Finding the Needle in a Haystack: Entropy Guided Exploration of Very Large Graph Cubes*

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

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

ABSTRACT

Graphs provide an elegant and versatile solution for modeling complex datasets, especially when the focus of the analysis is on highlighting interesting associations between data entities. Graph cubes permit analysis of the resulting data graphs at various levels of granularity based on their node and edge attributes. In this work, we utilize information entropy measures in order to help the analyst navigate within the rich information contained in a graph cube. Our metrics suggest navigations (drill-downs) towards more detailed data descriptions, conditioned on what has been observed at a coarser resolution. We propose a graph analysis workflow that first suggests interesting cuboids from the exponential collection of aggregations that exist in the graph cube. At a latter step, this workflow handpicks sub-graphs out of these aggregations that deviate significantly from the rest of the data. We experimentally validate our techniques using real datasets and demonstrate that the proposed entropy-based exploration can help eliminate large portions of the respective graph cubes from consideration. Our techniques help locate the "needle in the haystack" and steer the user towards data skew hidden within vast valleys of near-uniform interactions.

1 INTRODUCTION

Despite their versatility, graph data have specific characteristics that make their analysis often challenging. Of particular interest in graph data are the relationships between nodes captured by the edges of the graph. These relationships should be analyzed with respect to attribute values available at the nodes and edges. For example, a data scientist may want to investigate how users of a social network, depending on their gender, relate to other users based on their nationality. This inquiry can be accommodated by aggregating existing relationships (edges) in the data graph based on the attributes of their constituent nodes. This process forms a *graph cuboid*, as is depicted in Figure 1.

The graph cube contains all such possible cuboids that can be generated given the raw graph data [6, 10, 15, 22, 35]. As in the case of the data cube [11, 12, 16, 28], there is an exponential number of aggregations that define the space of all possible such cuboids. Moreover, each of these cuboids is not a flat relation, but a complex property graph filled with intrinsic structural information based on the formed relationships and annotated with computed summary statistics over the attributes of the graph nodes and edges. A data explorer, familiar with the simpler multidimensional framework of data cubes, may be overwhelmed when she tries to navigate this data deluge.

*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

In this work, we model the relationships between the graph cuboids 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 data graph. Using this model, we propose a graph cube analysis workflow that can be used to explore interesting associations hidden within very large graph cubes. Our suggested workflow utilizes two intuitive entropy measures, introduced in [5], in order to reveal associations that deviate from the expected behavior. The first measure termed as *external entropy* permit us to suggest certain drill-down navigations that reveal associations that deviate from what has already been observed at the higher-level aggregations of the graph cube. As demonstrated by our experiments, from the exponential possible navigations in the graph cube, only a very small percentage of them leads to interesting observations. The external entropy helps the data explorer navigates towards interesting cuboids in the graph cube lattice and may be used to prune a significant portion of the lattice from consideration.

In a second step of the workflow, we utilize entropy calculations in order to elevate particular data associations that deviate from the rest of the relationships within the cuboids selected from the first step. This is achieved by using an *internal entropy* metric that helps the analyst elevate aggregate interactions that are the result of skew in the data graph. These interactions become prominent when the raw data is aggregated at the levels denoted by the cuboid under investigation.

In our experimental section we present results of utilizing our techniques while processing real social datasets of realistic sizes. We compare our techniques against an alternative method that prunes parts of the graph cube based on a minimum support threshold, as in association rule mining. We observe that our framework maintains the most varied parts of the data distribution independently of their frequencies. Thus, many interesting trends revealed by our technique that focuses on data skew within and across cuboids, would be missed by methods that merely seek frequent patterns. We also discuss prominent trends reveled by our techniques on the real datasets used.

2 MOTIVATING EXAMPLE

We consider a social network which depicts relationships between different users. Each user can be represented as a node in a graph. Each user profile has three attributes: gender (male, female), nation (Greece, Spain, France) and profession (doctor, professor, musician). For brevity, we refer to these attributes values by their initial letter. Each edge in the data graph is associated with a numeric value that indicates the number of interactions between the respective users.

