Entropy-based Selection of Graph Cuboids

Dritan Bleco
Athens University of Economics and Business
76 Patission Street
Athens, Greece
dritanbleco@aueb.gr

Yannis Kotidis
Athens University of Economics and Business
76 Patission Street
Athens, Greece
kotidis@aueb.gr

ABSTRACT

Emerging applications face the need to store and analyze interconnected data that are naturally depicted as graphs. Recent proposals take the idea of data cubes that have been successfully applied to multidimensional data and extend them to work for interconnected datasets. In our work we revisit the graph cube framework and propose novel mechanisms inspired from information theory in order to help the analyst quickly locate interesting relationships within the rich information contained in the graph cube. The proposed entropy-based filtering of data reveals irregularities and non-uniformity, which are often what the decision maker is looking for. We experimentally validate our techniques and demonstrate that the proposed entropy-based filtering can help eliminate large portions of the respective graph cubes.

ACM Reference format:
Dritan Bleco and Yannis Kotidis. 2017. Entropy-based Selection of Graph Cuboids. In *Proceedings of GRADES'17, Chicago, IL, USA, May 19, 2017,* 6 pages.
DOI: http://dx.doi.org/10.1145/3078447.3078449

1 INTRODUCTION

Graph cubes [4, 9, 14] have been recently proposed to provide a solid foundation that an analyst may build upon, in a manner similar to what the data cube was for OLAP analysis [5–7, 11]. Graph cubes contain an exponential collection of aggregated graphs (cuboids). A decision maker, familiar with the simpler multidimensional framework of data cubes, may be overwhelmed when she tries to navigate not flat records, but rather complex graph cuboids containing aggregated views of graph nodes and relationships.

In order to help the analyst quickly locate interesting relationships in the aggregated graphs, we propose the use of *information entropy*. Our intuition is that the analysts are attracted mainly by data skew rather than data uniformity. Based in this premise we use the information entropy to elevate parts of the graph cube that experience this kind of disorder. As we will show, the entropy-based filtering prunes significant parts of the graph cube and at the same time gives valuable indicators as to where interesting associations exist.

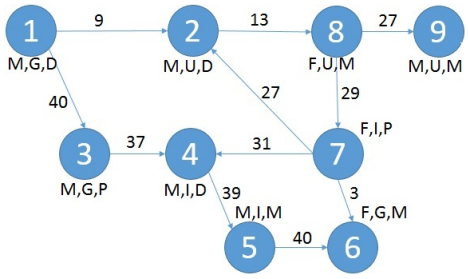


Figure 1: Sample dataset

Our major contributions are summarized as follows:

- We first revisit the graph cube framework highlighting the relationships of the constituent cuboids contained in it. These relationships are modeled as a graph cube lattice produced by taking the Cartesian product of simpler data cubes on the attributes of the nodes and edges of the graph.
- We introduce a measure termed external entropy that captures the entropy of a graph cuboid as a unit. We then show how to utilize this metric in order to decide whether a cuboid provides interesting information with respect to adjacent cuboids in the lattice. We also define the internal entropy in order to help the user navigate within the information contained in a large cuboid. The internal entropy helps the analyst elevate interesting interactions in the graph that become prominent when its raw data is aggregated at the levels denoted by the cuboid.
- We compare our techniques against alternative methods for pruning parts of the graph cube. We observe that our framework maintains the most varied parts of the data distribution resulting in significantly lower entropy values on the remaining parts of the graph cube.

