International Journal of Hybrid Information Technology
Vol.9, No.3 (2016), pp. 282-290
http://dx.doi.org/10.14257/ijhit.2016.9.3.26

Iceberg-cubes with Entropy Query for Data Compression Processing

Haiyong Luo

Department of Mathematics and Computer Science, Xinyu University, China
luohaiyongjx@163.com

Abstract

As the increasing usage of data generating devices like cameras, mobile phones, Auto-ID technologies, and so on, huge amount of data are created. Data compression is very important. In order to use the suitable compression methods on reducing the data volume, this paper uses iceberg-cubes to compress the data based on the entropy query mechanism. Using the definition of iceberg-cubes, this paper uses the entropy query principle for getting the key value from the original datasets. The iceberg-cubes are then used for generating the compressed data which should be stored in the hardware devices. It is observed that these proposed algorithms could achieve 32.46% compression ratio averagely.

Keywords*: Data-cube, Entropy Query, Algorithm, Iceberg-cubes*

1. Introduction

As the increasing usage of electronic devices like cameras, mobile phones, Auto-ID technologies, and so on, huge amount of data are created [1]. Take USA for example, the average consumer used 733 MB of data a month in the first quarter, according to Nielsen. Millions of subscribers own basic feature phones that consume hardly any data, but even smartphones have a tough time consuming more than 1 GB a month without a voracious video app. These data are with different format, heterogeneous sources, and various applications [2]. Thus, it is very important to process the data before different purposes.

The process of the data includes several steps such as data collection, cleansing, compression, interpretation, and ultimate application [3]. Within these steps, data compression is very important. First of all, the datasets may be so huge that it is difficult to find the decent information or knowledge from them. Secondly, the hardware for storing the datasets may be limited. Thus, it is necessary to carry out compression procedure before keeping the useful data into the datasets [4]. Thirdly, the data may carry lots of information and knowledge that the density plays critical role in organizing the data effectively and efficiently. As a result, suitable compression is needed for the purpose.

In order to use the suitable compression methods on reducing the data volume, great myriads of research and practices work have been carried out [5-7]. In Astrophysics, a (biased) view of using the statistical tools to achieve data compression of data [8]. Data cubes are widely used for organizing the large amount of data [9]. Iceberg-cube is a method to compress the data [1, 10]. An iceberg-cube is a sub-set of datasets. This method could be easily realized in the relation database.

This paper uses iceberg-cubes to compress the data based on the entropy query mechanism. The data volume could be compressed by the iceberg-cubes. However, the database files only keep the data elements partially. It is difficult to give response to the global query from the original data cubes. In the new algorithm, using the entropy query in iceberg-cube, the compression efficiency is achieved together with the query effectiveness availability and feasibility.

ISSN: 1738-9968 IJHIT

2. Problem Description

Let cube CQ_T with entropy coverage denotes a table T. A threshold will be defined. CQ_T thus, will be divided into two subsets: S_1 and S_2 . $S_1 = \{u_b \mid u_b \in CQ_T, u_b[M] \le threshold\}$, $S_2 = \{u_b \mid u_b \in CQ_T, u_b[M] \ge threshold\}$. When there is no upper boundary which meets $u_b[M] = threshold$ in CQ_T . $S_1 \cap S_2 = \varnothing$. Once it exists, $S_1 \cap S_2 = \{u_b \mid u_b \in CQ_T, u_b[M] = threshold\}$. Given the data cube compression problem, we have to propose several definitions:

Definition 1. The down iceberg-cube from a table T is defined as $D_{CQ_T} = \{u_b \mid u_b \in CQ_T, u_b[M] \le threshold\}$, the upper iceberg-cube is $U_{CQ_T} = \{u_b \mid u_b \in CQ_T, u_b[M] \ge threshold\}$. The down and upper iceberg-cubes from a table are the basic cube for data. There is a function f which is used for generating the data cubes D_T . f has several dimensions: $M, c_1, c_2 \in D_T$.

