Ontology-based Graph Visualization for Summarized View

Xin Huang[†], Byron Choi[†], Jianliang Xu[†], William K. Cheung[†], Yanchun Zhang[‡*], Jiming Liu[†]

[†]Hong Kong Baptist University, [‡]Victoria University, [*]Fudan University
{xinhuang,bchoi,xujl,william,jiming}@comp.hkbu.edu.hk,yanchun.zhang@vu.edu.au

ABSTRACT

Data summarization that presents a small subset of a dataset to users has been widely applied in numerous applications and systems. Many datasets are coded with hierarchical terminologies, e.g., the international classification of Diseases-9, Medical Subject Heading, and Gene Ontology, to name a few. In this paper, we study the problem of selecting a diverse set of k elements to summarize an input dataset with hierarchical terminologies, and visualize the summary in an ontology structure. We propose an efficient greedy algorithm to solve the problem with $(1-1/e) \approx 62\%$-approximation guarantee. Preliminary experimental results on real-world datasets show the effectiveness and efficiency of the proposed algorithm for data summarization.

1 INTRODUCTION

Graphs consisting of nodes and edges are commonly used as a visualization tool for depiction and presentation of complex datasets. Graph representation offers direct, simplified, intuitive and human-friendly images to help users understand the overview of an analyzed dataset [4]. However, graph visualization works only if the complexity of the displayed dataset is within human cognitive capacity.

In real applications from various domains, a large number of datasets are coded with hierarchical terminologies. For example, in biomedicine, log datasets obtained from literature search tools or electronic health records (EHR) are usually aggregated by events, such as occurrences of diseases or findings, or entries of search terms [3, 4]. The events are typically represented by ontology-based terminologies, such as Gene Ontology[1], Disease Ontology[2], the International Classification of Diseases-9 (ICD-9), Medical Subject Heading (MeSH), and Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT). However, users may have difficulty in understanding the essence of terminologies in situations where the summary graph contains numerous terminologies, even with the aid of a good visualization tool. For instance, as of 2011, SNOMED CT contains more than 311,000 medical concepts;[3] it is impossible to visualize them all in a single graph. Therefore, designing efficient and effective algorithms for data summarization and visualization faces significant challenges [4, 10].

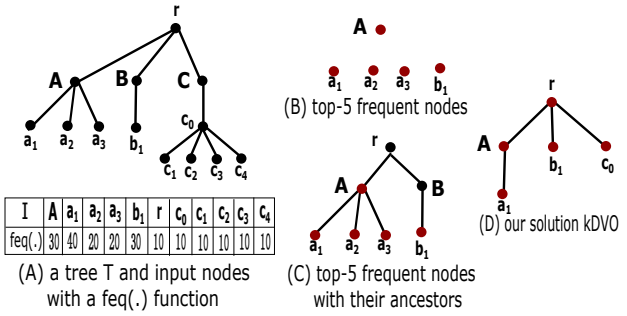


Figure 1: A running example

In the aforementioned applications, terminologies with hierarchical structures are often modeled as trees or directed acyclic graphs. In this study, we focus on tree structures. For instance, Figure 1(a) shows one sample example of disease ontology. The nodes $r, A, a_1, \ldots$ represent disease terminologies. The edges represent the instance relationship, e.g., (r, A) indicates that A is an instance of r. In general, the disease (node r) includes mental health disease (node A), syndrome disease (node B), and cellular proliferation disease (node C). Furthermore, the diseases of cellular proliferation (node C) have one instance of cancer (node c_0). In the third level, the types of cancers (node c_0) can be categorized into cells (node c_1), organ systems (node c_2), and so on. Given a table of frequencies that record the occurrence of diseases in a hospital (see the table in Figure 1(a)), one may seek a summary report that presents a clear structure of frequent diseases.