A possible inquiry on this 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

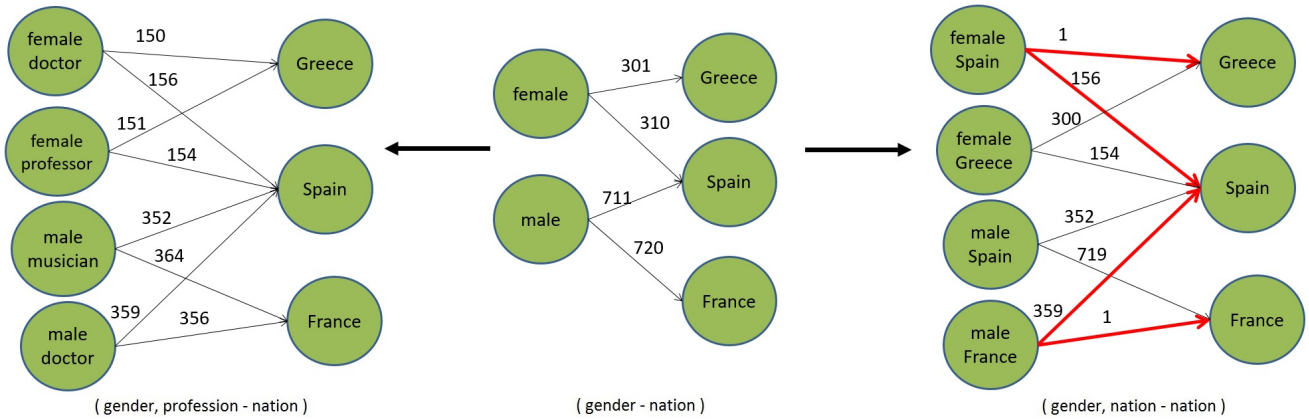


Figure 1: Three possible cuboids: (gender, profession - nation), (gender - nation) and (gender, nation - nation). Notice that the drill-down to the more fine-grained cuboid on the right reveals irregular associations, conditioned to what has been revealed by the cuboid in the middle. In contrast, the relationships contained on the (gender, profession-nation) cuboid seem to follow the same patterns as the original top-level cuboid.

to gender values male and female, respectively. Similarly, three aggregate nodes corresponding to nations Greece, Spain and France 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 at 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 middle of Figure 1. Based on its construction we refer to it as the (gender - nation) cuboid.

Continuing with the running example, the cuboid on the left part of the figure depicts the outcome of drilling-down from (gender - nation) to the (gender, profession - nation) cuboid. The intuition is that we would like to explore whether the profession of the source node, in addition to its gender, affects the number of observed relationships. In this contrived example, the aggregated edges from cuboid (gender - nation) are split almost evenly when drilling down to the (gender, profession - nation) cuboid. Thus, this particular navigation step does not seem to reveal interesting correlations for this data, conditioned on what is already observed in the (gender - nation) cuboid.

On the right part of Figure 1, we depict another possible drill-down, this time to the (gender, nation - nation) cuboid. In this new context, some interesting irregularities are revealed. First, while female users are linked evenly to users from Greece and Spain, when these links are conditioned based on her nationality we can see that females from Spain are mainly linked to users from the same country. Similarly, French males are mostly linked to users from Spain. Thus, while cuboid (gender - nation) suggest a uniform relationship based on the nationality of the target node, cuboid (gender, nation - nation) reveals that this is not true for certain members of the user community. It is worth noting that the majority of the links in the (gender, nation - nation) cuboid still follow the same uniform pattern suggested by the (gender - nation) cuboid, since most links emanate from female users in Greece and male users in Spain. Thus, the examples discussed above are exceptions to what is suggested by the (gender - nation) cuboid. These are depicted in red color inside the (gender, nation - nation) cuboid.

3 THE GRAPH CUBE

In our running example, each user profile has three attributes, namely gender (G), nation (N) and profession (P). If we treat these attributes as dimensions in OLAP analysis, the resulting data cube has 2^3=8 possible cuboids. The work of [35] extended the data cube framework to work on graph data by considering also the relationships between aggregated graph nodes. In particular, consider a data cube for the data attributes of the starting nodes in the graph and another one for the ending nodes. These data cubes share the same dimensions and are, thus, identical in structure (i.e. contain the same set of cuboids). The graph cube can be considered as the Cartesian product of these two data cubes: of the starting- and the ending-cube. In this running example, a graph cuboid can be ((gender, nation,*) - (*,nation,*)) or, for brevity, (gender, nation - nation). The starting nodes on this cuboid are aggregated graph nodes based on their gender, nation attribute values. Similarly, the ending nodes are aggregations of raw graph nodes based on the nation attribute values. Starting and ending nodes in this cuboid are interconnected according to the raw graph edges. These raw data edges are consolidated producing a graph cube edge along with a measure. The user may choose any combination of functions based on attributes on the constituent nodes and edges.

In many applications, edges of the data graph may have attributes that can also be treated during exploratory analysis as dimensions. Attributes on the edges of the data graph can be aggregated creating yet another set of cuboids in an edge-cube lattice. For example, in a social network a connection can have several attributes like the type T of the relationship (family, friend, sibling etc.) and the date D that this connection was established. Naturally the analyst may want to include those attributes and observe their interaction with the node attributes. As an example, let us consider the case where the data graph edges have a Type (T) and a Date (D) dimension (the latter being rolled-up in a suitable level, e.g. day, year or month). The edge-cube lattice in this example contains four cuboids, namely (*), (T), (D), and (T,D). These cuboids can also participate in the Cartesian product of the graph cube computation adding another dimension in the final cube. A cuboid in this extended cube is denoted as (starting node-aggregation - edge-aggregation - ending-node-aggregation).

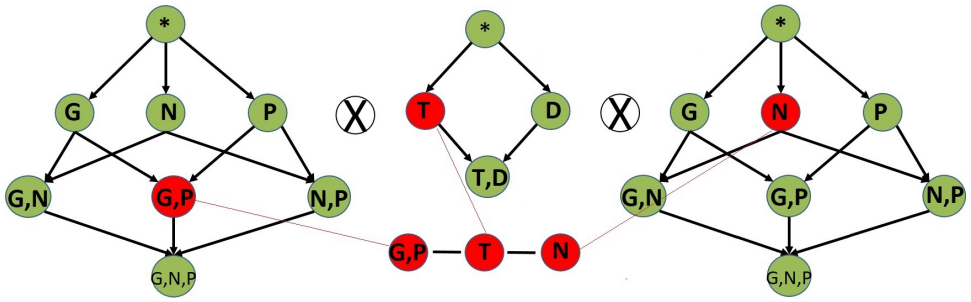


Figure 2: The graph cube when both node attributes (side data cubes) and edge attributes (middle data cube) are being used. The graph cube lattice is produced by taking the Cartesian product of the three data cube lattices that form the constituent data cubes.