Definition 2. If $c_1 < c_2$, then $c_1[M] \le c_2[M]$, the f is termed increasing function. That means c_2 has the smaller coverage set of basic elements from T then c_1 . If $c_1 < c_2$, then $c_1[M] \ge c_2[M]$, the f is termed decreasing function.

Definition 3. The basic table $T(A_1, A_2, ...A_n, M)$ has the entropy coverage cube $CQ_T(A_1, A_2, ...A_n, f(M))$. The iceberg-cube $ICQ_T \subseteq CQ_T$. If $q = u_b \in ICQ_T$, $q = UB_q$. That means q is the upper grid of D_T .

Using the above definitions, the following tables could be calculated the entropy coverage cube (ECC) CQ_T .

Table 1. ECC Calculation Example

D_1	D_1	D_1	MIN(M)	SUM(M)
*	*	*	3	18
*	*	d_2	3	9
*	b	*	6	15
t_{or}	b	d_2	6	6
v_{an}	*	*	3	12
v_{an}	b	d_1	9	9
v_{an}	b	d_2	3	3

From Table 1, the iceberg cover quotient cube (iceberg-cube) could be got as follows MIN (M)<3):

Table 2. Iceberg-Cube

D_1	D_1	D_1	SUM(M)
*	*	*	3
*	*	d_2	3
v_{an}	*	*	3
v_{an}	b	d_2	3

In the entropy query coverage cube using the function f, there is a partial ordering relation of the functional value and grid if the filter conditions are $\geq threshold$ or $\leq threshold$. Thus, the filtered grid set and reserved grid set has the partial ordering relation.

3. Proposed Algorithm

The proposed algorithm using entropy query mechanism to compress the iceberg-cubes is based on some lemmas, which are reported as follows.

Lemma 1. For the down iceberg-cube $D_{CQ_T} = \theta_{f(M)\leq threshold} CQ_T$, f is the increasing function, the entropy query q is less than u_b if $u_b \in D_{CQ_T}$. There is no existing of $u'_b \in D_{CQ_T}$ s.t. $u'_b < u_b$. Thus, $u_b = UB_q$ is observed.

PROOF. (Apagoge). Assume $u_b \neq UB_q$.

$\because\ u_b \in D_{CQ_T}\ \therefore\ u_b[f(M)] \leq threshold$

$\because\ q < u_b\ \therefore\ q[f(M)] \leq u_b[f(M)]$

Then $q[f(M)] \leq threshold$

$\therefore\ UB_q[f(M)] \leq threshold$

$\therefore\ UB_q \in D_{CQ_T}$ end.

Lemma 2. If $u_b \in ICQ_T$ does not exist, s.t. $q = u_b$ or $q < u_b$, then, $UB_q \notin ICQ_T$.

Both lemmas are suitable for down and upper iceberg-cube. Suing them, it is possible to figure out whether UB_q is in ICQ_T or not under different situations. For the entropy query of each point, the compression of the data cube is based on the entropy query coverage cube that is able to reflect the same results before the compression. Each compressed grid could present a certain coverage of grid from the original data [11]. And the presence is a unique mapping over the iceberg-cube.

Algorithm 1. Entropy query-based Iceberg-cube Initial

Input: Entropy Point $q = (..., eq_1, ...eq_i, ...eq_n)$, iceberg-cube ICQ_T, Type of ICQ_T, the type of function f when generating ICQ_T of the dimensions. // ICQ_T has been sequenced given the Key value increased.

Output: Grid q of the compressed attribute value

Steps:

1. If f is increasing function and the ICQ_T is down iceberg-cube, or f is the decreasing function and ICQ_T is upper iceburg-cube.
2. Return $F_1(q, ICQ_T)$;
3. Else reture FALSE; // the entropy query cannot be established.
4. Function $F_1(q, ICQ_T)$

{

5. Open ICQ_T;
6. $u_b =$ the first cell in ICQ_T;
7. $v = false$;
8. While not eof (ICQ_T)

{

If ($q = u_b$ or $q < u_b$)

$v = u_b.Ag.Value$ // the value of compressed results.

