency(i) /total record)*number of groups.(i) refers to state of specific attribute.

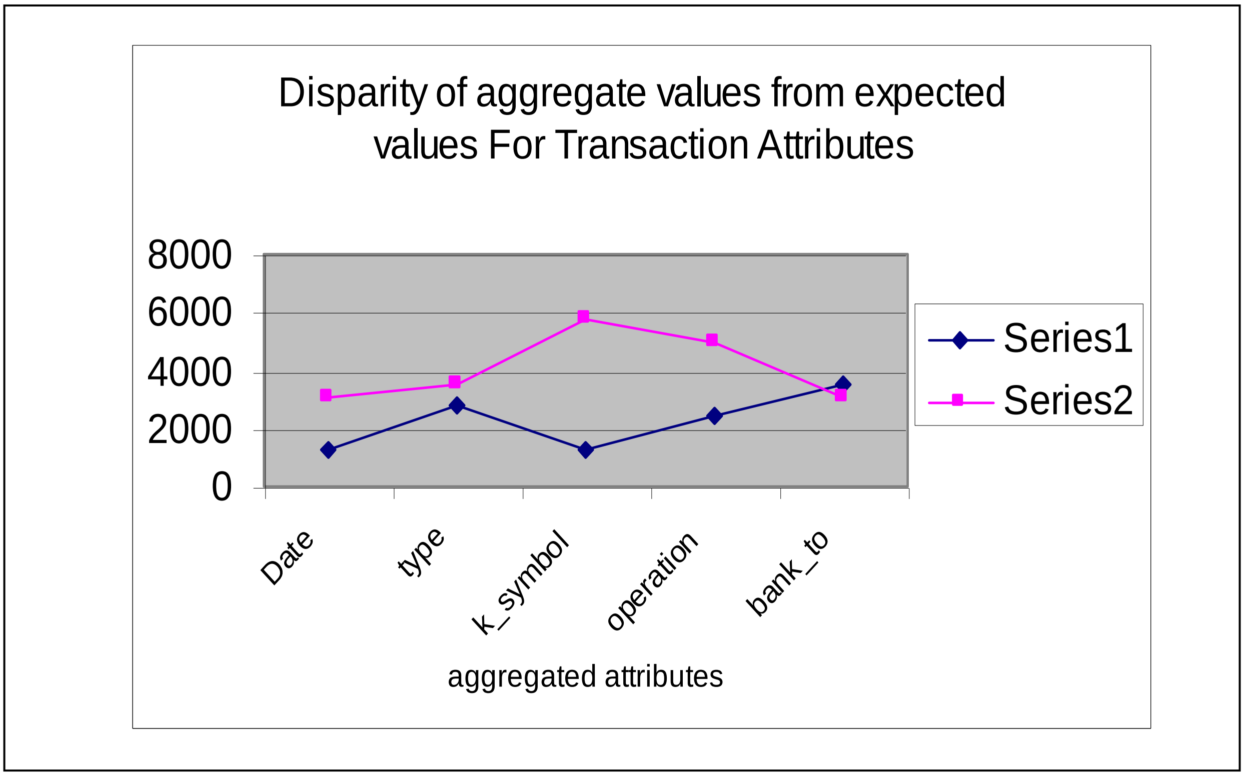
Figure (3) shows some of results that is specified with “Type” attribute. The frequencies of “Type” states(PRIJEM,VYDAG,VYBER) in source data in “Transactions are shown in Figure(3.a). After the data are aggregated according to conventional aggregated function, the resulted states values are shown in Figure(3.a). While Figure(3.c) describes the results of proposed aggregate function on “Type”. It is obvious from these results in Figure(3.b), the aggregate step hides one of “Type” states that is known “VYBER” because it has less frequencies while the proposed aggregate function rise this state as shown in Figure (3.c).

In addition we can see more results that explain the activity of each function on rest of attributes after we computed the disparity of each states to each attribute and collected them in Figure(4); series (2) represent the disparity of aggregated states from conventional function, series (1) refers to disparity of aggregated states from proposed function. Also, we can see the low disparity is achieved with series (1) for the most of attributes, while the high disparity come from using conventional function that is shown by series (2).



(a)

(b)



Figure(4) Disparity of resulted states frequencies in face with expected values.

5. Conclusion

The results, we obtained from our work approve that we must give some attention to using aggregate functions especially with discrete attributes and no function can be standard. There is some work to inspect to suitable one for each domain because we see not all attributes achieve the best disparity for proposed function, we notice attribute “Bank_to” has higher disparity value from proposed function than conventional function. The reason for this contrast is according to its data domain.

References

[1] Codd E. F., “Extending the Database Relational Model to Capture More Meaning”, ACM Transaction on Database Systems, vol.4, No.4,December 1979.

[2] Heckerman C. Meek and Koller D., “Probabilistic Models for Relational Data”, 2004, Technical Report MSR-TR-2004-30, Microsoft.

[3] Knobbe A. J., de Hass M., and Siebes A., "Propositionalisation and Aggregates", Principles of Data Mining and Knowledge Discovery, 5th Europea Conference, PKDD, Freiburg Germany, September 3-5 Preceding Lectures Notes in Computer Science Springer 2001 p p.277-28

[4] Berk 2000 Discovery Challenge.