Figure 2 depicts the graph cube lattice in this extended example where both node and edge attribute values are being used in the analysis. In what follows, for simplicity, we will only refer to examples where attributes on the nodes are being used when forming the graph cube. However, our techniques also work when attributes on the edges take also part in the analysis.

4 USING ENTROPY TO NAVIGATE THE GRAPH CUBE

4.1 Main concepts

In this work, we present techniques that help the analyst identify irregularities when navigating different aggregations of the original data graph. Because of the exponential number of cuboids in the graph cube, it is extremely difficult to manually explore all possible cuboids and all navigation steps among them (roll-up, drill-down) in search for interesting patterns. This realization provides the motivation for our framework. We seek to provide the analyst with solid mathematical tools derived from information theory and in particular the information entropy, that will help her reveal interesting irregularities.

In [5] we introduced two types of entropy calculations. The first one measures the significance of a whole cuboid and it is called *external entropy*. This type of entropy is used to detect whether a drill-down process during exploratory analysis to a more detailed cuboid provides additional insights or not. In our running example, external entropy calculations on the (gender, profession - nation) and (gender - nation) cuboid will suggest that no apparent irregularities are revealed by this drill-down and it can, thus, be omitted. In contrast, the external entropy metric will suggest that the drill down to the (gender, nation - nation) cuboid reveals certain skew in the calculated relationships that deviate from what is expected by observing the relationships in the (gender - nation) cuboid. The second type is the *internal entropy* that evaluates the relationships inside a cuboid. Internal entropy can help steer the user towards surprising, skewed relationships (such as those depicted in red in the figure) within a large cuboid, eliminating relationships that do not reveal trends that deviate from the expected behavior.

In what follows, we first introduce the suggested entropy calculations used in our navigation framework. More details on these metrics can be found in [5]. We discuss a graph cube analysis workflow that can be used for processing very large graph cubes.

4.2 External Entropy Metric

The edges from a cuboid C_i can be represented as a virtual relation. Each record in this virtual relation is associated with (i) a set of attribute values $s_1, \ldots s_t$ derived from the starting nodes of the corresponding edge, (ii) a set of values $e_1, \ldots e_w$ derived from the ending nodes and (iii) an aggregate value a that denotes the result of the selected aggregate function applied over the selected measures from these constituent nodes and edges. In the example of Figure 1, edge (female, Spain) of cuboid (gender - nation) will be mapped to a single row (female, Spain, 310) in the virtual table. Each such record r_j=($s_1, \ldots s_t, e_1, \ldots e_w, a$) can be viewed as a discrete probability distribution $P(s_1, \ldots s_t, e_1, \ldots e_w)$ by normalizing the aggregate a value on each record by the sum of all aggregate values in the instance of the relation. Thus, record r_j is associated with a probability value $p(a_j) = \frac{a_j}{\sum_{i=1}^{m}(a_i)}$. In our example, the probability value for the record that maps to edge (female,Spain) will be $\frac{310}{301+310+711+720}$. The external entropy (eH) of a cuboid is defined as the negative of the logarithm of the probability distribution of the records in the virtual relation (m in the formula bellow refers to the number of edges in the cuboid that also equals the number of records in the virtual table).

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

A drill-down process in the graph cube lattice is triggered by adding another attribute (starting or ending) in cuboid C_i. This leads the analyst to another *more detailed* cuboid C_k an the next level of the lattice. We refer to cuboid C_k as the "child" of C_i, while C_i is the "parent" of C_k. While 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:

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

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. The maximum external entropy of the child is obtained when the aggregate a of each edge is distributed evenly among the more detailed edges in C_k and their number is maximized. Let d_{max} denote the number of possible values of the attribute on which the drill down process was performed. In order to maximize the entropy of a child cuboid, an edge with aggregate value a_j^i in C_i is replaced during the drill-down with d_{max} more detailed edges in C_k with aggregate values a_o^k=$\frac{a_j^i}{d_{max}}$. Thus, the maximum possible external entropy value

of the child cuboid given its parent is

$$eH^i_{max}(C_k) = -\sum_{j=1}^{m} p(a^i_j) * \log_2 \frac{p(a^i_j)}{d_{max}} \quad (3)$$

The *external entropy rate* quantifies how informative, the process of drilling down from parent C_i to its child C_k is:

$$eH_{rate}(C_k, C_i) = \frac{eH(C_k) - eH(C_i)}{eH^i_{max}(C_k) - eH(C_i)} \quad (4)$$

This rate takes values between 0 and 1. A value that is close to 1 implies that the drill-down process doesn't change significantly the distribution of the records and, thus, no new insights are given to the analyst. The exact opposite happens when the value is close to, or zero. We can therefore exclude less interesting navigations in the lattice by defining a maximum external entropy rate threshold value between zero and one. When the external entropy rate of a drill down navigation step surpasses the threshold, then this drill down is omitted from consideration.

4.3 Internal Entropy Metric

With similar arguments we can introduce an internal entropy rate threshold in order to select subgraphs within a cuboid that differ significantly from the rest of the cuboid data. Since we consider directed data graphs, we distinguish between two kinds of internal entropy, namely starting internal entropy and ending internal entropy.