$Cube^{Com} = \{v \mid v_i \in ICQ_T\}$;

else $u_b =$ the next tuple of ICQ_T

}

9. Close ICQ_T;
10. Return $Cube^{Com}$;

}

This algorithm firstly opens the iceberg coverage cubes data file. The pointer is set to the first tuple (grid). From the definition 1, if $u_b = q$, u_b is the upper boundary of q. The compressed value should be aggregated from the entropy calculation. Since the ICQ_T is sequenced by the partial ordering relation, each tuple from ICQ_T could be reflected from the compressed cube $Cube^{Com}$.

Algorithm 2. Compressed value using entropy query.

Input: query q, iceberg-cube ICQ_T, entropy E_T of ICQ_T,array of dimension value

Output: Compressed aggregation value of UB_q.

Steps:

1. open ICQ_T, calculate E_T;
2. calculate the ordering key sk_q of q;
3. query the E_T of binary b in sk_q-th;
4. if (b==0)
   return false;
   else calculate the sk_b of b;
5. calculate the dimension value of sk_b;
6. generate the array of dimension values;
7. create the basic of each cube dimension $(cd_1, cd_2..., cd_n)$;
8. for ($j = 1; j \le n-1; j++$)
   {
   $Ag.value = sk / ((cd_{j+1}+1)\times(cd_{j+2}+1)\times...\times(cd_n+1))$;
   $sk = sk - Ag.value * (cd_j + 1)$
   $v_j = sk \times E_T \sum_{i=1}^{n} UB_{q_{ij}} \sqrt{sk / Ag.value} \prod_{i=1}^{n-1} cd_j$
   }
9. Return $(v_1, v_2, ..., v_n)$;
10. Merge $(v_1, v_2, ..., v_n)$ to the iceberg-cube compressed results;

The proposed algorithms use the calculation of entropy coverage cube in order to generate the compressed iceberg-cubes. It is using the query operation for this purpose. The principle is simple and feasible [12]. However, the efficiency has the space for further improvement. According to the value of threshold, the entropy query-based cube is able to get the sub-set.

4. Experiments and Discussion

The experiment is carried out by using the data from the weather data of U.S. The dataset is DEP85L.DAT, which includes large number of data with many dimensions, such as stationID (7037), longitude (352), solaraltitude (179), latitude (3809), presentweather (101), day (30), changecode (10), hour (8), brightness (2). Integer has been used for coding the data and the COUNT()<3 is set to be the threshold.

The environment of the experiments are as follows. CUP: INTEL Core 2 Duo, 1.86GHz; Memory: DDRII 8GB; Hard disk: (250GB), speed 7200 per minute; Operation System: Mircrosoft Windows 2003 Server Enterprise Edition Service Pack 2; Programming Environment; Microsoft Visual C++ 6.

Two experiments are carried out. Firstly, Eight groups of dataset is used for examining the effectiveness and efficiency of the proposed algorithms. Figure 1 shows the results which compare the original uncompressed dataset and the data cubes compressed by the proposed algorithm. It is observed that in the first group and last group, the compression results are not good. Because of the ordering key is not selected suitably. The fifth group has the best compression results because of the entropy query efficiency. The compression ratio is 47.86% that means it can save the storage space of almost 50%. And the average compression ratio of the proposed algorithm is 32.46%, which is able to save the data cube space by almost one third from the experiments.

