

# The Computation of Semantic Data Cube

Yubao Liu and Jian Yin

Department of Computer Science of Sun Yat-Sen University,  
510275, Guangzhou, P.R. China  
{liuyubao, issjyin}@zsu.edu.cn

**Abstract.** The multidimensional analysis based on data cube has been growing interest. However existing data cube model usually does not have the semantics of attributes and hence the analysis usually provides results with raw numbers and ignores the real meanings of these numbers. An example result is that the total sales of PC in this year are above 2000. The semantics of sale performance, high or low, is not clear and that is not easy to be understood for decision makers. The semantic data cube model with linguistic semantics is presented in this paper. The semantic data cube uses fuzzy set to represent the linguistic semantics of the dimensions and measures of data cube. The computation of semantic data cube is studied and the serial and parallel computation algorithms are presented. The experiments on the synthetic datasets show that the algorithms are scalable and efficient.

## 1 Introduction

Recently, there have been growing interests in multidimensional analysis of data warehouses [1]. Most of such analyses involve data cube [2] based summary analysis. However existing data cube model usually does not have the semantics and hence the analysis usually provides results with raw numbers and ignores the real meaning of these numbers. An example result is that the total sales of PC in this year are above 2000. The semantics of sale performance, high or low, is not clear and that is not easy to be understood for the decision makers. Fuzzy technology provides a useful method for describing the interface between human conceptual categories and data. In this paper, the semantic data cube model with such linguistic semantics is presented. It uses fuzzy set to represent the linguistic semantics of dimensions and measures of data cube. The computation of semantic data cube is studied and the serial computation algorithm sCRT and parallel psCRT (short for parallel semantic cube computation from relation table) are presented to compute the semantic data cube from relation table. The performance of the computation algorithms is tested on the synthetic (i.e. algorithmically generated) datasets and the results show that the algorithms are scalable and efficient.

There are some related works regarding our problem. In qc-trees [3], the structural semantics of data cube, such as, the roll up and drill down operations of data cube are studied. Those structural semantics are different from our linguistic semantics.

In explanatory semantics model [4], the fuzzy technology is also used to construct fuzzy data cube model and it is the closest to the semantic data cube in this paper. However, the explanatory semantics model mainly focuses on the semantics of measure attributes. The semantics of dimension attributes are not considered. In semantic data cube, the linguistic semantics of measures and dimensions are both represented. In addition, the computation of cube model is not given in [4]. The existing cube computation algorithm [5] mainly studied the computation of general data cube from relation table. The semantics are not included in the computation algorithm. The semantic cube model is also different from the RSM model included in the knowledge grid platform [6], in the foundation, the managed objects, the normalization basis, the operation feature, and the interchange basis aspects.

## 2 Preliminaries

### 2.1 Fuzzy Set

The concept of a fuzzy set extends the notion of a regular crisp set in order to express classes with ill-defined boundaries, corresponding in particular to linguistic values such as "tall", "young", "important" etc. Within this framework, there is a gradual rather than sharp transition between non-membership and full membership. A degree of membership is associated with every element  $x$  of the universal set  $X$ . It takes its value in the interval  $[0,1]$  instead of the pair  $\{0,1\}$ . Such a membership assigning function  $\mu_A: X \rightarrow [0,1]$  is called a membership function and the set defined by it is a fuzzy set. The concept of " $\alpha$ -cut" of a fuzzy set means a subset made of those elements whose membership is over or equal to  $\alpha$ :  $(A_\alpha = \{x \in X | \mu_A(x) \geq \alpha\})$ . A fuzzy predicate expresses the degree to which the arguments satisfy the predicate.

### 2.2 Data Cube Model

The CUBE BY operator [2] is a multidimensional extension of relational GROUP BY operator and is used to compute a data cube from a relation table. While the semantics of the CUBE BY operator is to partition a relation into groups based on the values of the attributes specified in the CUBE BY operator and then apply aggregations functions to each of such groups, the CUBE BY operator computes GROUP BY corresponding to all possible combinations of attributes in the CUBE BY operator. In general, a CUBE BY operator on  $n$  attributes computes  $2^n$  GROUP BYs, or cuboids. The grouping attributes are called dimensions and the aggregated attributes are called measures. A tuple with dimension attributes and measure attributes in a data cube is called a cell.

In this paper, we call such cuboid that is partitioned on  $j$  dimensions size- $j$  cuboid. For example, cuboid  $A$  is a size-1 cuboid and cuboid  $AB$ , that is partitioned on dimensions  $A$  and  $B$ , is a size-2 cuboid. Especially, the cuboid  $ALL$  is called size-0 cuboid.