Consider cuboid C_i with l distinct combinations of starting attribute values of the form $(s^y_1, s^y_2, \ldots, s^y_s)$. Let m_y is the sum of the aggregate values of all such edges, where $y \in [1, l]$. For each such combination (indicated by parameter y) there are f_y edges with different combinations of ending attribute values. Let z_{q_y} be sum of their aggregate values as well. We calculate the starting internal entropy as the conditional entropy of the ending attributes' values conditioned from each starting attribute combination of values.

$$siH(C^y_i) = -\sum_{j=1}^{f_y} p(q^y_j) * \log_2 p(q^y_j) \quad where\, p(q^y_j) = \frac{z_{q_y}}{m_y} \quad (5)$$

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. 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, as will be explained in the next Section.

5 GRAPH CUBE ANALYSIS WORKFLOW

Motivated by the examples of the previous subsections, in this work we present techniques that

- Weigh all possible navigations within a graph cube lattice and suggest drill-down operations that reveal surprising trends, conditioned on what is observed in the more abstract cuboids contained in the cube. This process eliminates a significant portion of the graph cube, steering the user towards cuboids that reveal skew that is hidden when focusing in more abstract aggregations.
- Evaluate the relationships within the cuboids suggested from the previous step in order to reveal parts of data that contain skewed relationships.

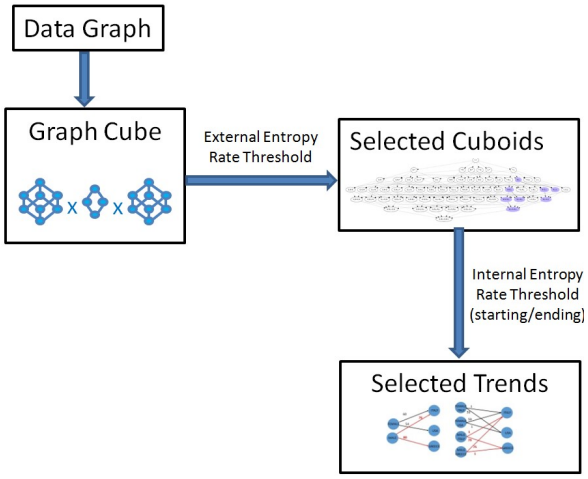


Figure 3: Graph Cube Analysis Workflow

In Figure 3 we depict the distinct steps involved in using our techniques for analyzing massive graph data cubes. After the graph cube is computed, we first utilize an external entropy rate threshold in order to prune edges of the lattice and, consequently, cuboids that do not provide significant insights with respect to their ancestors and descendants. For those cuboids that are connected by edges suggested by this process, we compute the internal entropy rates (for starting and ending attributes aggregated at the level denoted by the corresponding cuboid). We can then use a user-provided internal entropy rate threshold to only return relationships in these cuboids that do not exceed the threshold or, we can sort them and return the top-k selections in increasing order of internal entropy rate.