Obviously, if we show all diseases in the disease ontology, it is beyond the human cognition ability to distinguish any clear structure. Thus, we consider how to select a small set of k (e.g., $k = 5$) important and representative elements to summarize the entire dataset. The simplest approach is to pick the most frequent elements. However, as this approach does not make use of hierarchical terminologies, we cannot see the inter-relationships between the selected elements in the resulted summary (see Figure 1(b)). An improved approach is to also include all the ancestors of the top-k elements in the terminological structure (see Figure 1(c)). While this improved approach provides a more intuitive summary, it still suffers from two drawbacks. First, the summarization may lack diversity and miss specific but small groups (e.g., c_1, c_2, c_3, and c_4), which might yield limited aspects and inaccurate summarization for users. Second, similar elements are not summarized in a high-level concept. Moreover, to show all ancestors of frequent elements, a large graph might be resulted, e.g., Figure 1(c) has 7 nodes, which is greater than the given k. In contrast, Figure 1(d) depicts a better

[1] http://www.geneontology.org/
[2] http://disease-ontology.org
[3] https://en.wikipedia.org/wiki/SNOMED_CT

CIKM'17, November 6–10, 2017, Singapore.
 ISBN 978-1-4503-4918-5/17/11…$15.00
DOI: http://dx.doi.org/10.1145/3132847.3133113

summarization of the input dataset that describes four types of diseases (including A, a_1, b_1, and c_0), where element a_1 with the highest frequency represents a large proportion of type-A diseases.

To summarize, this paper makes the following contributions:

- We formally study the problem of selecting a diverse set of elements to summarize an input log-data set with hierarchical terminologies. We define the kVDO-problem for finding a set of k elements for graph **V**isulization from log-**D**ata using **O**ntology concepts. This new problem formulation takes into account the representativeness, diversity, and high-score coverage simultaneously (Section 3).
- We analyze the formulated objective score function, and formally prove its monotonity and submodularity properties, which offer the prospects for developing efficient and approximate algorithms (Section 4).
- We provide a novel method of summarizing large log-data by reducing the original dataset to a manageable size. It intends to depict, highlight, and distinguish the important nodes and links within the hierarchal structure. We propose an efficient algorithm that can achieve at least $(1 - 1/e)$ of the optimal in terms of our objective function (Section 5).
- We conduct experiments on real-world datasets to validate the efficiency and effectiveness of our proposed algorithm (Section 6).

2 RELATED WORK

Work closely related to our paper can be categorized into top-k diversification, data summarization and graph visualization.

Top-k diversification. In the literature, a large number of work studies the diversification of top-k query results [1, 5, 8, 11]. A comprehensive survey of top-k query processing can be found in [2]. The key distinction with these existing studies is that our approach takes a flexible method to find a summary graph with diversification and visualize it in an ontology structure.

Data summarization and graph visualization. There exist several studies on data summarization and graph visualization [3, 4, 7, 9, 10]. [4] investigates the problem of graphical visualization using ontology terminologies by filtering the nodes whose aggregate frequencies are less than a given threshold. [10] finds a set of k high-quality and diverse representatives for a surface, which does not consider the ontology structure associated with the data. Different from the above studies, our work considers the problem of data summarization using ontology terminologies, and formulates it as an optimization problem.

3 PROBLEM STATEMENT

In this section, we define basic notions and formalize our problem.

3.1 Preliminaries

We consider a finite set of n elements, $\mathcal{V}$, where the elements with inter-relations are organized into a tree-like structure. Let an undirected and unweighted tree $T = (\mathcal{V}, E)$ be rooted at $r \in \mathcal{V}$, where $E = \{(v, u) : v, u \in \mathcal{V}\}$ is the edge set. Tree T contains $n = |\mathcal{V}|$ nodes and $n - 1 = |E|$ edges. For each node v in T, we respectively denote the ancestors of node v by $\text{anc}(v)$ and the set of descendants of node v by $\text{dec}(v)$. Note that, we let $\text{anc}(v)$ and $\text{dec}(v)$ always contain v throughout this paper, i.e., $v \in \text{anc}(v)$ and $v \in \text{dec}(v)$. The node with no children is called leaf.

