Research Article

Nosratali Ashrafi-Payaman, Mohammad Reza Kangavari*, and Amir Mohammad Fander

A new method for graph stream summarization based on both the structure and concepts

https://doi.org/10.1515/eng-2019-0060
Received Dec 13, 2018; accepted Nov 11, 2019

Abstract: Graph datasets are common in many application domains and for which their graphs are usually massive. One solution to process such massive graphs is summarization. There are two kinds of graphs, stationary and stream. For stationary graphs, a number of summarization algorithms are available while for graph stream there is no a comprehensive summarization method that summarizes a graph stream based on the structure, vertex attributes or both with varying contributions. This is because of challenges of graph stream, which are volume of data and changing of data over time. In this paper, we propose a method based on sliding-window model for which summarizes a graph stream based on a combination of the structure and vertex attributes. We proposed a new structure for summary graphs and also proposed new methods for comparing two summary graphs. To the best of our knowledge, this is the first method that summarizes a graph stream based on both the structure and vertex attributes with varying contributions. Through extensive experiments on real dataset of Amazon co-purchasing products, we have demonstrated the performance of the proposed method.

Keywords: Graph Stream Summarization, Attributed Graph, Summary Graph, Super-node, Super-edge

1 Introduction

Graph summarization is a useful and interesting topic that has been recently studied in the literature [1] widely. The general goal of summarization is to reduce a massive graph to a smaller one by removing unimportant details and preserving general properties of the graph. In a number of applications, data and their relations are modeled by a structural graph, *e.g.* cities and their ways. These graphs are summarized based on nodes and their relations [2–4]. On the other hand, a number of applications generate attributed graphs for which a number of attributes has been associated to vertices or even edges [5, 6] *e.g.* social networks such as Facebook. In Facebook, each node represents a person and has attributes such as name, family, country, and stuff.

In general, summarization is performed by grouping similar nodes into one group and dissimilar nodes into different groups [1]. Similarity of two vertices can be structurally or attribute-based or both. For example in Facebook, both edges and vertex attributes can take into account for constructing summary. Therefore based on the important of the structure, vertex attributes or both, summarization will be structural [4], attribute-based [7, 8] or hybrid [9]. The similarity of the nodes has an important impact on the resultant summary and can be calculated based on vertex connectivity or vertex attributes or both. Therefore, the similarity criterion of two vertices specifies the type of the resultant summary.

These days many application generate data which are received as a graph stream [10] such as selling products in supermarkets. For this example, the relationship between sold products are received as a stream of edges, an edge represents each pair of sold products. Although a number of algorithms have been proposed for summarizing stationery attributed graphs based on both the structure and vertex attributes with varying contributions of each [6, 8, 11], to the best of our knowledge there is no method capable of summarizing a graph stream based on both the structure and vertex attributes or both. This is the main challenge of graph stream summarization. In this paper, a new method has been proposed that addresses this challenge. The proposed method summarizes a graph stream based on sliding window paradigm. By using this proposed method, always a summary of the graph stream is available. We have provided experimental results on

***Corresponding Author: Mohammad Reza Kangavari:** School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran; Email: kangavari@iust.ac.ir
Nosratali Ashrafi-Payaman: School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran; Email: ashrafi@khu.ac.ir
Amir Mohammad Fander: School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran; Email: amirfander@gmail.com

Amazon product co-purchasing network dataset for evaluating the proposed method.

We propose a method for graph stream summarization based on sliding-window model. In this method a graph is summarized based on both the structure and vertex attributes. For comparing two summaries, we introduce a new schema for a summary graph and a new algorithm for calculating the difference between two summaries. In overall our contribution are as follows:

- A new method for graph stream summarization
- A new schema for a summary graph
- A new algorithm for comparing two summaries and calculating their distance.

The rest of this paper is organized as follows. In Section 2, related works are reviewed. Section 3 is dedicated to the proposed method. Our experiments are explained in Section 4. Discussions are presented in Section 5 and finally we have provided conclusions and discussion of future work in Section 6.

2 Related Works

In this section, we review previous works on four different types of graph summarization to discuss the main challenges of graph summarization.

2.1 Structural summarization

In [12] a method has been proposed for graph structural summarization. In this method, a graph is compressed by partitioning similar nodes into one group and dissimilar nodes into different groups. For a compressed graph, a super-edge is the aggregated edges between a pair of supernodes. In this method, a graph is compressed based on the Minimum Description Length (MDL) idea. Firstly, they developed a greedy algorithm and secondly to reduce the runtime of the algorithm, they proposed a randomized version.

Riondato *et al.* [3] proposed another method to summarize structural graphs. In this method, the aim is to guarantee the quality of a summary and minimizing the reconstruction errors. Riondato *et al.* have presented a connection between graph summarization and geometric clustering problems for the first time. Based on this connection, the authors developed a polynomial-time algorithm to generate the best possible summary of the expected size.

Tian *et al.* [13] proposed three distributed summarization algorithms named DistGreedy, DistRandom and DistLSH to summarize large scale graphs. These algorithms differ in how they select a pair of nodes to merge, which they select greedy, randomly, and using locality sensitive hashing theory, respectively.

Chen *et al.* [14] proposed a method based on producing randomized summary graphs for identifying frequent patterns. Structural summaries can be beneficial for frequent pattern mining. In fact, instead of mining massive and time-consuming original graphs, summary graphs are mined.