### 3 Semantic Data Cube Model

Table.1 shows a semantic data cube with historical sales data. There are three cells in the data cube. The dimensions attributes are ‘product’, ‘year’ and ‘location’ and the measure attribute is ‘sales amount’.

**Table 1.** An example of a semantic data cube

Product	Year semantics	Location	Sales semantics
PC	Recent/1	Guangzhou	High/1
TV	Recent/1	Shanghai	Ordinary/0.8
Camera	Old/0.6	Beijing	Low/1

In table.1, the ‘year’ dimension includes the semantic information, ‘recent/1’ and ‘old/0.6’, and the ‘sales amount’ measure also includes the similar semantics that describe the sales performance such as ‘high/1’, ‘ordinary/0.8’ and ‘low/1’. As shown in the table, the semantics of dimensions and measures attributes of data cube are included in the semantic data cube. The semantics of attributes includes the linguistic semantics such as ‘recent’ and the corresponding degrees of membership function such as ‘1’ in ‘recent/1’. In general, the field experts assign the membership functions according to the fact requirements.

Based on the semantic data cube model, the multidimensional analyses with the semantics can be directly supported. For example, find out the products that have high sales in the recent years based on the semantic data cube in table 1. The answer may be the product of PC and the degree is 1.

## 4 The Computation Algorithms

### 4.1 The SCRT Algorithm

The sCRT algorithm is an extension of the existing cube computation algorithm BUC [5]. Similar to the BUC algorithm, the framework of sCRT is a recursive processing and computes semantic data cube from the bottom of cuboids-tree to the up and has the same partition procedure. The difference from BUC algorithm is that sCRT is used to compute completely semantic data cube but traditional and partial data cube (i.e. iceberg cube). The description of sCRT algorithm is given in table 2. In the description, the cardinality of one attribute is the number of different attribute values. The membership function of measures and dimensions are the input arguments. In the computation process, these functions are used to compute the semantics of measures and dimensions.

**Table 2.** The description of sCRT algorithm

---

**PROCEDURE sCRT**

Inputs: input: the relation to aggregate. dim: the starting dimension for this iteration. memberFunction[i]: the membership function associated with the dimensions and measures.

Globals: Constant numDims: the total number of dimensions. Constant numMeas: the total number of measures. Constant cardinality[numDims]: the cardinality of each dimension. outputRec: the current output record. dataCount[numDims]: stores the size of each partition. dataCount[i] is a list of integers of size cardinality[i].

Outputs: Recursively output one record of semantic data cube.

Method:

1. Aggregate (input); {place results in outputRec }
  2. for j=numDims; j< numDims+numMeas; j++ do
  3.   if memberFunction[j] then {check the membership function list}
  4.    outputRec.dim[j]=  $f_m(\text{outputRec.aggr})$ ; {compute the semantics of measures }
  5.   end if
  6. write outputRec;
  7. for d=dim; d< numDims; d++ do
  8.   let C=cardinality[d]
  9.   Partition (input, d, C, dataCount[d]);
  10.   let k=0;
  11.   for i=0; i<C; i++ do {For each partition}
  12.     let c=dataCount[d][i]
  13.     if c>=1 then {The sCRT stop here}
  14.       if memberFunction[i] then
  15.          outputRec.dim[d]= $f_m(\text{input}[k].\text{dim}[d])$ ; {compute dimension semantics}
  16.          sCRT (input[k..k+c], d+1);
  17.       end if
  18.     end if
  19.     k+=c;
  20.   end for
  21. outputRec.dim[d]=ALL;
  22. end for
- 

## 4.2 The PSCRT Algorithm

The size of a complete data cube is huge and the cost of cube computation is expensive. In order to enhance the computing efficiency, the parallel algorithm psCRT is designed. The algorithm assumes a shared-nothing architecture where each of  $n$  processors  $p_i$  ( $1 \leq i \leq n$ ) has a private memory and the processors are connected by a communication network and can communicate by passing messages. The synchronization of processors is controlled by a processor  $p_1$ .

In psCRT, the processor-cuboid map represents the relation between a processor and the cuboids handled by the processor. For example, suppose that the processors are  $p_1$ ,  $p_2$  and  $p_3$ . The dimensions are A, B, C and D. For balancing the number of

cells handled by each processor, size-1 cuboid is assigned to  $p_1$ , size-2 cuboid is assigned to  $p_2$ , and size-0, size-3 and size-4 cuboid are assigned to  $p_3$ . In general, suppose that Numcuboid (i) denotes the number of cells in size-i cuboid. From the description of data cube model, it is easy to know that Numcuboid (i)  $\geq$  Numcuboid (j) where  $0 < i < j$ . The local cuboids in psCRT are computed by sCRT. The combination procedure in psCRT algorithm searches each cell of the local cuboids and inserts the cells with computed semantics into the semantic data cube D.