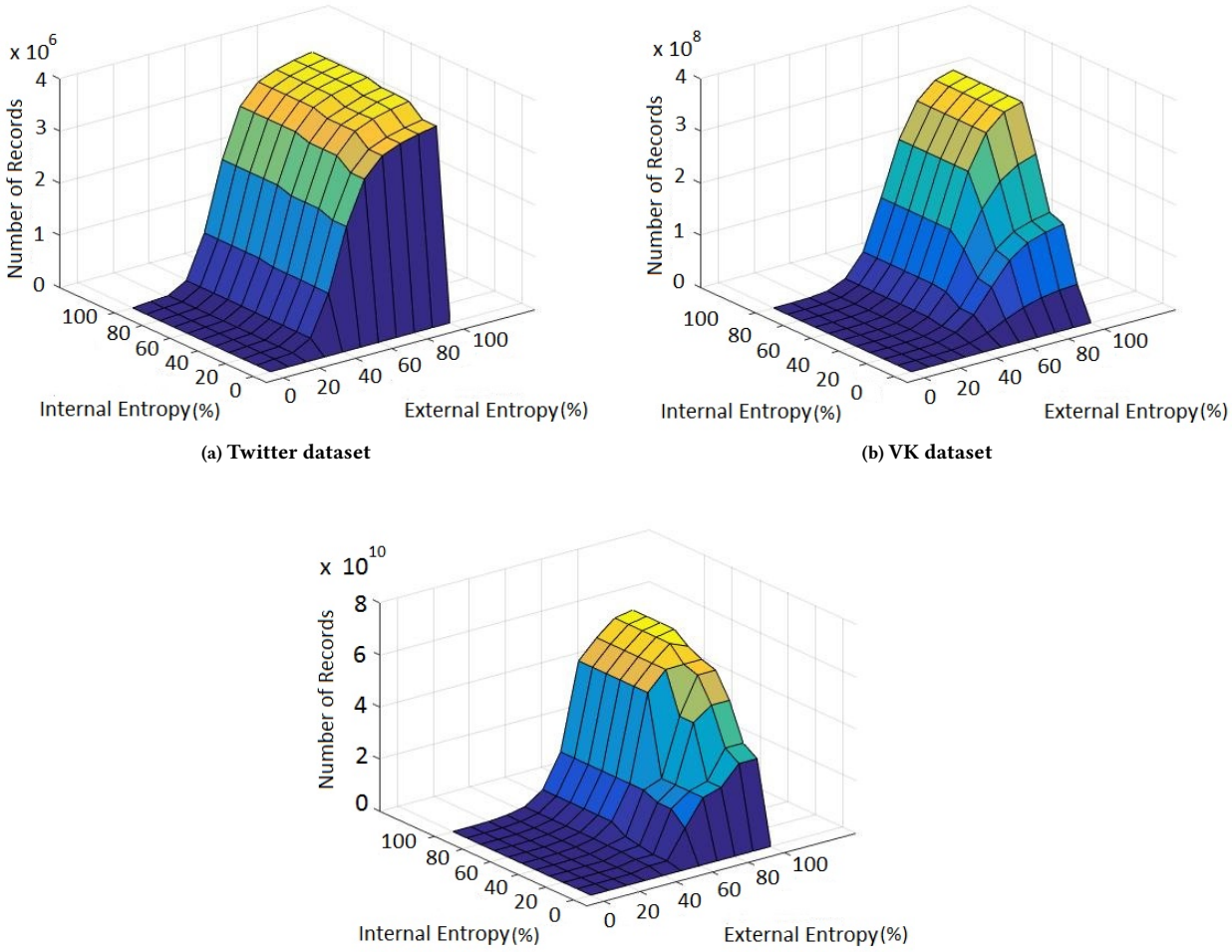
6 EXPERIMENTS

In this section, we provide preliminary results from applying our suggested framework on three real social network datasets. The focus on this exposition is to first highlight the pruning power of using entropy to navigate very large graph cubes and then to discuss some of the main trends observed in the social datasets used.

The datasets used are summarized in Table 1. The Twitter dataset was crawled by our team and contains 3 attributes: gender, location and language, used in each user profile. We also crawled the VK dataset from VKontakte, the largest European online social networking service. The sample contains 5 attributes: birthyear, country, city, gender and education level of the user. Finally, the Pokec dataset, available from [20] is a social-network from Slovakia and 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 1TB 7200rpm HDDs. We used the popular Apache Spark [34] framework on 8 VMs (one being the master) running on this cluster. The graph cube for each dataset was computed using an extension of the BUC algorithm discussed in [5].

In first experiment, we utilize the suggested data analysis workflow and evaluate the pruning power of the external and the



(a) Twitter dataset

(b) VK dataset

(c) Pokec dataset

Figure 4: Number of records in the graph cube, scaling both internal and external entropy rates

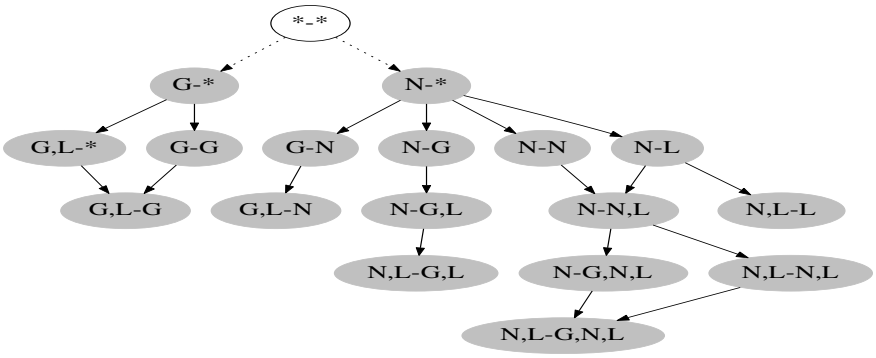
	Twitter	VK	Pokec
Profiles (nodes)	34M	3,9M	1,6M
Relations (edges)	910M	493M	31M
Number of Attributes	3	5	6
Number of Cuboids	64	1024	4096
Graph Cube Records	4M	362M	66,3B
Graph Cube Size	143MB	235GB	1.58TB
Cluster CPUs	4 × 4 Cores		
Cluster RAM	4 × 4 GB		

Table 1: Description of datasets and hardware used

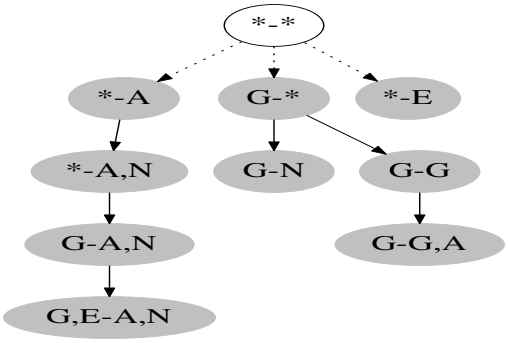
internal entropy metrics. Figures 4a,4b and 4c illustrate how the starting internal and external rates reduce the number of records of the graph cube, in each dataset. Plots for the using the ending internal entropy are similar and are omitted due to lack of space. The plots suggest a steep reduction in the sizes of the graph cubes for all datasets, as the respective entropy rate thresholds are increased. We observe that using thresholds in the ranges from 5% to 20% helps trim the million or billions (in the case of the Pokec dataset) records in the corresponding graph cubes to manageable sizes. This suggests that indeed, in these real data, there is a needle in the haystack that begs to be revealed. This is more evident in the largest graph cube from the Pokec dataset that contains 4096 cuboids and more than 66 billion records. In that dataset, a 10% external entropy threshold leads the analyst to focus on less than 0.002% of the aggregated graph cube records that contain 9 out of the 10 more prominent associations (when ranked in decreasing order of their internal entropy).

