Flow Provenance in Temporal Interaction Networks

Chrysanthi Kosyfaki
University of Ioannina
xkosifaki@cs.uoi.gr

ABSTRACT

In temporal interaction networks, such as financial transaction networks, vertices model entities which exchange quantities (e.g., money) over time. We study the problem of identifying the origin of the quantities that flow into the vertices of the network over time. We consider various models of flow relay, which are related to different application scenarios and develop corresponding techniques for flow provenance.

KEYWORDS

interaction networks, quantity, provenance

ACM Reference Format:
Chrysanthi Kosyfaki. 2021. Flow Provenance in Temporal Interaction Networks. In *Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21), June 18–27, 2021, Virtual Event, China.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3448016.3450581

1 INTRODUCTION

Many real world applications can be represented as *temporal interaction networks*, which capture the information flow between entities over time. Interaction networks are considered as a useful tool to model a wide variety of problems and have been studied extensively in literature [9], [6], [8], [4]. Examples of such graphs are road networks, social networks, traffic networks, the Bitcoin Network, food web etc. Hence, each edge of the network stores the history of transactions between the corresponding nodes. Each transaction is modeled by a timestamp and a quantity (e.g., money, messages, kbytes, vehicles), which was transferred at that time. Figure 1 shows a small interaction network, where each edge holds a sequence of (time,quantity) pairs. For example, pair (2, 5) on edge (v_0, v_1) means that vertex v_0 transferred 5 units to v_1 at time 2. For example, the users of a cryptocurrency network could be the vertices and the edges could model the transactions between them (i.e., the times and the transferred amounts).

We study a *flow provenance* problem in interaction networks; our goal is to find the origin of the quantities that have been accumulated at one or more vertices of the network throughout the history of interactions. To our knowledge, there is no previous work on finding the provenance of the quantities that flow into vertices in a temporal interaction network. This problem finds different applications in different research fields. Finding the origin of money in a financial entity can facilitate checking for suspicious activities or fraud. In a traffic network, the origin of vehicles involved in a traffic jam can be used to re-design the network or planning of activities. Tracking the origin of malicious data that flow in a communication network can help toward alleviating such activities.

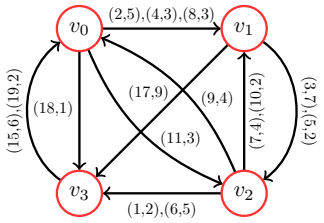

Figure 1: An example of an interaction network

2 BACKGROUND AND RELATED WORK

Data Provenance is a core concept in graphs [11],[3], [13]. In query evaluation, for example, it is important to know which data in the database contribute to a query result. The graph in this case is the query evaluation plan. In their seminal paper, Buneman et al. [1] study and define the problem of data provenance in database systems from two different perspectives: *why* and *where*. According to the authors, the goal of why provenance is to explain the existence of a tuple in a query result. That is, the reasons behind its inclusion. Why provenance can be answered by finding the path(s) in the query evaluation plan that contribute to the result. Where provenance is much simpler as it only requires to find the tuples in the source tables of the query that contribute to the result. Buneman et al. proposed a deterministic model, where the graph edges are labeled by data that they use and the provenance is modeled using paths. Our models are similar to the deterministic model, however in our case we consider (i) the temporal information of the interactions on the edges and (ii) the time of birth of the transferred quantities and (iii) different models for the transfer of quantities between vertices.

Provenance has also been studied in blockchain systems especially after the huge success of the Bitcoin. In [12], a secure and efficient system called LineageChain is proposed for capturing the provenance on runtime and safely stored. It was implemented using the Hyperledger, a well-known framework in blockchain systems. Another work that studies the problem of provenance is [2]. In this paper, the authors proposed techniques for reducing the storage requirements for provenance in database systems. They proposed a number of techniques based on factorization, which find common subtrees and unify them. Other techniques that use to solve the problem are related to inheritance and prediction. Titian [5] is a Spark-based system for data provenance.