In fact spectral graph clustering can be used for structural summarization. Spectral graph clustering partitions a graph based on eigenvalues and eigenvectors of the graph adjacency matrix [15–19]. This technique can be beneficial in image segmentation and social network analysis. There are a number of applications that use spectral graph clustering for finding communities in networks [20]. In this applications, initially a large graph is converted into a small one by summarization and then use spectral graph clustering to cluster the resultant small summary graph [21].

Community detection algorithms has many applications and recently, many articles [22–26] have been published on this subject. Graph summarization can be beneficial for detecting communities in a network.

Of-course, there are some similar methods/models [27, 28], subgraph mining models, which are limited in comparison with summarization methods. These models rather than summarizing, choose one or more subsets from graphs.

2.2 Attribute-based summarization

In [7], a summarization method with two novel operations **S**ummarization by grouping **N**odes on **A**ttributes and **P**airwise relations (SNAP) and k-SNAP has been proposed. These operations are used for grouping nodes and summarizing attributed graphs. Tian *et al.* defined attribute and relation compatible grouping. They also improved SNAP operation by proposing k-SNAP operation. In k-SNAP operation, k is the summary size where is determined by the user. The k-SNAP operation improved by Zhang *et al.* [8] by proposing the CANAL algorithm in 2009. The CANAL algorithm is used to categorize attribute values automatically, and also to provide a criterion to measure the quality of a summary.

In 2008, the OLAP framework has been proposed by Chen *et al.* [29]. In the OLAP framework, the cubes are created on the graph based on dimensions and measures. In

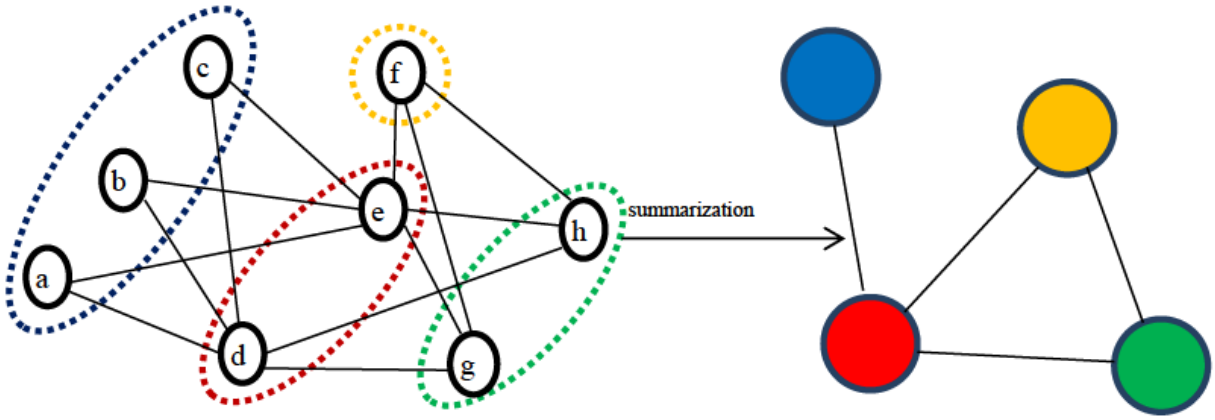


Figure 1: Original graph (left) and its summary (right)

this framework, a graph is summarized based on both selected attributes and input information.

2.3 Hybrid summarization

In [6] a method was proposed for clustering a graph based on both the structure and vertex attributes. In this method, for a given graph a new graph, named the augmented graph, with real and virtual links is constructed. Because of attribute-based similarity of vertices, the virtual links are added to the new graph. In the augmented graph, both real and virtual links are considered to measure the similarity of two nodes. If the number of associated attributes is relatively high, the augmented graph will be massive and finally the runtime of the algorithm is high.

In [11] another method has been proposed to hybrid summarization of a graph. In this method, initially a graph is summarized based on vertex attributes, without take into consideration the graph structure, and then by moving nodes between super-nodes adjust the summary to the graph structure. This method for situations where the structure has an important impact in constructing summary may be inefficient.

In [30] a method has been proposed for attributed graphs which constructs a hybrid summary by considering MDL principle to model the graph summarization problem into a code cost function and utilizing greedy method to compute an optimal summary. In this method, the user's needs and also the ontology of the graph have not been considered.

2.4 Graph stream summarization

There has been some research work on the subject of graph stream summarization but the contribution of these works in the scope of graph stream summarization is not significant. Major research work done on graph stream are as follows.

In this [31] a novel Graph Stream Sketch (GSS) has been proposed to summarize graph streams with linear space cost ($O(|E|)$) and constant update time complexity ($O(1)$). The aim of Gou *et al.* has been constructing a summary for query answering with the controllable errors.

In [32] the focus is on calculating the rank of a node in a graph stream with the minimum passes over the stream and the minimum space, of-course up to an adaptive error. Therefore, algorithms and models has presented in this regard.

In [33] Feigenbaum *et al.* have been interested in the trade-offs between model parameters such as per-data-item processing time, required space, and the required number of passes over the stream. These trade-offs have been considered for solving problems such as Spanner Construction, BFS-Tree Construction, Graph Distance Lower-Bounds.

In [34] a new variation of streaming model with a helper which can provide annotations for data streams have been proposed by Cormode *et al.* They have discussed that by giving linear sized annotations, the memory for many problems is reduced to constant time.

In [35] Feigenbaum proposed a new streaming model and formulized it. They believed this model is necessary for proposing efficient algorithms to solve problems on massive graphs. They have considered an upper bound for required spaces foe such algorithms. They applied the proposed model on special problems.