**Table.3** The description of psCRT algorithm

---

PROCEDURE psCRT

Inputs: R is the given relation table, n is the number of processor, and the processor-cuboid map is a list that represents the relation between cuboids and processors.  
 memberFunction: The membership function associated with the dimensions and measures.

Outputs: D is the semantic data cube

Method:

1. The given relation table R is evenly divided n partitions  $R_i$  ( $1 \leq i \leq n$ ). The processor-cuboid map is sent to the other processors.
  2. The processor  $p_i$  ( $1 \leq i \leq n$ ) computes the corresponding data cube on  $R_i$  in parallel and send the local cuboids to the corresponding processors according to the processor-cuboid map.
  3. The processor  $p_i$  ( $1 \leq i \leq n$ ) in parallel combines the different cuboids into a locally data cube  $D_i$ .
  4. The local data cube  $D_i$  ( $1 \leq i \leq n$ ) is sent to the processor  $p_1$ , and  $p_1$  combines  $D_i$  into a complete data cube D.
- 

## 5 The Performance Tests

In this section, we present our experiments on the performance of sCRT and psCRT. All experiments are performed using synthetic datasets. The dataset tuples follow the uniform distribution. The number of dimension is set to 5. The cardinality of all attributes is set to 1000. The size of datasets (i.e. the number of tuples) is varied from 10,000 to 30,000 with an interval 10,000. The aggregate function used in the cube computation algorithm is SUM function. In all tests, one dimension (the first dimension) and one measure are chosen as the semantic attributes and associated with the membership function. All experiments are conducted on PC platform and two processors are used. The first processor  $p_1$  is with an Intel Pentium 1.5GHz CPU, 256M RAM, and the second processor  $p_2$  is with an Intel Pentium 500M CPU, 256M RAM. The tests of sCRT are based on the first processor. The processor-cuboid map is shown as following: size-1 cuboid, size-5 cuboid, size-3 cuboid are assigned to  $p_1$ . size-2 cuboid, size-4 cuboid and size-0 cuboid (i.e. ALL) are assigned to  $p_2$ . Just as shown in table 4, both algorithms are scalable with the size of dataset. However psCRT outperforms sCRT. The key factor that affects the efficiency of the psCRT algorithm is the spending of communication between processors.

**Table 4.** The performance test results

Size of datasets	Runtime of sCRT (ms)	Runtime of psCRT (ms)
10,000	2433	1735
20,000	4716	3841
30,000	5758	4720

## 6 Conclusions

In this paper, the semantic data cube model with linguistic semantics is presented. The semantic data cube uses fuzzy set to represent the linguistic semantics of dimensions and measures of data cube. The computation algorithms for the semantic data cube are presented. The test results show that the algorithms are effective and scalable. The parallel data analysis based on the semantic cube is our future work.

## Acknowledgements

The paper is supported by National Natural Science Foundation of China (No.60205007), Natural Science Foundation of Guangdong Province (No.031558, No.04300462), Research Foundation of National Science and Technology Plan Project (No.2004BA721A02), Research Foundation of Science, Technology Plan Project in Guangdong Province (No.2003C50118), Research Foundation of Science, Technology Plan Project in Guangzhou City(No.2002Z3-E0017) and Youth Research Foundation of School of Information Science & Technology at Sun Yat-sen University (No.350416).

## References

1. Palpanas, T.: Knowledge Discovery in Data Warehouses. *SIGMOD Record*. 3 (2000) 88-100
2. Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In: Stanley Y.W.Su (ed.): *Proceedings of Int. Conf. on Data Engineering*. IEEE Computer Society Press (1996) 152-159
3. Lakshmanan, L.V.S., Pei, J., Zhao, Y.: QC-Trees: an efficient Summary Structure for semantic OLAP. In: Alon Y.Halevy, Zachary G.Ives, Doan, A. (eds.): *Proceedings of ACM SIGMOD Int. Conf. on Management of Data*. ACM Press (2003) 64-75
4. Feng, L., Dillon, T. S.: Using Fuzzy Linguistic Representations to Provide Explanatory Semantics for Data Warehouses. *IEEE Transactions on Knowledge and Data Engineering*. 1 (2003) 86-102
5. Beyer, K., Ramakrishnan, R.: Bottom-up computation of sparse and Iceberg CUBE. In: Delis, A., Faloutsos, C., Ghandeharizadeh, S. (eds.): *Proceedings of ACM SIGMOD Int. Conf. on Management of Data*. ACM Press (1999) 359-370
6. Zhuge, H.: Resource Space Grid: Model, Method and Platform. *Concurrency and Computation: Practice and Experience*. 14 (2004) 1385-1413