In Figures 5a, 5b and 5c we depict the filtered sub-lattices (sets of cuboids) selected when using an external rate threshold of 3.5% in the graph cube analysis workflow of Figure 3. For the Twitter dataset 17 out of the 64 cuboids of the graph cube are chosen. For the VK dataset 9 out of 1024 cuboids are retained. Finally, for the Pokec dataset only 10 from the 4096 cuboids are kept for post-processing. Based on the characteristics of the datasets shown in Table 1 we observe that the external entropy helps prune more cuboids when the number of node attributes is increased, as this results in larger lattices for the full graph cube.

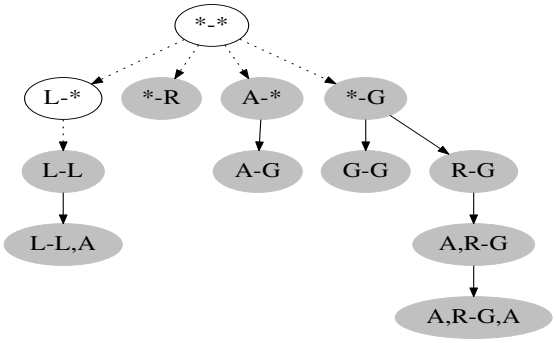
These filtered cuboids are used as input for the final stage of our workflow that further selects parts of these cuboids based in their internal entropy. For that step we used a rate threshold of 20% and present in Table 2 some characteristic results for each



(a) Twitter dataset



(b) VK dataset



(c) Pokec dataset

Figure 5: Selected sub-lattices for a 3.5% external entropy rate threshold

dataset. Due to space limitations the attributes in the table are shown with their first letter. Thus, N stands for nation, L for language, G for gender, A for age and E for education level.

In the Twitter dataset, we find that users from all countries follow mostly users from the USA. Exceptions include users from Portugal, Romania, Latvia, Venezuela, Taiwan, Chile, Brunei, Brazil and Norway. Users of these countries seek to follow mainly other users from the same country. From the cuboid (nation - gender) the entropy reveals that users from Monaco and Nauru follow males 2.2 times more often than females. Similarly, users from Thailand follow men 1.7 times more often than women. On the contrary, Mongolia users follow women 2.1 times more often than men.

From the VK dataset, we mine some other trends. Most connections are towards 35-year-old users from Russia and after that from Ukraine. Most connected users are born between 1986 and 1990. Users from USA are connected mostly with women, the same appears for users from Kazkhstan. Users connected with Turkish profiles are 70% men. Women are related uniformly with both genders while men are connected 60% with other men and 40% with women. Most users are connected to other profiles without a university degree and after that with users that got their diploma between 2008-2012.

Using the entropy-based techniques in the Pokec dataset we see other interesting trends. First, we observe that most relationships are towards women. Specifically, users between 19 and 22 years old have mainly connections to women. On the other hand, 19-year-old females are more frequently connected with other females. With respect to location, connections between the same cities dominate. Also, the most connections are with users from the Presovsky kraj and Presov regions. Users from most of the regions are connected with female users except for those from Nitriansky kraj and Nitra that are associated with more men. 19-year-old users from Presovsky kraj, Bardejov are connected mainly with male peers. Users between 32 and 37 years old from Banskobystricky kraj, Banska are connected mainly with females that are 22 years old.

The rightmost column of Table 2 depicts the support of the corresponding trend. The numbers validate our intuition that skewed trends are quite often hidden within valleys of uniform behavior. Indeed, most trends have small support values and would be, thus, missed by a frequent itemset counting algorithm.

7 RELATED WORK

The work in [35] 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

Dataset	*Trend*	Cuboid	$min(siH_{rate}, eiH_{rate})$	Support
Twitter	* - En	N - L	11.05%	87.12%
Twitter	* - USA	N - N	12.15%	36.07%
Twitter	Portugal - Portugal	N - N	12.21%	0.054%
Twitter	Romania - Romania	N - N	15.01%	0.025%
Twitter	Latvia - Latvia	N - N	15.23%	0.006%
Twitter	Venezuela - Venezuela	N - N	15.73%	0.009%
Twitter	Taiwan - Taiwan	N - N	16.10%	0.006%
Twitter	Chile - Chile	N - N	16.39%	0.029%
Twitter	Brunei - Brunei	N - N	16.88%	0.001%
Twitter	Brazil - Brazil	N - N	16.89%	0.564%
Twitter	Norway - Norway	N - N	17.03%	0.061%
Twitter	Monaco - Male	N - G	17.31%	0.002%
Twitter	Nauru - Male	N - G	17.71%	0.00004%
Twitter	Thailand - Male	N - G	17.93%	0.021%
Twitter	Mongolia - Female	N - G	18.06%	0.001%
VK	* - 35, Russia|Ukraine	* - A, N	13.23%	0.963%
VK	* - [1986..1990]	* - A	14.11%	2.388%
VK	Female - Usa|Kazakhstan	G - N	15.53%	0.478%
VK	Male - Turkey	G - N	16.42%	0.082%
VK	Male - Male	G - G	16.51%	37.74%
VK	* - No Diploma|Diploma 2008-2012	* - E	17.01%	13.42%
Pokec	age:[19..22] - Female	A - G	12.97%	0.098%
Pokec	age:19, Female - Female	A - G	13.05%	0.001%
Pokec	$City_x$ - $City_x$ (same city connection)	L - L	15.66%	9.908%
Pokec	Female - Male , Male - Female	G - G	15.98%	65.34%
Pokec	* - Presovsky kraj|Presov region	* - R	16.18%	0.032%
Pokec	Nitrianskykraj, Nitra - Man	R - G	16.47%	0.001%
Pokec	age:19, Presovsky kraj|Bardejov - Male	A, R - G	17.02%	0.712%
Pokec	age:[32..37], Branska - Female,22	A, R - G, A	17.11%	0.012%