In [36] Aggarwal also *et al*. proposed a method for graph stream clustering by introducing micro-clusters and compressing them with hash functions. The proposed method can be beneficial for special applications. Aggarwal proposed a new method for classification a massive domain graph stream [37]. Aggarwal has proposed a probabilistic approach for constructing a summary that can be stored in main memory. Aggarwal used this method for determining special patterns in a graph stream.

There are other works on graph stream such as the problems of connectivity [35], counting subgraphs *e.g.* triangles [38, 39], calculating the degree of nodes [40], spanners [41], sparsification [42]. Thus to the best of our knowledge, there is no capable method for summarizing a graph stream which converts a graph to a smaller one by removing unnecessary details and preserving overall properties.

3 The proposed method

In the proposed method, we use sliding-window model for summarizing a graph stream. We take into account the edges of the first window over the graph stream and construct the graph of this window. This graph is summarized using hybrid summarization method proposed by the authors of this paper [9, 43] for summarizing an attributed graph based on both the structure and vertex attributes. The summary graph is maintained as a reference. We take into account edges of the second window over the graph stream and its graph is constructed and summarized. This summary is named current summary. The current summary is compared to the reference summary. Depending on whether they are matched or not, one of them is skipped. If they are matched then the current summary is skipped otherwise the distance of two summaries is higher than a given threshold. For the latter case, the current summary is maintained as the reference summary. In this case, the current summary is the best representative of the graph stream.

By continuing this trend, a summary of the graph stream is available at any moment. The paradigm of the new method for graph stream summarization depicted in Figure 2.

3.1 Algorithm

The proposed method has been summarized in Algorithm 1. In this algorithm, summarizing a graph, comparing two summary graphs and calculating the distance of two summary graphs are not clear and should be illustrated more. In the following subsections, we illustrate the structure of a summary graph, similarity of two super-nodes, comparing two summary graphs and finally the proposed method is illustrated by an example.

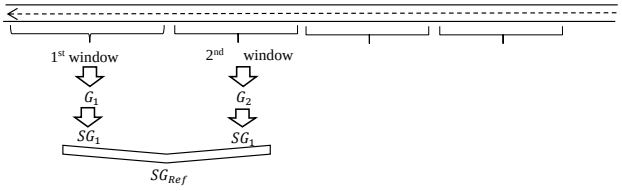


Figure 2: The proposed method for graph stream summarization

Algorithm 1: Graph stream summarization

Input: A graph stream;
Output: A reference summary graph;
1: Begin
2: consider the window w_0 on the graph stream;
3: Construct the graph of w_0 and summarize it as SG_R.
4: While(true)
5: consider the next window w_c on the graph stream;
6: construct the graph of w_c and summarize it as SG_c.
7: compare two summary graphs and let d=dist(SG_R, SG_c);
8: if(d > threshold) replace SG_R with SG_c;
9: else ignore SG_c.
10: Endwhile;
11: end.

3.2 The summary structure

In the proposed method, an attributed graph is summarized based on both the structure and vertex attributes. Every super-node in the summary graph is a vector of structural and semantical attributes. Structural attributes are the number of vertices in the super-node, the degree of the super-node and the percentage of vertices, which are relevant with nodes of the other super-nodes. Semantical attributes are considered as the percentage of vertices, which have a value on an attribute. In fact, for every value of a vertex attribute this percentage value is calculated. In Section 3.5 we illustrate the summary structure by an example.

3.3 Similarity of two super-nodes

Based on the proposed structure for the summary graph, a super-node is a vector of attributes and the similarity of two super-nodes is calculated based on their vectors. The similarity of two super-nodes is calculated using Equation (1), which also uses Equations (2) through (6). Initially, the distance of two super-nodes is calculated and then by subtracting this value from one, the similarity of two super-nodes is obtained.

$$\text{sim}(V_p, V_q) = 1 - dis(V_p, V_q) \tag{1}$$

The distance between two super-nodes is equal to summation of structural and attribute-based distance of two super-nodes, which is presented by Equation (2).

$$dis(V_p, V_q) = \frac{1}{2}\left(dis_{st}(V_p, V_q) + dis_{att}(V_p, V_q)\right) \tag{2}$$

The number of vertices, the degree of super-nodes and the number of vertices which relevant to vertices of other super-nodes (the weight of edges) are considered as structural attributes. These structural attributes determine the structural distance of two super-nodes. Equation (3) describes the structural distance of two super-nodes. As seen in Equation (3), the value of structural distance belongs to [0, 1]. For each of the three parts of Equation (3), if the denominator is zero, the value of that part is considered to be zero.

$$dis_{st}(V_p, V_q) = \frac{1}{3}\left(\frac{(n_p - n_q)^2}{n^2_{\max}} + \frac{(d_p - d_q)^2}{d^2_{\max}} + \frac{1}{\max(d, d')}\sum_{i=1}^{\max(d,d')}(pe_{p,i} - pe_{q,i})^2\right) \tag{3}$$

The attribute-based distance between two super-nodes with k attributes and each attribute with k' values is calculated using Equation (4).

$$dis_{att}(V_p, V_q) = \frac{1}{k}\sum_{i=1}^{k} dis_{att}(V_{p,i}, V_{q,i}) \tag{4}$$

where

$$dis_{att}(V_{p,i}, V_{q,i}) = \frac{1}{k'}\sum_{j=1}^{k'} dis_{att}(V_{p,i,j}, V_{q,i,j})^2 \tag{5}$$

where

$$dis_{att}(V_{p,i,j}, V_{q,i,j}) = per_{p,i,j} - per_{q,i,j} \tag{6}$$

where n_p and d_q are the number of vertices in V_p and the degree of V_p, respectively.