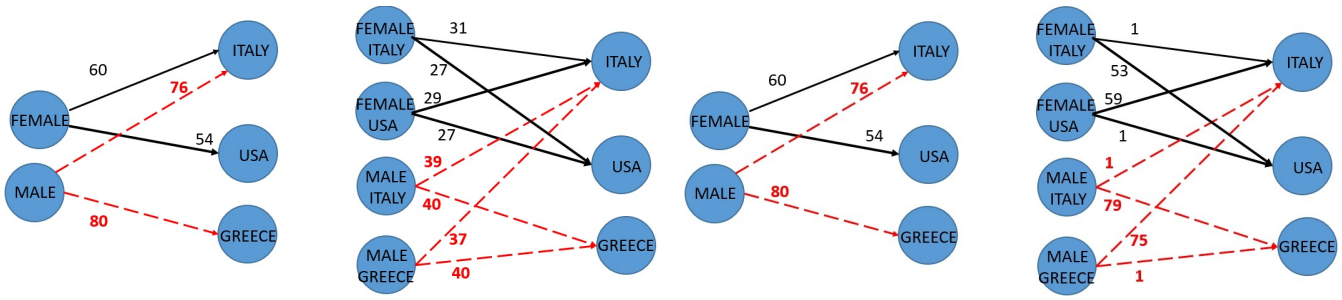
2 MOTIVATION

As a motivating example, we consider a social network which depicts relationships between different users. Each user can be represented as a node in a graph. Nodes may have attributes related to the user such as gender, nation and profession. In Figure 1 we can see a running example for some data produced by the social network. Each profile (node) has three attributes: gender (male, female), nation (Greece, Italy, USA) and profession (doctor, professor, musician). For brevity, we refer to these attributes values by their initial letter. Each edge is associated with a numeric value (weight)

This research is financed by the Research Centre of Athens University of Economics and Business, in the framework of the project entitled 'Original Scientific Publications'.

GRADES'17, Chicago, IL, USA

978-1-4503-5038-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3078447.3078449



(a) Drill down (original data) (b) Drill down (alternative data)

Figure 2: Drill-down from (gender - nation) to (gender,nation - nation) for original and alternative dataset

that in this example indicates the number of interactions between the respective users.

A possible inquiry on the presented network is to examine how users depending on their gender relate to other users based on their nationality. To accommodate this query we need to perform three different aggregations. First, starting nodes (i.e. nodes with outgoing edges) are grouped into two aggregate nodes corresponding to gender values male and female, respectively. Similarly, three aggregate nodes corresponding to nations Greece, Italy and USA are formed. Finally, each edge of the network, depending on the gender attribute value of its starting node and the nation attribute value of its ending node is aggregated into an edge between the corresponding aggregate nodes created in the previous steps. At this time, a desired aggregate function can be computed. In this example, we assume that this function is SUM(). The resulting aggregate graph is depicted in the left-most graph of Figure 2a. Based on its construction we refer to it as the (gender - nation) cuboid.

Continuing the running example, Figure 2a depicts the process of drilling down from (gender - nation) to (gender,nation - nation). The intuition is that we would like to explore whether the nationality of the source node, in addition to its gender, affects the number of depicted relationships. In this contrived example, the aggregated edges from cuboid (gender - nation) are split almost evenly when drilling down to the (gender,nation - nation) cuboid. Thus, this particular drill-down on the graph cuboids does not seem to reveal interesting correlations for this dataset.

In Figure 2b we depict another example of this process for an alternative dataset. In contrast with the first case, here we can find some irregularities in the data. These non-uniformities reveal certain trends, such as that females from Italy are linked mainly to users from USA, while males from Greece are related to users from Italy.

Because of the exponential number of cuboids in the graph cube, it is extremely difficult for an analyst to manually explore all possible cuboids and navigation steps among them (roll-up, drill-down) in search for interesting patterns. This realization provides the motivation of our techniques. We seek to provide the analyst with solid mathematical tools derived from information theory and in particular the information entropy, that can help her reveal such interesting irregularities.