Table 2: Main trends derived from the three social datasets

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. Another distinction is that the work of [35] 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 [33] considers aggregate attributed graphs. The authors name their model as a hyper graph cube and show how to compute it using MapReduce 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. OLAP-style summarization in the context of RDF graphs has been recently studied in [2]. The most significant difference from the previous works in graph cubes, is that our techniques address the vast size and complexity of the produced cuboids. To the best of our knowledge we are the first that utilize the entropy in order to filter the information of a graph cube.

The authors of [24] propose a novel framework for reconstructing multidimensional data from stored aggregates using the maximum entropy principle. In a nutshell, the proposed technique finds the model with the least information (maximum entropy) given a set of constraints that can be the $2^n - 2$ different aggregations in the cube (excluding the raw data and the grand total aggregate). The method uses a multi-pass algorithm called Iterative Proportional Filtering (IPF) that converges to the maximum entropy solution.

The information entropy was first introduced in [29] as a measure of unpredictability of information content. It measures how much information there is in an event. Entropy is frequently used for splitting decisions when computing Decision Trees [27] The information gain measures the change in information entropy from a prior state to new state after a split. Our external entropy rate measure utilizes the information gain metric in the nominator of its respective formula but differs in that it also takes into consideration the maximum possible increase in the entropy of a child cuboid in a drill down step. By conditioning the information gain over this quantity we are able to obtain the bounds that our selection algorithm utilizes.

Recently, an entropy-based model has been proposed [25] 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 [3, 4] 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.

Our techniques can be used in conjunction with existing systems for parallel graph processing [30] and tools like Perseus [19] that summarizes an input graph using statistics such as PageRank, radius, degree and flags outlier nodes [31], graph visualization tools [18], or with systems that recommend promising visualizations on aggregated datasets like SEEDB [32]. Our techniques may also be combined with the work of [13] that seeks intuitive drill-down operations from aggregated views of data.

Application of graph mining techniques [1, 8, 17, 21, 23, 26] is also orthogonal to our framework and can be used in conjunction. For instance, the work of [23] looks for structural patterns (or motifs) in the k-hop neighborhood of a node. The work of [21] suggests aggregation of graph nodes scores on vertices that contain some attribute of interest. Unlike conventional iceberg queries, the authors propose an aggregation method that is based on random walks and demonstrate their effectiveness and scalability. The authors of [7] explore data mining techniques to analyze tagging behavior on social graphs. The authors of [9] introduce graph-pattern association rules (GPAR). These rules extend traditional association rules with graph patterns that specify association between entities in a social graph.

There is recent work on systems that permit interactive exploration of very large data cubes. For example DICE [14] is a distributed system that utilizes faceted exploration in order to limit the number of possible queries in an interactive session. Extending this technique for graph cubes is an interesting research direction. Our entropy-based cube navigation framework can be combined with the idea of faceted exploration, either as a pre-processing step that limits the set of possible aggregations (cuboids) that need to be considered, or during interactive exploration by using the external/internal entropy rates in order to steer the user towards skewed correlations.

8 CONCLUSIONS

Graph data is becoming popular due to emerging applications that need to process and analyze interconnected datasets. In this work we proposed a graph data analysis framework based on the graph cube operator. Similar to the data cube, graph cubes contain an exponential number of aggregations of the raw data graph. Moreover, these aggregations are not simple flat records but rather complex graph structures that make their exploration cumbersome.

To overcome these obstacles our framework utilizes two novel entropy metrics that help locate unusual patterns hidden within billions of graph data aggregations. We put our framework to the test using three real social datasets of realistic sizes. Our preliminary results demonstrate that indeed entropy-guided exploration can help prune lots of uniform correlations enabling the analyst to focus on skewed parts of the data that often reveal interesting trends.

REFERENCES

[1] A. Arora, M. Sachan, and A. Bhattacharya. 2014. Mining Statistically Significant Connected Subgraphs in Vertex Labeled Graphs. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*. 1003–1014.

[2] E. Akbari Azirani, F. Goasdoué, I. Manolescu, and A. Roatis. 2015. Efficient OLAP operations for RDF analytics. In *ICDE Workshops*. 71–76.

[3] 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.

[4] Dritan Bleco and Yannis Kotidis. 2014. Graph Analytics on Massive Collections of Small Graphs. In *Proceedings of the EDBT, Athens, Greece*. 523–534.

[5] Dritan Bleco and Yannis Kotidis. 2017. Entropy-based Selection of Graph Cuboids. In *Proceedings of the 5th International Workshop on Graph Data Management Experiences and Systems (GRADES)*.