Definition 3.1 (Node Level). Given a tree T rooted at r, the level of a tree node $v \in \mathcal{V}$ is the number of hops between v and r, denoted by $l(v)$.

For example, consider a tree T in Figure 1(a). For node C, the set of descendants of C is $\text{dec}(C) = \{C, c_0, c_1, c_2, c_3, c_4\}$, and the set of ancestors is $\text{anc}(C) = \{r, C\}$. The level of node C is $l(C) = 1$, and the level of node c_2 is $l(c_2) = 3$.

Desiderata of a good summarization. Given a tree $T = (\mathcal{V}, E)$ and a finite set of input elements $\mathcal{I} \subseteq \mathcal{V}$ with a non-negative real-valued function feq, our goal, intuitively, is to select a small set of elements $\mathcal{S}$ from $\mathcal{V}$ that depicts a good summarization of the high-score data of $\mathcal{I}$ by satisfying the following three criteria:

1. (Diversity) The elements of $\mathcal{S}$ should not be very similar;
2. (Small-scale) The size of $\mathcal{S}$ is small enough to be visible;
3. (High-score Coverage and Correlation) A summary score function $\text{f}_{\mathcal{S}}(\mathcal{I})$ that measures the coverage and correlation of S in input nodes of I is high.

3.2 Summary Score Function

In this subsection, we propose a summary score function $\text{f}_{\mathcal{S}}(\mathcal{I})$ by formalizing the desiderata of diversity, high-score coverage, and correlations in a unified way. We first give the definitions of coverage and correlation below.

Coverage. Given two nodes x, y in tree T, we say x covers y if and only if x is one ancestor of y, i.e., $y \in \text{dec}(x)$. In the concept tree T, x covers y, indicating that x is a more general concept than y. This shows x can be a summary representative of y in a higher level of concept understanding. For instance, in Figure 1(a), node c_0 covers a set of nodes $\{c_1, c_2, c_3, c_4\}$, which means c_0 can be a good summary of all concepts in $\{c_1, c_2, c_3, c_4\}$.

Representative Impact. Based on the definition of coverage, we define the representative impact as follows.

Definition 3.2 (Representative Impact). Given two elements x, y and $y \in \text{dec}(x)$, we define the representative impact of x on the element y using a function rep_x :

$\text{rep}_x(y) = \text{feq}(y) \cdot \text{dis}_x(y)$,

where $\text{dis}_x : \mathcal{V} \to \mathbb{R}^{\geq 0}$ is the summarized relevance function.

Here, x serves as a candidate representative of y. The summarized impact of x on y is proportional to $\text{feq}(y)$, the score of y, and is discounted by $\text{dis}_x(y)$. Specifically, the summarized relevance of x achieves the maximum at $y = x$, and decreases for y further away from x. Note that, if x does not cover y, i.e., $y \notin \text{dec}(x)$, then $\text{dis}_x(y) = 0$ and certainly $\text{rep}_x(y) = 0$. In this paper, we suggest one natural choice of correlation function

$$\text{dis}_x(y) = \begin{cases} \dfrac{1}{l(y) - l(x) + 1}, & if\ y \in \text{dec}(x) \\ 0, & otherwise \end{cases} \quad (1)$$

For example, consider the tree T and the frequency function of elements as $\text{feq}(\cdot)$ in Figure 1(a). For nodes B and b_1 with the level $l(B) = 1$ and $l(b_1) = 2$, the summarized relevance of B on

b_1 is $\text{dis}_B(b_1) = 1/2$, and thus representative impact of B on b_1 is $\text{rep}_B(b_1) = \text{feq}(b_1) \cdot \text{dis}_B(b_1) = 30 \times 1/2 = 15$. On the other hand, the summarized relevance of r on b_1 is $\text{dis}_r(b_1) = 1/3$, and the representative impact $\text{rep}_r(b_1) = 10 < \text{rep}_B(b_1)$, indicating that B is a better summarized representative outperforming r, due to the more specification of B compared to r. Our models can adopt other settings of $\text{dis}_x(y)$ satisfying the principle of summarized relevance, and also our proposed techniques can be easily extended to solve a variant of problems with different $\text{dis}_x(y)$ functions.