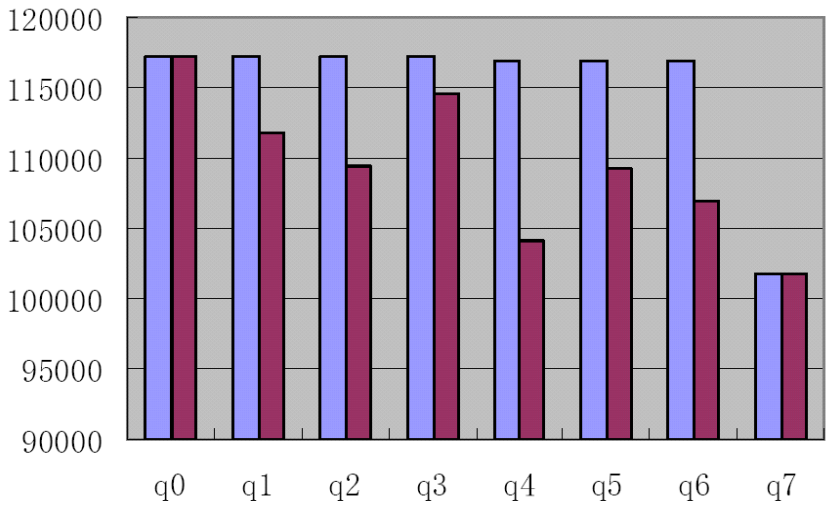

Figure 1. Compression Results Comparing with Original Datasets

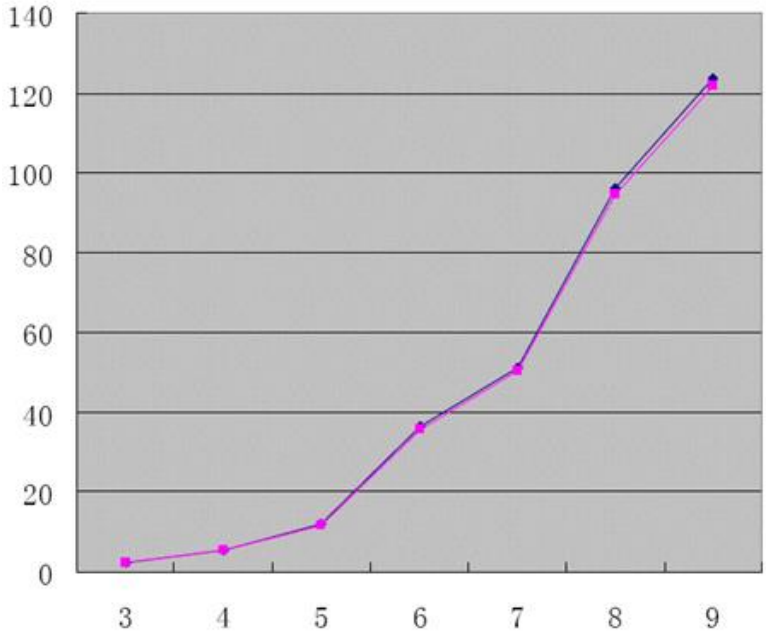

Figure 2. Experiment Results of Presence of Compressed Data Cubes

The second experiment is to examine the presence of the compressed cubes comparing with the uncompressed original datasets. Figure 2 presents the experimental results from comparing the results of uncompressed dataset and data cubes compressed by the algorithms. Different dimensions (0~10) are used for carrying out the experiments. The information unit from 0 to 140 is used for representing the meaning of the dataset. It is observed that the compressed data is able to reflect the information or meaning from the entropy query. With the increasing of dimensions, the accuracy will be decrease a little bit. However, the variations are not very large that the efficiency and feasibility of the proposed algorithm will be guaranteed.

5. Conclusion and Remarks

This paper proposes a new algorithm for data cube compression using the entropy query mechanism for large amount of dataset. Due to the immense of digital devices used in our daily life, great myriad of data are created which should be compressed efficiently and effectively. Using the definition of iceberg-cubes, this paper uses the entropy query principle for getting the key value from the original datasets. The iceberg-cubes are then used for generating the compressed data which should be stored in the hardware devices.

Several contributions are significant from this research. Firstly, several definitions are given to define the problems faced by the data processing from both theoretical and practical aspects. The definitions divide the iceberg-cube into two parts: down and upper given the threshold boundary principle. Secondly, the proposed algorithm uses entropy query method to find the compressed value from a large dataset. Finally, the feasibility and efficiency of the algorithms are examined by carrying out the experiments from two perspectives.