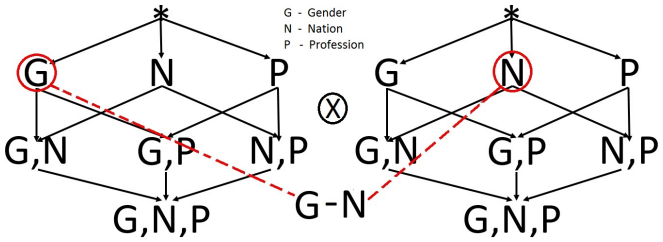


Figure 3: The Graph Cube

3 THE GRAPH CUBE

The graph cube is the Cartesian product of two cubes: of the starting- and the ending-cube, as is depicted in Figure 3. In this example a graph cuboid can be ((gender, *,*) - (*,nation,*)) or, for brevity, (gender - nation). The starting nodes on this cuboid are aggregated graph nodes based on the gender attribute. Similarly, the ending nodes are aggregations of raw graph nodes based on the nation attribute. Starting and ending nodes in this cuboid are interconnected according to the raw graph edges. These raw data edges are aggregated producing a graph cube edge along with a measure. The user can choose any combination of functions based on measure attributes on the constituent nodes and edges. For simplicity, we assume that this function is the SUM() function along a single numerical measure on the edges, in our running examples.

Clearly, the graph cube is significantly more complex than the data cube on the plain node attributes. The number of cuboids increases from 2^n (data cube) to $2^{(2n)}$, in the graph cube. Moreover, each of these cuboids, is not a flat relation, but an aggregated graph, as is depicted in Figures 2a and 2b.

The graph cube framework can be extended by considering attributes on the edges of the data graph that can be used as another set of dimensions in the analysis. Since this extension is orthogonal to the techniques we present next, for brevity in the discussion and ease of notation, we only consider dimensions used in the graph nodes.

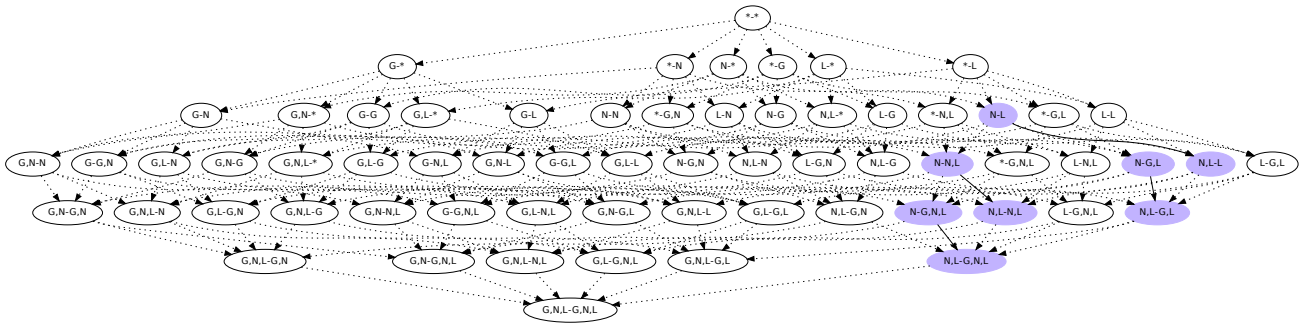


Figure 4: Selecting cuboids based on their external entropy, Twitter dataset

Record		Cardinality
$gender_s$	$nation_e$	
male	Greece	80
male	USA	22
male	Italy	76
female	Italy	60
female	USA	54
female	Greece	3

Table 1: Dual representation of cuboid (gender - nation). The table shows the distinct records and their cardinality

4 ENTROPY CALCULATIONS ON GRAPH CUBOID RECORDS

Let us consider cuboid C_i from the graph cube lattice. The cuboid is a graph but it can be (virtually) flattened into a relation with bag semantics (i.e. a table). Each cuboid edge is mapped to a set of records (rows) with cardinality equal to the aggregate measure computed on the edge and attribute values those of the constituent nodes. For example cuboid (gender - nation) of Figure 2a is virtually represented by a relation with attributes (gender, nation) as is shown in Table 1. The domain of attribute gender is male, female and that of attribute nation Italy, Greece, USA. The cuboid edge (female, Italy) is represented by a set of records of the form (female, Italy). There would be 60 such records in this virtual table. Similarly, there would be 3 records of the form (female, Greece).