3.4 Comparing two summary graphs

To compute the distance of two summary graphs, initially the similarity of each pair of super-nodes of two summary graphs is calculated using Equation (1). The super-node pairs with the most similarity are associated. After associating the super-nodes of two summary graphs, the distance of two summary graphs is calculated. The distance is equal to summation of distances of each pair of matched super-nodes. The approach for calculating the distance of two summary graphs has been described in Algorithm 2.

Algorithm 2: Calculating distance two summary graphs

Input: summary graphs: GS1 and GS2 and the size of summary graph: size;
Output: distance of two summaries;
1: Begin
2: Calculate the distance of every two super-nodes
3: Add every super-node pair with its calculated distance to ascending priority queue q;
4: n=summary graph size;
5: While(n>0)
6: Remove a super-node pair;
7: Match two super-node of the pair;
8: n−;
9: endwhile;
10: set **dsit** to sum of distances of the matched super-node pairs;
11: end.

3.5 Illustrating the proposed method with an example

To clarify the issue, we consider two summary graphs SG_1 and SG_2 with above-mentioned structure, each with three

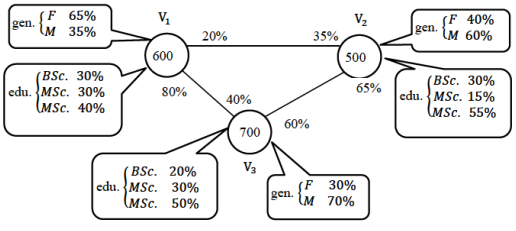


Figure 3: First summary with three super-nodes and two attributes. Attribute values, size of super-nodes and the percentage of vertices, which are in relationship to vertices of the other super-nodes, are shown in the summary.

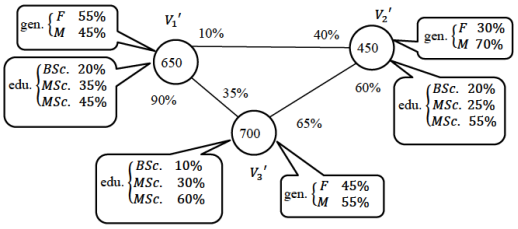


Figure 4: Second summary with three super-nodes and two attributes. Attribute values, size of super-nodes and the percentage of vertices, which are in relationship to vertices of the other super-nodes, shown in the summary.

super-nodes and two attributes. Attributes are gender and education level, gender with values of Male and Female and education level with values of BSc., MSc. and Ph.D. As we see in Figure 3, the summary graph shows the overall and important information of the original graph. For example, super-node V_1 shows a group of 600 people where 20% are in relationship with people of V_2, 80% are in relationship with people of V_3, 65% are female and 35% are male, 30% are bachelor of science, 30% are master of science and 30% are Doctor of Philosophy.

As already mentioned, initially the similarity of every pair of super-nodes of two summary graphs is calculated. For clarify, in the following we calculate the similarity of two super-nodes V_1 and V_1', step-by-step.

$$\begin{aligned} dis_{st}(V_1, V_1') &= \frac{1}{3}\left(\frac{(600-650)^2}{650^2} + \frac{(2-2)^2}{2^2} + \frac{1}{\max(2,2)}\sum_{i=1}^{\max(2,2)}(pe_{p,i} - pe_{q,i})^2\right) \\ &= \frac{1}{3}\left(\frac{(50)^2}{650^2} + 0 + \frac{1}{2}\left((0.9-0.8)^2 + (0.2-0.1)^2\right)\right) \\ &= \frac{1}{3}\left(\frac{5\times 5}{65\times 65} + 0 + \frac{1}{2}\left(0.01^2 + 0.01^2\right)\right) \\ &= \frac{1}{3}\left(\frac{5\times 5}{65\times 65} + 0 + \frac{1}{2}(0.01+0.01)\right) \\ &= \frac{1}{3}\left(\frac{1}{13\times 13} + 0 + 0.01\right) = 0.0053 \end{aligned}$$

$$\begin{aligned} dis_{att}(V_1, V_1') &= \frac{1}{2}\left(\frac{1}{2}\left((0.65-0.55)^2 + (0.35-0.45)^2\right) + \frac{1}{3}\left((0.3-0.2)^2 + (0.3-0.35)^2 + (0.4-0.45)^2\right)\right) = \frac{1}{2}\left(\frac{1}{2}(0.01+0.01) + \frac{1}{3}(0.01+0.0025+0.0025)\right) \\ &= \frac{1}{2}\left(0.01 + \frac{1}{3}(0.01+0.0025+0.0025)\right) \\ &= \frac{1}{2}(0.01+0.005) = 0.0075 \end{aligned}$$

$$dis_{att}(V_1, V_1') = \frac{1}{2}(0.0053 + 0.0075) = 0.0064$$
$$\Longrightarrow sim_{att}(V_1, V_1') = 0.9936$$

For saving time, we have refused to provide computational steps for other super-nodes pairs and only have entered their final similarity values in Table 1.