[6] C. Chen, X. Yan, F. Zhu, J. Han, and P. S. Yu. 2008. Graph OLAP: Towards Online Analytical Processing on Graphs. In *ICDM*. 103–112.

[7] M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. 2014. An Expressive Framework and Efficient Algorithms for the Analysis of Collaborative Tagging. *VLDB J.* 23, 2 (2014), 201–226.

[8] Mohammed Elseidy, Ehab Abdelhamid, Spiros Skiadopoulos, and Panos Kalnis. 2014. GRAMI: Frequent Subgraph and Pattern Mining in a Single Large Graph. *PVLDB* 7, 7 (2014), 517–528.

[9] W. Fan, X. Wang, Y. Wu, and J. Xu. 2015. Association Rules with Graph Patterns. *PVLDB* 8, 12 (2015), 1502–1513.

[10] A. Ghrab, O. Romero, S. Skhiri, A. A. Vaisman, and E. Zimányi. 2015. A Framework for Building OLAP Cubes on Graphs. In *Proceedings of ADBIS*.

[11] 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.

[12] W. H. Inmon. 1992. *Building the Data Warehouse*. QED Information Sciences, Inc., Wellesley, MA, USA.

[13] M. Joglekar, H. Garcia-Molina, and A. G. Parameswaran. 2016. Interactive Data Exploration with Smart Drill-down. In *Proceedings of ICDE*.

[14] Niranjan Kamat, Prasanth Jayachandran, Karthik Tunga, and Arnab Nandi. 2014. Distributed and Interactive Cube Exploration. In *IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31 - April 4, 2014*. 472–483.

[15] Kifayat-Ullah Khan, Kamran Najeebullah, Waqas Nawaz, and Young-Koo Lee. 2014. OLAP on Structurally Significant Data in Graphs. *CoRR* abs/1401.6887 (2014).

[16] Ralph Kimball and Margy Ross. 2002. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (2nd ed.). John Wiley & Sons, Inc., New York, NY, USA.

[17] Benny Kimelfeld and Phokion G. Kolaitis. 2014. The Complexity of Mining Maximal Frequent Subgraphs. *ACM Trans. Database Syst.* 39, 4 (2014), 32:1–32:33.

[18] D. Koop, J. Freire, and C. T. Silva. 2013. Visual Summaries for Graph Collections. In *IEEE Pacific Visualization Symposium, PacificVis 2013, February 27 2013-March 1, 2013, Sydney, NSW, Australia*. 57–64.

[19] D. Koutra, D. Jin, Y. Ning, and C. Faloutsos. 2015. Perseus: An Interactive Large-Scale Graph Mining and Visualization Tool. *PVLDB* 8, 12 (2015).

[20] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).

[21] N. Li, Z. Guan, L. Ren, J. Wu, J. Han, and X. Yan. 2013. gIceberg: Towards Iceberg Analysis in Large Graphs. In *29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013*. 1021–1032.

[22] Xiaolei Li, Jiawei Han, and Hector Gonzalez. 2004. High-Dimensional OLAP: A Minimal Cubing Approach. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*. 528–539.

[23] W. E. Moustafa, A. Deshpande, and L. Getoor. 2012. Ego-centric Graph Pattern Census. In *Proceedings of ICDE*. 234–245.

[24] T. Palpanas and N. Koudas. 2001. Entropy Based Approximate Querying and Exploration of Datacubes. In *Proceedings of SSDM*. 81–90.

[25] H. Pham, C. Shahabi, and Y. Liu. 2013. EBM: An Entropy-Based Model to Infer Social Strength from Spatiotemporal Data. In *Proc. of SIGMOD*.

[26] G. Qi, C. C. Aggarwal, and T. S. Huang. 2012. Community Detection with Edge Content in Social Media Networks. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*. 534–545.

[27] J. R. Quinlan. 1986. Induction of Decision Trees. *Mach. Learn.* 1, 1 (March 1986), 81–106.

[28] N. Roussopoulos, Y. Kotidis, and M. Roussopoulos. 1997. Cubetree: Organization of and Bulk Incremental Updates on the Data Cube. In *Proceedings of ACM SIGMOD, Tucson, Arizona*. 89–99.

[29] C. E. Shannon. 2001. A Mathematical Theory of Communication. *SIGMOBILE Mob. Comput. Commun. Rev.* 5, 1 (Jan. 2001), 3–55.

[30] V. Spyropoulos and Y. Kotidis. 2017. Digree: Building A Distributed Graph Processing Engine out of Single-node Graph Database Installations. *SIGMOD Record* 46, 4 (December 2017), 22–27.

[31] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. 2005. Neighborhood Formation and Anomaly Detection in Bipartite Graphs. In *Proceedings of ICDM*.

[32] M. Vartak, S. Rahman, S. Madden, A. G. Parameswaran, and N. Polyzotis. 2015. SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics. *PVLDB* 8, 13 (2015), 2182–2193.

[33] Zhengkui Wang, Qi Fan, Huiju Wang, Kian-Lee Tan, Divyakant Agrawal, and Amr El Abbadi. 2014. Pagrol: Parallel graph olap over large-scale attributed graphs. In *ICDE*.

[34] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *Proceedings of HotCloud*.

[35] Peixiang Zhao, Xiaolei Li, Dong Xin, and Jiawei Han. 2011. Graph Cube: On Warehousing and OLAP Multidimensional Networks. In *Proceedings of ACM SIGMOD*.