We refer to this virtual relation as the dual representation of the cuboid. The number of rows (records) in the dual representation is the size N of the cuboid. In the running example, the size of cuboid (gender - nation) is N=295.

We define the external entropy (eH) of a cuboid as the negative of the logarithm of the probability distribution of the cuboid records in its dual representation. The virtual relation contains N records and consists of m distinct rows with cardinality $a_1, a_2, \ldots, a_m$. E.g. $a_{(female,Italy)}$=60. The external entropy of C_i is calculated as

$$eH(C_i) = -\sum_{j=1}^{m} p(a_j) * \log_2 p(a_j) \quad , where\, p(a_j) = \frac{a_j}{N} \qquad (1)$$

Recall that each cuboid has a certain selection of starting and ending attributes. Lets assume that cuboid C_i consists of $s_1, s_2, \ldots, s_t$ starting attributes and $e_1, e_2, \ldots, e_w$ ending attributes. Thus, each row r_j in the dual representation has tuples of the form $(s_{j_1}, s_{j_2}, \ldots, s_{j_t}, e_{j_1}, e_{j_2}, \ldots, e_{j_w})$. If we add another attribute (starting or ending) in the cuboid C_i then we get another cuboid C_k of the lattice. We refer to cuboid C_k as the "child" of C_i, while C_i is the "parent" of C_k. For each row of C_i there is a more detailed row in C_k.

Let us consider the relationship between the external entropies of these cuboids. Drilling down from the parent C_i to the child C_k we can calculate the delta-entropy, i.e. the difference between the two external entropies as:

$$\delta_{(C_k,C_i)} = eH(C_k) - eH(C_i) \qquad (2)$$

This difference equals to the conditional entropy of the child given the parent.

$$\delta_{(C_k,C_i)} = \sum_{j=1}^{m} p(a_j^i) \times eH(C_k|C_i = a_j^i)$$

$$= -\sum_{j=1}^{m} \{p(a_j^i) \times \sum_{o=1}^{d} p(a_o^k) \times \log_2 p(a_o^k)\} \qquad (3)$$

where $p(a_o^k) = \frac{a_o^k}{a_j^i}$, r_o^k is a more detailed C_k row of (C_i)'s row r_j^i and there are d distinct values for the detailed rows of r_j^i.

The delta entropy is a non-negative number. This is because the external entropy of the child cuboid C_k is greater or equal to the external entropy of its parent C_i. For the minimum and the maximum entropy of C_k it holds that

$$0 \leq eH_{min}(C_i) = eH_{min}(C_k) \leq eH_{max}(C_i) \leq eH_{max}(C_k) \qquad (4)$$

The minimum entropy value of the child equals to the entropy of its parent, i.e. when the delta entropy $\delta_{(C_k,C_i)} = 0$. In this case for each distinct record of the parent (in the dual representation), there is only a single distinct record for the child. The maximum external entropy of the child is obtained when each distinct record of the parent dual representation is distributed uniformly among the more detailed records of the child. For example if the parent

cuboid C_i has m distinct records with cardinality $a_1, a_2, \ldots, a_m$, respectively and for each $a_j, j \in [1, m]$ there are d_j distinct records in the child dual then

$$eH_{max}(C_k) = -\sum_{j=1}^{m} p(a_j) * \log_2 \frac{p(a_j)}{d_j} \quad (5)$$

Based on these observations, we introduce the *delta entropy rate* in order to quantify how informative the process of drilling down from parent C_i to its child C_k is. We define the *external entropy rate* as

$$eH_{rate}(C_k, C_i) = \frac{eH(C_k) - eH(C_i)}{eH_{max}(C_k) - eH(C_i)} \quad (6)$$

Where $0 \leq eH_{rate}(C_k, C_i) \leq 1$. When this value is close to 1, the drill-down process doesn't change significantly the distribution of the records and, thus, no new insights are given to the analyst. We can therefore exclude less interesting navigations in the lattice by defining an external entropy rate threshold value between zero and one. When the eH_{rate} of a drill down surpasses the threshold, then this drill down is omitted from consideration.