Table 1: The similarity of super-nodes

	V_1'	V_2'	V_3'
V_1	0.9899	0.9522	0.9762
V_2	0.9713	0.9945	0.9755
V_3	0.9602	0.9783	0.9920

Based on Table 1, matched super-nodes are (V_2, V_2'), (V_3, V_3') and (V_1, V_1'), respectively. The first component of each pair is a super-node of the first summary graph and the second component is its matched super-node of the second summary graph. According to this matching, the distance of two summary graphs SG_1 and SG_2 is calculated as follows:

$$\begin{aligned} dis(SG_1, SG_2) &= dis(V_1, V_1') + dis(V_2, V_2') + dis(V_3, V_3') \\ &= 0.9899 + 0.9945 + 0.9920 = 2.9764 \end{aligned}$$

4 Experiment

In this section, we conducted experiments to evaluate the performance of the proposed method on real-world graphs. The proposed method was implemented in Java programming language.

4.1 Dataset

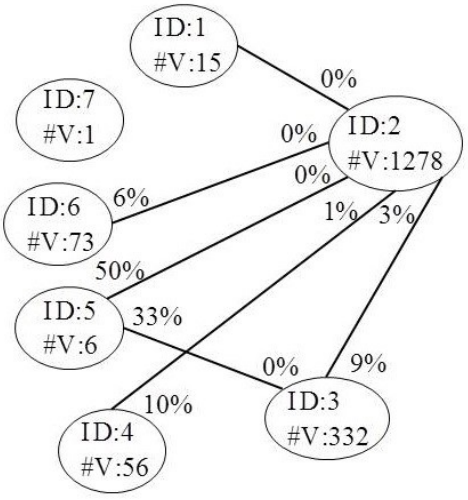
Amazon co-purchasing network

This data is available in address[1] and includes information about different products such as the books, music

1 https://snap.stanford.edu/data

Table 2: The information of Amazon co-purchasing network

Name	#Nodes	#Edges	Duration
amazon0302	262,111	1,234,877	Amazon product co-purchasing network from March 2 2003
amazon0312	400,727	3,200,440	Amazon product co-purchasing network from March 12 2003
amazon0505	410,236	3,356,824	Amazon product co-purchasing network from May 5 2003
amazon0601	403,394	3,387,388	Amazon product co-purchasing network from June 1 2003
amazon-meta	548,552	1,788,725	Amazon product metadata: product info and all reviews on around 548,552 products



Number of products

Super-nodes

	#dis	#Book	#Video	#Music	#DVD	#Toy
1	15	0	0	0	0	0
2	0	1258	8	5	7	0
3	0	0	0	332	0	0
4	0	0	0	1	55	0
5	0	3	0	0	3	0
6	0	0	72	1	0	0
7	0	0	0	0	0	1

Figure 5: The first summary graph of the size of 7

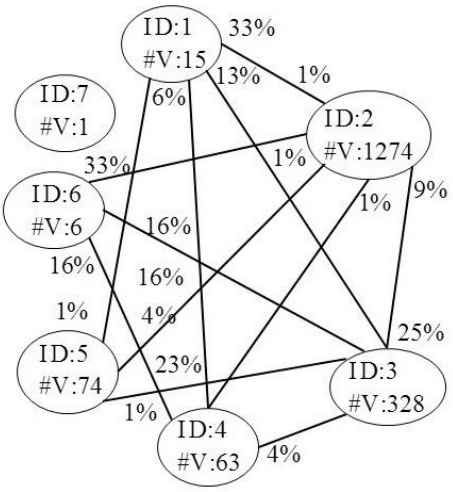
CDs, DVDs and VHS video tapes. There are 548,552 products and for each product, the information such as title, salesrank, list of similar products, category and reviews is available. This data are about Amazon co-purchasing products of 2003 and has been collected in summer 2006 by Jure Leskovec with crawling Amazon website. The information of this products and their graph streams are presented in Table 2. Rows second to fifth show four directed graph streams. Each graph is a graph stream where each edge (x, y) shows product y has frequently co-purchased with product x. We chose Id, ASIN, group and salesrank fields for providing experiments.

4.2 Evaluation

To the best of our knowledge, our proposed method is a novel general-purpose method for graph stream summarization and there is no competitor method for exact evaluation. We believed that comparing the results of our proposed method with the changes of real constructed graphs are more reasonable and reliable than comparing to other competitor methods.

Therefore, for evaluating the proposed method for graph stream summarization, we chose amazon0302 file and set the window size to 1000 edges. We considered the first window over the first 1000 edges of the file and constructed the first graph. For every window, the vertices are those, which are appeared at least as one end of the first 5000 edges. The graph of the first window has been summarized and resulted a summary graph with the size of 7. The summary graph is maintained as a reference. The next windows are also considered over the graph stream and in the following tasks such as summarizing graphs, comparing every coming graph with the reference summary and changing the reference summary(if necessary) are done. In this experiment, window size was fixed, 1000 edges, but usually the first window is considered bigger than the others are. In the following, 5 summary graphs are presented in Figures 5 through 9. In these Figures, **dis** and **toy** represent discontinued and toy products. These two categories do not belong to the main categories which are mentioned in the description of the dataset.