Summary Score. Given a set $\mathcal{S} \subseteq \mathcal{V}$ of representative elements, we define the summary score of S on an input element $y \in \mathcal{V}$, denoted by $\text{smy}_S(y)$, as the maximum impact y among all individual representatives:

$$\text{smy}_S(y) = \max_{x \in S \cap \text{anc}(y)} \text{rep}_x(y). \quad (2)$$

Intuitively, each input element y is to be represented by some ancestor of y that appears in S (a.k.a. $x \in S \cap \text{anc}(y)$) and has the maximum summary impact on y. Based on the definition of summary score, the total summary impact of $\mathcal{S}$ on all elements of $\mathcal{I}$ is defined as:

$$\text{g}(S) = \sum_{y \in I} \text{smy}_S(y) = \sum_{y \in I} \max_{x \in S \cap \text{anc}(y)} (\text{feq}(y) \cdot \text{dis}_x(y)). \quad (3)$$

To recap, the problem of graph **V**isulization of log-**D**ata using **O**ntology concepts (kVDO-problem) studied in this paper can be formally formulated as follows.

kVDO-problem. Given a tree $T = (\mathcal{V}, E)$, a set of input elements $\mathcal{I} \subseteq \mathcal{V}$ with a non-negative real-valued function feq, and a number $k > 0$, find a set of representatives $S \subseteq \mathcal{V}$, such that S achieves the maximum score $\text{g}(S)$ with $|S| = k$.

Example 3.3. We use the example in Figure 1 to illustrate our kVDO-problem ($k = 5$) for visualizing the large dataset I in Figure 1(a) with the summary graph $S = \{r, A, a_1, b_1, c_0\}$ in Figure 1(d). For node $a_1 \in I$, the best representative of S is a_1 and the summary score of S on a_1 is $\text{smy}_S(a_1) = 40 \times 1 = 40$. Overall, the summary graph in Figure 1(d) achieves the score of $\text{g}(S) = 40 + 50 + 30 + 10 + 30 = 160$.

4 PROBLEM ANALYSIS

In this section, we analyze the properties of the objective score function of our problem.

Monotonity and Submodularity A set function $f : 2^U \rightarrow \mathbb{R}^{\geq 0}$ is said to be submodular provided for all sets $S \subset T \subset U$ and element $x \in U \setminus T$, $f(T \cup \{x\}) - f(T) \leq f(S \cup \{x\}) - f(S)$, i.e., the marginal gain of an element has the so-called "diminishing returns" property.

Lemma 4.1. g *is monotone, i.e., for all* $S_1, S_2 \subseteq \mathcal{V}$ *such that* $S_1 \subseteq S_2$, *we have* $\text{g}(S_1) \leq \text{g}(S_2)$.

Proof. The proof is trival and thus ommited. □

Given a summary node $x \in \mathcal{S}$, let the set of nodes that take x as their summary node, denoted by $\Phi_S(x) = \{y \in \text{dec}(x) : \text{smy}_S(y) = \text{rep}_x(y)\}$.