4.1 Internal Entropy

In order to gain insight into the distribution of records within a cuboid, we introduce an additional type of entropy termed internal entropy. Due to the fact that we consider directed data graphs, we distinguish between two kinds of internal entropies namely starting internal entropy and ending internal entropy.

Consider cuboid C_i with N records, s starting attributes and t ending attributes. Furthermore, there are l distinct combinations of starting attribute values of the form $(a_1^y, a_2^y, \ldots, a_s^y) : m_y$, where $y \in [1, l]$ and m_y is their cardinality in the cuboid. For each such combination (indicated by parameter y) there are f_y different combinations of ending attribute values with cardinality z_{q_y}. We calculate the starting internal entropy as the conditional entropy of the ending attributes' values conditioned from each starting attribute combination of values. Thus, for the combination of starting attribute values indicated by y, we define the starting internal entropy as

$$siH(C_i^y) = -\sum_{j=1}^{f_y} p(q_j^y) * \log_2 p(q_j^y) \quad where\, p(q_j^y) = \frac{z_{q_y}}{m_y} \quad (7)$$

The ending internal entropy eiH is defined in an analogous manner. As in the case of external entropy, we introduce the internal entropy rate (for the starting or ending internal entropy, respectively) as the fraction between the (starting/ending) internal entropy and the maximum possible value of internal entropy. For example the starting internal entropy rate is defined as

$$siH_{rate}(C_i^y) = \frac{siH(C_i^y)}{siH_{max}(C_i^y)} \quad (8)$$

The value of the internal entropy rate is between 0 and 1 and can be used to select the most prominent trends within a cuboid.

5 EXPERIMENTS

In this section we provide an experimental evaluation of the proposed framework. We worked with three real social datasets. The first one consists of data sampled from Twitter. The second dataset is from VKontakte (VK). VK is the largest European on-line social networking service. It is available in several languages, but is especially popular among Russian-speaking users. The last dataset is from Pokec, the most popular on-line social network in Slovakia. Pokec has operated for more than 10 years and connects more than 1.6 million people. This dataset contains anonymized data of the whole network. The first two datasets were crawled by our team while the Pokec dataset is available at [8].

The characteristics of these datasets are shown in Table 2. The Twitter dataset contains 3 attributes on the nodes (profiles): the gender, location and language used from the profile. The VK dataset contains 5 attributes: birthyear, country, city, gender and education level of the user. Finally, the Pokec dataset uses 6 node attributes: age, region, gender, registration year, public profile and completion percentage of the profile.

In order to compute the graph cubes of these datasets, we set up a small cluster of 4 PCs equipped with Intel i7-3770 CPUs clocked at 3.40GHz, 4GB of memory and 300GB 7200rpm HDDs. We used the popular Apache Spark [13] framework on 8 VMs (one being the master) running on this cluster. At a pre-processing step we computed all cuboids for all three datasets using the BUC algorithm [1] that we adapted for graph cubes. Given a data cube lattice, the BUC algorithm initiates a recursive computation of the cuboids by performing a bottom-up depth-first-search traversal of the lattice. In the case of data graphs, the graph cube lattice is a cross-product of lattices and this implies that the algorithm may proceed in two directions (starting/ending node aggregation) its recursive computation, exploiting the parallelism provided by Spark. In Figure 5 we depict the flow of the algorithm for a simpler graph cube lattice on two attributes gender (G) and nation (N). In this example, after the (gender - *) cuboid is computed, the modified BUC algorithm may proceed and compute in parallel cuboids (gender,nation - *) and (gender - gender). Thus, the modified BUC utilizes a number of parallel DFS processes. The numbers depicted on the edges of the lattice in the figure indicate the relative order of computation.

Other algorithms for computing data cubes can also be extended for the graph cube. This selection is orthogonal to our techniques. We used BUC because we also wanted to evaluate the pruning obtained by our methods against computing an Iceberg cube, which as will be explained, is computed using BUC.