The semantic of each summary graph has been extracted and shown in Table 3. Semantic changes of two consecutive summary graphs are presented in Table 4. Fig-

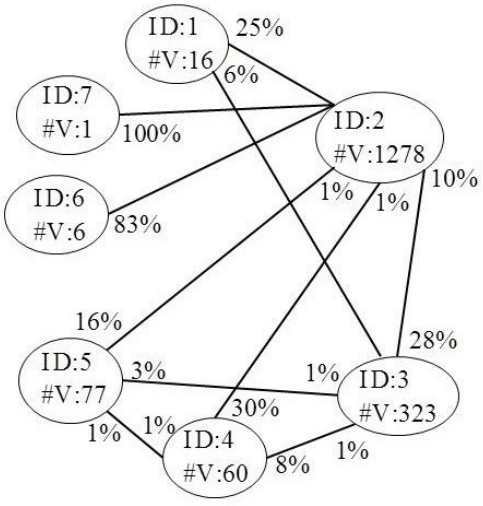


Number of products

Super_node

	#dis	#Book	#Video	#Music	#DVD	#Toy
1	15	0	0	0	0	0
2	0	1254	4	14	2	0
3	0	5	1	322	0	0
4	0	0	0	0	63	0
5	0	0	74	0	0	0
6	0	2	1	3	0	0
7	0	0	0	0	0	1

Figure 6: The second summary graph of the size of 7

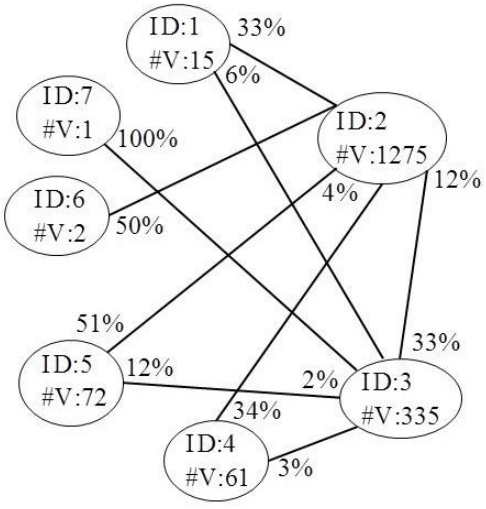


Number of products

Super_node

	#dis	#Book	#Video	#Music	#DVD	#Toy
1	15	0	0	1	0	0
2	0	1257	3	15	3	0
3	0	2	0	319	2	0
4	0	0	0	0	60	0
5	0	0	76	1	0	0
6	0	2	1	3	0	0
7	0	0	0	0	0	1

Figure 7: The third summary graph of the size of 7

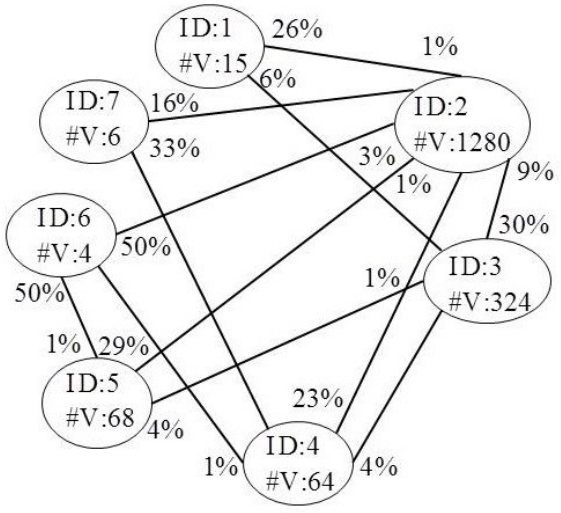


Number of products

Super_node

	#dis	#Book	#Video	#Music	#DVD	#Toy
1	15	0	0	0	0	0
2	0	1254	7	12	2	0
3	0	7	2	324	2	0
4	0	0	1	0	60	0
5	0	0	70	2	0	0
6	0	0	0	1	1	0
7	0	0	0	0	0	1

Figure 8: The fourth summary graph of the size of 7



Number of products

Super_node

	#dis	#Book	#Video	#Music	#DVD	#Toy
1	15	0	0	0	0	0
2	0	1254	9	15	2	1
3	0	2	4	318	0	0
4	0	0	0	2	62	0
5	0	0	66	2	0	0
6	0	2	0	1	1	0
7	0	3	1	2	0	0

Figure 9: The fifth summary graph of the size of 7

ures 5 through 9 and Tables 3 and 4 are used to discuss the experimental results.

4.3 Time complexity

In the proposed method, the dominant time belongs to the summarization algorithm. According to summarization algorithm in [9, 43], initially the similarity of each pair of nodes is calculated and after that the graph is summarized by merging nodes/super-nodes. In the worst case, the summarization algorithm performs at most |V| merge operations to obtain the expected summary. Henceforth, the time complexity of this method is $O(|E|\times|V|)$. Time complexity of other processes such as calculating distance between two super-nodes, matching super-nodes of two summary graphs and finally calculating the distance of two summary graphs is less than the runtime of summarization algorithm.

5 Discussions

The summary graphs as shown in Figures 5 through 9, the semantic of each summary as shown in Table 3 and distance of every two summary graphs as shown in Table 4, help us to justify distance of every two consecutive summary graphs intuitively. In fact, the calculated distances based on above-mentioned formulas should be supported by intuitive structural and semantical changes.

As shown in Table 4, the first and second summary graphs have distance value of about 0.3 and intuitively these two summary graphs differ in two cases as shown in the fourth column. Therefore, the distance of these two summary graphs is supported by the intuitive changes. Such a situation can be seen for the second and third summary graphs. On the other hand, the third and fourth summary graphs have a lower distance in comparison to the previous consecutive summary graphs and this is also in line with the intuitive changes. Third and fourth summary graphs differ intuitively only in one case, the existence or absence of 4-clique. The situation for the fourth and fifth summary graphs is similar to consecutive summary graphs of the first through third. Therefore, the calculated distance for each pair of consecutive summary graphs is according to intuitive distance of summary graphs and it is reliable.