Lemma 4.2. g *is submodular.*

Algorithm 1 GVDO (T, I, k)

Require: A tree $T = (\mathcal{V}, E)$, a query $I \subseteq \mathcal{V}$, a number k.
Ensure: A set of k summary elements S.
1: Let $S \leftarrow \emptyset$;
2: **while** $|S| < k$ **do**
3: $x^* \leftarrow \arg\max_{x \in \mathcal{V}/S} \triangle_g(x|S)$;
4: $S \leftarrow S \cup \{x^*\}$;
5: **return** S;

Proof. Give two sets $S \subset T \subset \mathcal{V}$ and an element $x \in \mathcal{V} \setminus T$, let $T' = T \cup \{x\}$ and $S' = S \cup \{x\}$. We establish the correctness of Lemma 4.2 by following three facts below.

First, for any element $y \in \mathcal{V}$, $\text{smy}_T(y) \geq \text{smy}_S(y)$ and $\text{smy}_{T'}(y) \geq \text{smy}_{S'}(y)$ holds. Second, $\Phi_{T'}(x) \subseteq \Phi_{S'}(x)$. Since $\forall y \in \Phi_{T'}(x)$, we have $\text{rep}_x(y) = \text{smy}_{T'}(y) \geq \text{smy}_{S'}(y)$ and $\text{rep}_x(y) \leq \text{smy}_{S'}(y)$ for $x \in S'$. As a result, we obtain $\text{rep}_x(y) = \text{smy}_{S'}(y)$ and $y \in \Phi_{S'}(x)$. Therefore, $\Phi_{T'}(x) \subseteq \Phi_{S'}(x)$ holds. Third, we have $\text{g}(T') - \text{g}(T) = \sum_{y \in \mathcal{V}} (\text{smy}_{T'}(y) - \text{smy}_T(y)) = \sum_{y \in \Phi_{T'}(x)} (\text{rep}_x(y) - \text{smy}_T(y))$. Thus, we can obtain $\text{g}(S') - \text{g}(S) = \sum_{y \in \Phi_{S'}(x)} (\text{rep}_x(y) - \text{smy}_S(y)) \geq \sum_{y \in \Phi_{T'}(x))} (\text{rep}_x(y) - \text{smy}_S(y)) \geq \sum_{y \in \Phi_{T'}(x))} (\text{rep}_x(y) - \text{smy}_T(y)) = \text{g}(T') - \text{g}(T)$. As a result, $\text{g}(S') - \text{g}(S) \geq \text{g}(T') - \text{g}(T)$. □

5 GVDO ALGORITHM

In this section, we first give the framework of our greedy algorithm called GVDO. Then, we show its approximation guarantee and present several techniques for improving its efficiency.

Marginal gain. We begin with marginal gain. Monotonicity of function g implies that for any $S \subseteq \mathcal{V}$ and $x \in \mathcal{V}$, we have $\triangle_g(x|S) = g(S \cup \{x\}) - g(S) \geq 0$. The term $\triangle_g(x|S)$ is called the marginal gain of x to the set S. We would like to add the node with the largest marginal gain into the answer. This greedy strategy motivates the following algorithm GVDO.

Algorithm overview. GVDO starts out with an empty solution set $S = \emptyset$. In each subsequent iteration, GVDO iteratively adds one more summary node x^* to solution S, which grows the answer set by one. This summary node x^* is chosen from the remaining candidate elements $\mathcal{V}/S$ such that it achieves the largest marginal gain, i.e., $x^* \leftarrow \arg\max_{x \in \mathcal{V}/S} \triangle_g(x|S)$. Finally, GVDO returns S after $|S| = k$. The detailed description is presented in Algorithm 1.