All presented experiments in what follows were executed on a small desktop PC running a commercial database server. The desktop PCs is equipped with Intel i5-4200 CPUs clocked at 2.30GHz, 8GB of memory and two 2TB 7200rpm HDDs We loaded the graph cubes obtained by the Spark program in the database and computed the internal and external entropy calculations within the SQL engine.

In Figure 6a we plot the percentage of records that are retained in the graph cube (y-axis) for all the datasets when we vary the threshold for the external entropy rate (x-axis). The absolute sizes of the corresponding cubes are presented in Table 2. A small value of the external entropy rate threshold filters out a large portion of the

	Twitter	VK	Pokec
Profiles (nodes)	34,062,759	3,917,224	1,632,803
Relations (edges)	910,526,369	493,137,167	30,622,564
Number of Attributes	3	5	6
Number of Cuboids	64	1024	4096
Graph Cube Records	4,010,022	362,149,881	66,352,625,425
Graph Cube Size	18 GB	235 GB	1.58 TB

Table 2: Description of datasets used

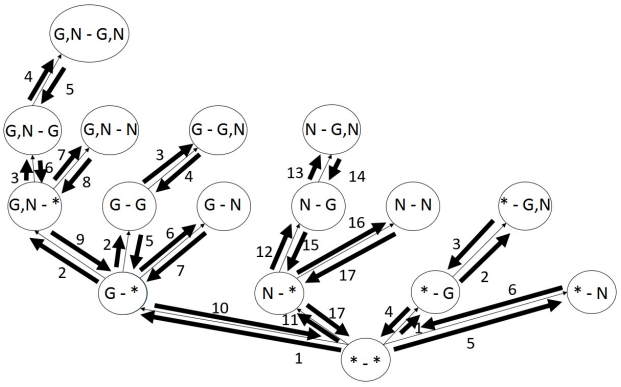


Figure 5: The BUC algorithm adapted for graph cubes

graph cuboids. These cuboids provide no significant information with respect to their ancestors and descendants in the lattice.

The figure reveals a steep reduction in the graph cube sizes, when we decrease this threshold bellow a certain value. For the Twitter dataset only 14% of the cube remains for a threshold rate of 3,5%. Moving up this threshold to 4%, the percentage jumps to 50% of the Twitter graph cube. This suggests that there is skew in the distribution of values across cuboids that we can investigate further using the internal entropy rates (discussed next). On the other hand, an increase of the external entropy rate threshold beyond 4% overwhelms the user with a significant increase in the result set, as many near-uniform relationships are retained complicating further analysis.

The same phenomenon arises in the VK dataset but is less profound. Still, with a threshold of 10% for the external rate we are left only with 17% of the VK cube records. The Pokec dataset exhibits the same behavior. With a threshold of 9% there are only 13% of graph cube records remaining for analysis.

Figures 6b and 6c illustrate the percentage of records of the graph cubes retained for the three datasets, scaling the starting and ending internal entropy rates, respectively. Similar observations can be made for the pruning power of the internal entropy. For a starting internal entropy rate threshold of 10% we are left with just 0.7% of the Twitter graph cube, 0.0026% of the VK graph cube and 0.0019% of the Pokec graph cube records. For a more relaxed 40% threshold there are only 26% of Twitter, 5% of VK and 3% of Pokec graph cube recods filtered in. In conclusion, only a small percentage of the billions of records in these graph cubes reveal interconnections that are far from uniformity.