By setting the threshold value of the distance between two summary graphs, it is determined whether the reference needs to be changed or not. For example, if we set the threshold to 0.5 then the first summary graph is remained as the reference. On the other hand, if we set the threshold to 0.2 then initially the reference will be the first summary graph, with the appearance of the second graph, this new summary graph will be the reference summary graph. This will also happen for the third summary graph, and third summary graph will be the reference. With the appearance of the fourth summary graph, the reference summary does not change. With the advent of the fifth summary graph, the reference will change and it will be replaced with the fifth summary graph. The threshold value can be determined in terms of scope and precision.

It is obvious that our proposed method is a general-purpose method, because of taking into account the struc-

Table 3: semantic of the summary graphs

row	Summary graph	Summary graph semantic
1	Figure 5	Discontinued products are related with books.
		The majority of books are related to themselves.
		The majority of the super-nodes are isolated.
2	Figure 6	Discontinued products are related with all other products.
		Super-nodes are related to each other (a near clique).
		Only toy the super-node of Toy is isolated.
3	Figure 7	There is no an isolated super-node.
		All super-nodes are in relationship with the super-node of book.
		There is 4-clique in the graph.
4	Figure 8	There is no an isolated super-node.
		All super-nodes are in relationship with book super-node.
		The number of sold music products is maximal.
5	Figure 9	There is no category of toy in the summary graph
		All super-nodes are in relationship with book super-node.
		The majority of books are in relationship with each other.

Table 4: semantic changes of the consecutive summary graphs

row	Summary graphs	distance	Semantic changes
1	Figure 5	0.32612	The graph become denser
	Figure 6		Discontinued products have been sold with other products.
2	Figure 6	0.4389434	isolated super-node.
	Figure 7		4-clique
3	Figure 7	0.14072233	4-clique.
	Figure 8		
4	Figure 8	0.36400673	Toy category
	Figure 9		The number of edges

ture, vertex attributes, user's needs and graph ontology in summarization. Hence, the summary graph can be beneficial in community detection, node degree calculations and stuff. By initially setting parameters in summarization algorithm, it is possible to change the orientation of the summarization algorithm.

6 Conclusions

In this paper, we proposed a method for graph stream summarization based on the sliding-window model. In the proposed method, a graph is summarized based on both the structure and vertex attributes. The super-nodes of two summary graphs are matched to each other and the distance of every pair of matched super-nodes is calculated. The distance of two summary graphs is calculated based on the calculated distances of the matched super-nodes. If the difference of the new summary graph and the reference is higher than the threshold, then the reference will be replaced with the new summary graph.

To the best of our knowledge, this is the first method for summarizing a graph stream based on both the structure and vertex attributes. In this way, always the summary of the graph stream is available. The summary graph is a representative of the graph stream, which has the overall properties of the graph stream and can be used for decision-making. In this paper, a number of algorithms have been proposed for calculating the similarity of two super-nodes, matching super-nodes and calculating the distance of two summary graphs.

In the proposed method, the window size was chose fixed while considering windows of varying length is more logical. We plan to extend the proposed method by consider windows of varying length and learning the size of window algorithmically.

In real-world applications, some of data are missed and this issue should be considered in providing experiments. On the other hand, a number of applications generate more than one graph streams and these graph streams should be summarized simultaneously. A future research venue would be summarizing multiple graph streams.

References

[1] Liu Y., Safavi T., Dighe A., Koutra D., Graph Summarization Methods and Applications: A Survey, *ACM Comput. Surv.*, 2018, 51.3, p. 62.

[2] Navlakha S., Rastogi R., Shrivastava N., Graph summarization with bounded error, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, 2008, p. 419.

[3] Riondato M., García-Soriano D., Bonchi F., Graph summarization with quality guarantees, *Data Min. Knowl. Discov.*, 2017, 31. 2, 314–349.

[4] LeFevre K. , Terzi E., GraSS: Graph Structure Summarization, in *Proceedings of 2010 SIAM International Conference on Data Mining*, 2013, 454–465.

[5] Basu-Roy S., Eliassi-Rad T., Papadimitriou S., Fast and Effective Pattern Matching on Weighted Attributed Graphs, *ACM Knowl. Discov. Data Min.*, 2013.

[6] Cheng H., Zhou Y., Yu J. X., Clustering Large Attributed Graphs: A Balance between Structural and Attribute Similarities, *ACM Trans. Knowl. Discov. Data*, 2011, 5.2, 12:1-12:33.

[7] Tian Y., Hankins R. A., Patel J. M., Efficient aggregation for graph summarization, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, 2008, 567–80.

[8] Zhang N., Tian Y., Patel J. M., Discovery-driven graph summarization, in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 2010, 880–891.

[9] Ashrafi-Payaman N. , Kangavari M. R., GSSC: Graph summarization based on both structure and concepts, *Int. J. Inf. Commun. Technol. Res.*, 2017, 9.1, 33–44.

[10] McGregor A., Graph Stream Algorithms: A Survey, *ACM SIGMOD Rec.*, 2014, 43.1, 9–20,.

[11] Bei Y., Lin Z., Chen D., Summarizing scale-free networks based on virtual and real links, *Phys. A Stat. Mech. its Appl.*, 2016, 444, 360–372.

[12] Navlakha S., Rastogi R., Shrivastava N., Graph summarization with bounded error, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data - SIGMOD '08*, 2008, 419–432.

[13] Liu X., Tian Y., He Q., Lee W. C., McPherson J., Distributed Graph Summarization, in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14*, 2014, 799–808.