Future research will be carried out from two aspects. First of all, the efficiency could be improved that the calculation loops are time cost due to the large dataset. Parallel calculation method could be used for easing the efficiency. Secondly, a demo system could be established for examining the data compression by using different types of datasets. This paper uses weather data from the experiments, which is not typical in our daily life. More datasets could be used for using other data types like RFID or sensor data.

References

[1] R. Jimenez, "Data Compression Methods in Astrophysics", in Statistical Challenges in Modern Astronomy, Springer, **(2012)**, pp. 309-320.
[2] R. Y. Zhong, Z. Li, A. L. Y. Pang, Y. Pan, T. Qu and G. Q. Huang, "RFID-enabled Real-time Advanced Planning and Scheduling Shell for Production Decision-making", International Journal of Computer Integrated Manufacturing, vol. 26, **(2013)**, pp. 649-662.
[3] R. Y. Zhong, Q. Y. Dai, T. Qu, G. J. Hu and G. Q. Huang, "RFID-enabled Real-time Manufacturing Execution System for Mass-customization Production", Robotics and Computer-Integrated Manufacturing, vol. 29, **(2013)**, pp. 283-292.
[4] R. Kaur and M. M. Goyal, "A Survey on the different text data compression techniques", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2, no. 2, **(2013)**, pp. 711-714.
[5] Q. Y. Dai, R. Y. Zhong, G. Q. Huang, T. Qu, T. Zhang and T. Y. Luo, "Radio frequency identification-enabled real-time manufacturing execution system: a case study in an automotive part manufacturer", International Journal of Computer Integrated Manufacturing, vol. 25, **(2012)**, pp. 51-65.
[6] D. J. López-Araujo, A. Zavala-Río, V. Santibáñez and F. Reyes, "Output-feedback adaptive control for the global regulation of robot manipulators with bounded inputs", International Journal of Control, Automation and Systems, vol. 11, **(2013)**, pp. 105-115.
[7] R. Y. Zhong, Q. Y. Dai, K. Zhou and X. B. Dai, "Design and Implementation of DMES Based on RFID", in 2nd International Conference on Anti-counterfeiting, Security and Identification, **(2008)** August 20-23, pp. 475-477, Guiyang, China.
[8] B. H. Brinkmann, M. R. Bower, K. A. Stengel, G. A. Worrell and M. Stead, "Large-scale electrophysiology: acquisition, compression, encryption, and storage of big data", Journal of neuroscience methods, vol. 180, no. 1, **(2009)**, pp. 185-192.
[9] R. Y. Zhong, G. Q. Huang, S. L. Lan, Q. Y. Dai, C. Xu and T. Zhang, "A Big Data Approach for Logistics Trajectory Discovery from RFID-enabled Production Data", International Journal of Production Economics, vol. 165, **(2015)**, pp. 260-272.
[10] R. Y. Zhong, G. Q. Huang, Q. Y. Dai and T. Zhang, "Mining SOTs and Dispatching Rules from RFID-enabled Real-time Shopfloor Production Data", Journal of Intelligent Manufacturing, vol. 25, **(2013)**.
[11] S. R. Jeffery, M. Garofalakis and M. J. Franklin, "Adaptive cleaning for RFID data streams", in Proceedings of the 32nd international conference on Very large data bases, **(2006)**, pp. 163-174.
[12] R. Y. Zhong, G. Q. Huang, S. Lan, Q. Dai, T. Zhang and C. Xu, "A two-level advanced production planning and scheduling model for RFID-enabled ubiquitous manufacturing", Advanced Engineering Informatics, **(2015)**, doi:10.1016/j.aei.2015.01.002.

Author

Haiyong Luo, He is a senior lecturer from Department of Computer Science, Mr. Luo got the master degree of Software Engineering from East China Normal University. He has been published a large number of articles in international journals and conferences. His main research areas are algorithm design and Software Engineering.