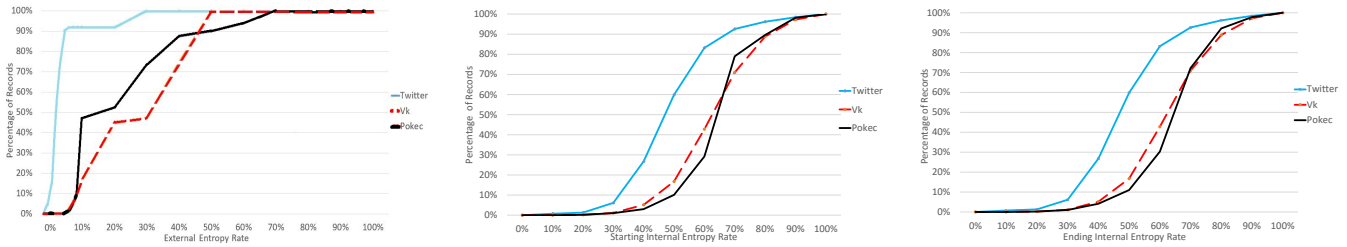
An alternative method for filtering out records from the graph cube is to use a minimum support threshold. Using such a threshold, we may omit aggregate records (relationships) that are generated from fewer than the required number of "base" graph records. This idea has been used under the name of Iceberg cubes [1].

In the following experiments we compute the Iceberg graph cube, for different values of minimum support and then, we adjust the internal entropy rate threshold so as to retain the same number of graph cube records. We then compare the resulting subsets of the graph cube in terms of the entropy retained in them. Recall that a smaller value of entropy implies that more skew is evident in the dataset. In Figure 7a we show the differences between the produced graph cubes for the Twitter dataset. The x-axis in the figure is the number of records in the produced graph cubes. For the same output size the internal entropy rate used by our techniques for selecting pieces of the graph cube results in a dataset with more skew or, equivalently, less random behavior. Similar observations hold for the other two datasets, as is depicted in Figures 7b and 7c. In some instances, the cube retained by our method holds three orders of magnitude smaller entropy values than an Iceberg cube of the same size.

6 RELATED WORK

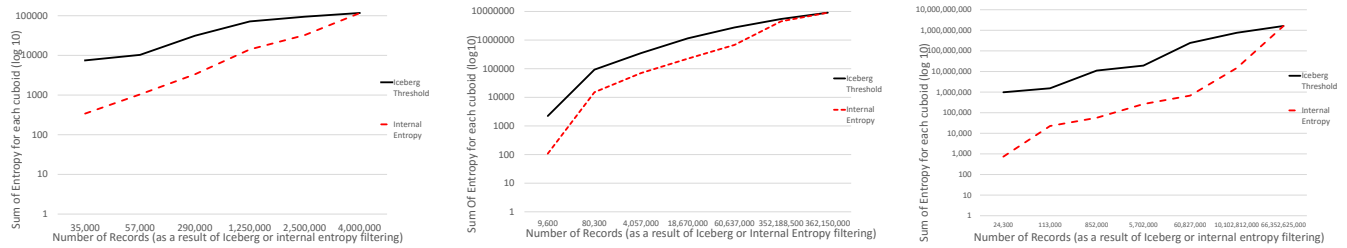
The work in [14] introduced the graph cube that takes into account both attribute aggregation and structure summarization of the underlying graphs. This work is mainly focused on cuboids that aggregate the starting and ending nodes on the same dimensions, e.g. (nation - nation). More general aggregations that differentiate between the starting and ending nodes of the graph are not specifically mentioned but can be addressed under a cross-cuboid computation that is mentioned as an extension. In our work we elevate such cuboids as first-class-citizens in the graph cube framework. As our experiments with real datasets indicate, such cuboids often hold significant insights for the underlying interconnections. Furthermore, the work of [14] considers all records in the proposed graph cube. As we show in our work, only a small part of a complex graph cube carries interesting information when analyzed under the lens of our entropy-based navigation framework.

A recent work [12] considers aggregate attributed graphs. The authors name their model as a hyper graph cube while its computation is done with map-reduce batches. The hyper graph cubes aggregate separately attributes at vertices and edges and then calculate the Cartesian product between them. Thus, they do not exploit and analyze the existing relationships under different levels of aggregation on the starting and ending nodes of the graph.