[14] Chen C. ,Lin C., Mining graph patterns efficiently via randomized summaries, in *Proceedings of the VLDB Endowment 2.1*, 2009, 2.1 742–753.

[15] Von Luxburg U., A tutorial on spectral clustering, *Stat. Comput.*, 2007, 17.4, 395–416,.

[16] Dhillon I. S., Guan Y., Kulis B., A Unified View of Kernel k-means, *Spectral Clustering and Graph Cuts*. 2004.

[17] Auffarth B., Spectral Graph Clustering, *Univ. Barcelona course Rep. Tech. Av. Aprendizaj Univ. Politec. Catalunya*, 2007.

[18] Uw S., Ng A. Y., Jordan M. I., Weiss Y., On spectral clustering: Analysis and an algorithm, *Adv. Neural Inf. Process. Syst.*, 2002, 14, 849–856.

[19] Zhou D. , Burges C. J. C., Spectral clustering and transductive learning with multiple views, in *Proceedings of the 24th international conference on Machine learning - ICML '07*, 2007, 1159–1166.

[20] Smyth S. , White S., A spectral clustering approach to finding communities in graphs, in *Proceedings of the 5th SIAM International Conference on Data Mining*, 2005, 274–285.

[21] Liu J., Wang C., Danilevsky M., Han J., Large-scale spectral clustering on graphs, in *IJCAI International Joint Conference on Artificial Intelligence*, 2013.

[22] Wang C.-D., Lai J. H., Yu P. S., Dynamic Community Detection in Weighted Graph Streams, in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013, 151–161.

[23] Arts G., Member S., Overlapping Community Detection Algorithms in Dynamic Networks: An Overview, *Int. J. Emerg. Technol. Comput. Appl. Sci.*, 2013.

[24] Lancichinetti A. , Fortunato S., Community detection algorithms: A comparative analysis, *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, 2009, 80.5, 1–11.

[25] Wang W., Street W. N., A novel algorithm for community detection and influence ranking in social networks, *ASONAM 2014 - Proc. 2014 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Min.*, 2014, 555–560.

[26] Benyahia O., Largeron C., Jeudy B., Community detection in dynamic graphs with missing edges, in *Proceedings of International Conference on Research Challenges in Information Science*, 2017, 372–381.

[27] Hosseini S., Yin H., Zhang M., Elovici Y., Zhou X., Mining subgraphs from propagation networks through temporal dynamic analysis, in *Proceedings* of *IEEE International Conference on Mobile Data Management*, 2018, 66–75.

[28] Hosseini S., Yin H., Cheung N. M., Leng K. P., Elovici Y., Zhou X., Exploiting reshaping subgraphs from bilateral propagation graphs, in *International Conference on Database Systems for Advanced Applications*, 2018, 342–351.

[29] Chen C., Yan X., Zhu F., Han J., Yu P. S., Graph OLAP: Towards online analytical processing on graphs, in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008, 103–112.

[30] Wu Y., Zhong Z., Xiong W., Jing N., Graph summarization for attributed graphs, in *Proceedings - 2014 International Conference on Information Science, Electronics and Electrical Engineering, ISEEE 2014*, 2014, 503–507.

[31] Gou X., Zou L., Zhao C., Yang T., Fast and Accurate Graph Stream Summarization, in *IEEE 35th International Conference on Data Engineering (ICDE). IEEE*, 2019, 1118–1129.

[32] Das Sarma A., Gollapudi S., Panigrahy R., Estimating PageRank on graph streams, *J. ACM*, 2011, 58.3, p. 13.

[33] Feigenbaum J., Kannan S., McGregor A., Suri S., Zhang J., Graph Distances in the Data-Stream Model, *SIAM J. Comput.*, vol. 38, no. 5, pp. 1709–1727.

[34] Cormode G., Mitzenmacher M., Thaler J., Streaming graph computations with a helpful advisor, *Algorithmica*, 2013, 65.2, 409–442.

[35] Feigenbaum J., Kannan S., Mcgregor A., Suri S., Zhang J., On graph problems in a semi-streaming model, , 2005, 348. 2, 207–216.

[36] Aggarwal C. C., Zhao Y., Yu P. S., On Clustering Graph Streams, in *Proceedings of the 2010 SIAM International Conference on Data Mining*, 2010, 478–489.

[37] Aggarwal C. C., On Classification of Graph Streams, in *Proceedings of the 2011 SIAM International Conference on Data Mining*, 2013, 652–663.

[38] Tsourakakis C. E., Kang U., Miller G. L., Faloutsos C., DOULION: Counting Triangles in Massive Graphs with a Coin, in *KDD '09: 15th International Conference on Knowledge Discovery and Data Mining*, 2009, 837–846.

[39] Braverman V., Ostrovsky R., Vilenchik D., How hard is counting triangles in the streaming model?, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, vol. 7965 LNCS, no. PART 1, 244–254.

[40] Cormode G., Muthukrishnan S., An Improved Data-Stream Summary. The Count-min Sketch and its Applications, *J. Algorithms*, 2005, 55.1, 58–75.

[41] Ahn K. J., Guha S., McGregor A., Graph sketches: sparsification, spanners, and subgraphs, in *Proceedings of the 31st symposium on Principles of Database Systems - PODS '12*, 2012, 5–14.

[42] Ahn K. J., Guha S., Graph sparsification in the semi-streaming model, in *In International Colloquium on Automata, Languages, and Programming*, 2009, 328–338.

[43] Ashrafi-Payaman N., Kangavari M. R., Graph Hybrid Summarization, 2018, 6.2, 335–340.