SIGMOD '21, June 18–27, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8343-1/21/06…$15.00
https://doi.org/10.1145/3448016.3450581

In summary, although the concept of data provenance in databases is well-studied, there is no previous work on *flow provenance* in temporal interaction networks. In the next section, we define the problem and present our methodology.

3 APPROACH

The input to our problem is a network, which captures the history of interactions between vertices. For all vertices of the interaction network (or a given subset of vertices), our goal is to compute the origin of the quantity that is accumulated to them by the end of the timeline.

The quantity at each vertex is computed based on the assumption that each interaction (q, t) from a vertex v to a vertex u transfers from the accumulated quantity at the source vertex v by time t, either q (if v has at least q units in its buffer) or the entire accumulated quantity if it is less than q. In the latter case, v *generates* the difference q', and q' is marked to have v as its origin as it flows in the network. Hence, we assume that throughout the history of interactions, each vertex v has a *buffer* B_v. At the end of the timeline, B_v represents the total quantity accumulated at v (which was not relayed). By accessing all interactions in time order, we can compute the buffered quantities of all vertices. Our provenance problem is to find the origin(s) of B_v for each vertex v.

If, for an interaction (q, t), the source vertex v has a buffer $B_v > q$, then it is necessary to select which part of B_v will be relayed to the destination vertex u. Hence, for the case where $B_v > q$, we study different relay models, which are based on realistic assumptions:

- *least recently born prioritization:* this model gives priority to the quantities that have the oldest birth timestamps. As an example, consider the transaction $(6, 4)$ shown in Figure 2 from v to u. Assume that, by time $t = 6$, $B_v = 5$, i.e., $q = 4 < B_v$. Assume that B_v is analyzed to $\{[(x, 2), 1], [(y, 3), 2]\}$, meaning that from the total B_v, 2 units originate from vertex x, born at time 1 and 3 units originate from vertex y, born at time 2. Based on this model, v will relay to u the $q = 4$ quantity units which were the least recently born, i.e., quantities $\{[(x, 2), 1], [(y, 2), 2]\}$; hence, B_v will be updated to $B_v = \{[(y, 1), 2]\}$.
- *most recently born prioritization:* this is the same as the previous model, with the only difference being that priority is given to the most recently born quantities.
- *proportional selection:* The transfered quantity is selected proportionally, based on the origin. Consider again our previous example with transaction $(6, 4)$ from vertex v with $B_v = 5$. As Figure 3 shows, If 2 units of B_v originate from x and 3 units originate from y, then quantities $\{[x, 1.6], [y, 2.4]\}$ will be transferred from B_v to B_u.

4 RESULTS AND CONTRIBUTIONS

Our contributions include (i) the formalization of a flow provenance problem in temporal interaction networks, (ii) the consideration of different flow relay models, (iii) the implementation of provenance tracking techniques based on these models, and (iv) the study of the runtime performance of the techniques and the impact/relevance of the different models on interaction networks in real applications.