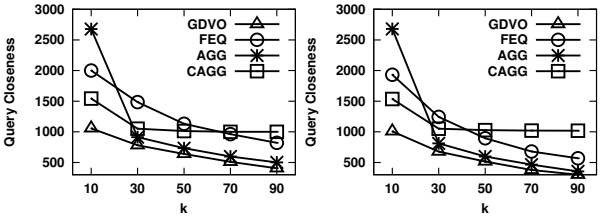
Computing $\triangle_g(x|S)$. We present an efficient algorithm (Algorithm 2) for computing the marginal gain $\triangle_g(x|S)$. Let $S' = S \cup \{x\}$, and T_x be a subtree of T rooted at x (lines 1-2). The procedure computes $\Phi_{S'}(x)$ by performing one traversal of tree T_x and finding all nodes regarding x as its new summary node. Afterwards, if we can find the nearest ancestor z of x in S, i.e. $\text{anc}(x) \cap S \neq \emptyset$, and calculate the marginal gain $\triangle_g(x|S) = \sum_{y \in \Phi_{S'}(x)} (\text{rep}_x(y) - \text{rep}_z(y))$; otherwise, if such an ancestor z does not exist, the algorithm directly returns $\triangle_g(x|S) = \sum_{y \in \Phi_{S'}(x)} \text{rep}_x(y)$.

Approximation Analysis. [6] shows that a greedy algorithm provides a $(1 - 1/e)$–approximation for maximizing a monotone submodular set function with cardinality constraint. Our method GVDO is one instantiation of this algorithm for kVDO-problem.

Theorem 5.1. *Let* S *be the answer obtained by* GVDO, *and* S^* *be the optimal answer,* $\text{g}(S) \geq (1 - \frac{1}{e}) \cdot \text{g}(S^*)$ *holds.*

Algorithm 2 Computing $\triangle_g(x|S)$

Require: A tree T, a query I, a summary set S, a node $x \in \mathcal{V}$.
Ensure: $\triangle_g(x|S)$.
1: $S' \leftarrow S \cup \{x\}$;
2: Compute $\Phi_{S'}(x) = \{y \in \text{dec}(x) : \text{smy}_{S'}(y) = \text{rep}_x(y)\}$;
3: **if** anc (x) $\cap S \neq \emptyset$ **then**
4: Let $z \in S$ be the nearest ancestor of x;
5: $\triangle_g(x|S) = \sum_{y \in \Phi_{S'}(x)} (\text{rep}_x(y) - \text{rep}_z(y))$;
6: **else**
7: $\triangle_g(x|S) = \sum_{y \in \Phi_{S'}(x)} \text{rep}_x(y)$;
8: **return** $\triangle_g(x|S)$;



(a) physician (b) nurses

Figure 2: Quality evaluation on *physician* and *nurses* data

Complexity Analysis. The overall time complexity of Algorithm 1 is $O(n^2k)$ time in worst cases. The space complexity is $O(n)$.

6 EXPERIMENTS

In this section, we test our algorithms in experiments.

Datasets. We use a real-world dataset of tree T containing hierarchical terminologies that are extracted from the Medical Entity Dictionary (MED) [4]. The tree contains 4,226 nodes. In addition, we use two datasets of $\mathcal{I}$, where one dataset *physician* contains the information about how physicians query online knowledge resources, and the other dataset *nurses* contains the query information of nurses. These two datasets contain 2,425 records and 2,034 records, respectively. Each record consists of a MED term with a frequency count of its occurrence in the log file.

Methods Compared. To evaluate our algorithm GVDO, we evaluate and compare three algorithms – FEQ, AGG, and CAGG. Here, FEQ is a baseline approach, which selects k nodes with the highest frequencies [4]. The algorithm AGG picks a set of k nodes with the highest aggregate frequencies, where the aggregate frequency of a node x is defined as $AF(v) = \sum_{y \in \text{dec}(x)} \text{feq}(y)$. CAGG is a variant method of AGG using another metric of contribution ratio. For a node x, the contribution ratio of x is defined by $R(v) = \frac{AF(x)}{AF(y)}$ where y is the parent of x. Given a ratio threshold θ, CAGG selects the k nodes that have the highest aggregate frequencies and the contribution ratio no less than θ. We set $\theta = 0.4$ by following [4]. For all methods, we set the parameter $k = 30$ by default.