(a) Scaling external entropy rate (b) Scaling starting internal entropy rate (c) Scaling ending internal entropy rate

Figure 6: Records remaining in the graph cube using proposed entropy rates



(a) Twitter dataset (b) VK dataset (c) Pokec dataset

Figure 7: Total entropy (log 10) for the same number of output records from the internal and Iceberg threshold, respectively

Our solutions are based on the foundational model of information entropy and, as our experiments demonstrated, help steer the analyst towards the most important parts of the aggregated graph dataset. A common trend that we observed in all three real datasets that we used in our evaluation study is that only a small fraction of the aggregate graph data shows significant skew. As a result, even on graph cubes containing tens of billions of records (e.g. as in the Pokec dataset), we can prune the majority of the records based on the computed entropy model (internal and external entropies). To the best of our knowledge we are the first that utilize the entropy in order to filter the information that a graph cube holds. Recently, an entropy-based model has been proposed [10] in order to estimate the strength of social connections by analyzing users' occurrences in space and time. This work considers triplets of (user, location, time) data and utilizes entropy to measure the diversity of user co-occurrences. In our work we utilize entropy to measure the diversity within and across graph cuboids. The works of [2, 3] consider the case of analyzing very large collections of smaller data graphs, while in this work we consider a singe massive graph that is under investigation.

7 CONCLUSIONS

In this work we proposed intuitive measures derived from information theory in order to select interesting substructures from graph cubes computed via aggregation a raw data graph over its node attributes. Our experimental results validate the effectiveness of our techniques on real datasets of realistic sizes. As a future direction we plan to explore ways to prune some of the required entropy computations based on the parent-child relationships that we have identified between the cuboids.

REFERENCES

[1] K. Beyer and R. Ramakrishnan. 1999. Bottom-up Computation of Sparse and Iceberg CUBE. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD '99)*. 359–370.
[2] D. Bleco and Y. Kotidis. 2012. Business Intelligence on Complex Graph Data. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany*. 13–20.
[3] D. Bleco and Y. Kotidis. 2014. Graph Analytics on Massive Collections of Small Graphs. In *Proceedings of the EDBT, Athens, Greece*. 523–534.
[4] A. Ghrab, O. Romero, S. Skhiri, A. A. Vaisman, and E. Zimányi. 2015. A Framework for Building OLAP Cubes on Graphs. In *Advances in Databases and Information Systems, Poitiers, France, September 8-11, 2015, Proceedings*. 92–105.
[5] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. 1996. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. In *ICDE*. 152–159.
[6] W. H. Inmon. 1992. *Building the Data Warehouse*. QED Information Sciences, Inc., Wellesley, MA, USA.
[7] Y. Kotidis and N. Roussopoulos. 2001. A Case for Dynamic View Management. *ACM Transactions on Database Systems* 26, 4 (2001).
[8] J. Leskovec and A. Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).
[9] X. Li, J. Han, and G. Hector. 2004. High-dimensional OLAP: A Minimal Cubing Approach. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30 (VLDB '04)*. 528–539.
[10] H. Pham, C. Shahabi, and Y. Liu. 2013. EBM: An Entropy-Based Model to Infer Social Strength from Spatiotemporal Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, NY,2013*. 265–276.
[11] N. Roussopoulos, Y. Kotidis, and M. Roussopoulos. 1997. Cubetree: Organization of and Bulk Incremental Updates on the Data Cube. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*. 89–99.
[12] Z. Wang, Q. Fan, H. Wang, K.-Lee Tan, D. Agrawal, and A. El Abbadi. 2014. Pagrol: Parallel graph olap over large-scale attributed graphs. In *IEEE 30th International Conference on Data Engineering, Chicago, IL, USA, 2014*. 496–507.
[13] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. 2010. Spark: Cluster Computing with Working Sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'10)*. 10–10.
[14] P. Zhao, X. Li, D. Xin, and J. Han. 2011. Graph Cube: On Warehousing and OLAP Multidimensional Networks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '11)*. 853–864.