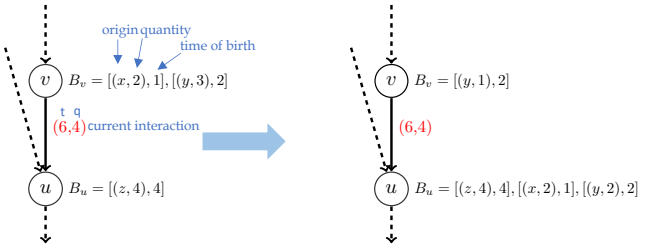

Figure 2: Least recently born relay model

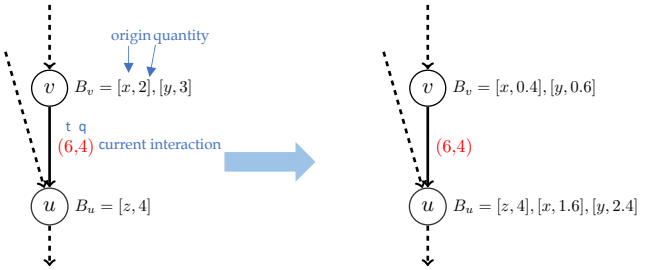

Figure 3: Proportional selection model

We are currently developing and testing the above models, in terms of their applicability and their efficiency. We conducted preliminary experiments on two real temporal interaction networks. The first one includes Bitcoin transactions [7]; the vertices represent users and the edges are transactions between them. The second one is constructed from an online peer-to-peer loan service (https://www.prosper.com); vertices are users who exchange money as loans. In Table 1, we compare the runtime of using the models to track provenance as the vertex buffers are updated. Method NoProv just updates the buffers without computing any provenance information. We observe that the first two models are not very expensive, because the provenance information stored at the buffers is relatively small. On the other hand, the last model is very expensive, because each interaction typically adds multiple new provenance elements to the destination vertex. This greatly increases the memory requirements, causing the model to be very slow. On the Bitcoin, the proportional selection model could not terminate within 24h.

Table 1: Runtime of execution for each proposed method

Dataset	NoProv	Most Recently	Least Recently	ProvProp
Bitcoin	0.19 sec	29.34 sec	9.29 sec	–
Prosper Loans	0.0065 sec	0.080 sec	0.072 sec	45.024 sec

Hence, we are considering to use the last two models only for the case of the problem where we are interested in tracking the provenance at a small subset of vertices of the graph. In addition, we are studying techniques for scaling up the performance for the most expensive models (proportional selection). We plan to use data parallelism in order to speed-up the updates of buffers, which are linear operations on data vectors. For this purpose, we plan to confine the models to consider as origins only the vertices which contribute the most to the network (i.e., the most important vertices); this can greatly reduce the space requirements.

REFERENCES

[1] Peter Buneman, Sanjeev Khanna, and Tan Wang-Chiew. 2001. Why and where: A characterization of data provenance. In *International conference on database theory*. Springer, 316–330.

[2] Adriane P Chapman, Hosagrahar V Jagadish, and Prakash Ramanan. 2008. Efficient provenance storage. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 993–1006.

[3] Zaheer Chothia, John Liagouris, Frank McSherry, and Timothy Roscoe. 2016. *Explaining outputs in modern data analytics.* Technical Report. ETH Zurich.

[4] Christian Decker and Roger Wattenhofer. 2013. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*. IEEE, 1–10.

[5] Matteo Interlandi, Kshitij Shah, Sai Deep Tetali, Muhammad Ali Gulzar, Seunghyun Yoo, Miryung Kim, Todd Millstein, and Tyson Condie. 2015. Titian: Data provenance support in spark. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*. 216.

[6] David Kempe, Jon Kleinberg, and Éva Tardos. 2005. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*. Springer, 1127–1138.

[7] Dániel Kondor, Márton Pósfai, István Csabai, and Gábor Vattay. 2013. Do the rich get richer? An empirical analysis of the BitCoin transaction network. *PLoS ONE* 9, 2 (2013), e86197.

[8] Dariusz Król, Damien Fay, and Bogdan Gabrys. 2015. *Propagation phenomena in real world networks*. Vol. 85. Springer.

[9] Rohit Kumar and Toon Calders. 2017. Information Propagation in Interaction Networks.. In *EDBT*, Vol. 17. 270–281.

[10] Satoshi Nakamoto. 2019. *Bitcoin: A peer-to-peer electronic cash system.* Technical Report. Manubot.

[11] Fotis Psallidas and Eugene Wu. 2018. SMOKE: Fine-grained Lineage at Interactive Speed. *Proceedings of the VLDB Endowment* (2018).

[12] Pingcheng Ruan, Gang Chen, Tien Tuan Anh Dinh, Qian Lin, Beng Chin Ooi, and Meihui Zhang. 2019. Fine-grained, secure and efficient data provenance on blockchain systems. *Proceedings of the VLDB Endowment* (2019), 975–988.

[13] Lukas Rupprecht, James C Davis, Constantine Arnold, Yaniv Gur, and Deepavali Bhagwat. 2020. Improving reproducibility of data science pipelines through transparent provenance capture. *Proceedings of the VLDB Endowment* (2020), 3354–3368.