Evaluation Metrics. To evaluate the quality of summary result S found by all algorithms, we randomly generate a set of query nodes Q following the frequency distribution of input nodes, and measure the closeness distance between query Q and summary S, denoted by $D(Q, S) = \sum_{q \in Q} \min_{x \in S} \text{dist}_T(q, x)$, where $\text{dist}_T(q, x)$ is the number of edges connecting q and x in tree T. The smaller is $D(Q, S)$, the better is the summary.

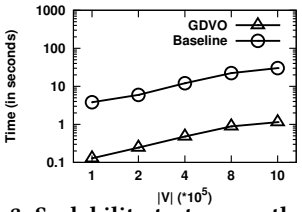


Figure 3: Scalability test on synthetic data

Quality Evaluation. Figures 2(a) and 2(b) show the quality evaluation on *physician* and *nurses* data by all algorithms. All approaches achieve smaller closeness distance with the increased k. Our approach GVDO is a clear winner of all competitors. It significantly outperforms the other methods for a smaller k, which is a great help to shrink large datasets for data summarization and visualization. The similar results can be observed in Figure 2(b).

Scalability Test. In this experiment, we evaluate the scalability of GVDO by varying the size of tree $|\mathcal{V}|$. We randomly generate 5 trees with size varying from 10^5 to 10^6. In addition, to verify the efficiency of computing $\triangle_g(x|S)$ by Algorithm 2, we compare one approach Baseline that follows Algorithm 1 by computing $\triangle_g(x|S)$ from scratch. The results of running time are shown in Figure 3. As we can see, GVDO is scalable very well with the increased size of tree nodes $|\mathcal{V}|$. Meanwhile, GVDO is much more efficient than Baseline, indicating the efficient strategy of Algorithm 2.

7 CONCLUSION

In this paper, we study the problem of ontology-based graph summary for visualization, and propose an efficient greedy algorithm with quality guarantee. Experiments on real-world datasets demonstrate the superiority of our proposed algorithm.

ACKNOWLEDGMENTS

This work was supported by the Hong Kong General Research Fund (GRF) Project Nos. HKBU 12200917, 12232716, 12200114, 12244916, and NSFC Grant No. 61672161.

REFERENCES

[1] I. Catallo, E. Ciceri, P. Fraternali, D. Martinenghi, and M. Tagliasacchi. Top-k diversity queries over bounded regions. *ACM Transactions on Database Systems (TODS)*, 38(2):10, 2013.
[2] I. F. Ilyas, G. Beskales, and M. A. Soliman. A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys (CSUR)*, 40(4):11, 2008.
[3] X. Jing, J. Cimino, et al. A complementary graphical method for reducing and analyzing large data sets. *Methods of information in medicine*, 53(3):173–185, 2014.
[4] X. Jing and J. J. Cimino. Graphical methods for reducing, visualizing and analyzing large data sets using hierarchical terminologies. In *AMIA Annual Symposium Proceedings*, volume 2011, page 635, 2011.
[5] R.-H. Li, J. X. Yu, X. Huang, H. Cheng, and Z. Shang. Measuring robustness of complex networks under mvc attack. In *CIKM*, pages 1512–1516, 2012.
[6] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions-i. *Mathematical Programming*, 14(1):265–294, 1978.
[7] S. Noel and S. Jajodia. Managing attack graph complexity through visual hierarchical aggregation. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 109–118, 2004.
[8] L. Qin, J. X. Yu, and L. Chang. Diversifying top-k results. *PVLDB*, 5(11):1124–1135, 2012.
[9] Y. Tian, R. A. Hankins, and J. M. Patel. Efficient aggregation for graph summarization. In *SIGMOD*, pages 567–580, 2008.
[10] Y. Wu, J. Gao, P. K. Agarwal, and J. Yang. Finding diverse, high-value representatives on a surface of answers. *PVLDB*, 10(7):793–804, 2017.
[11] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.