# Frequent Temporal Social Behavior Search in Information Networks

Hsun-Ping Hsieh
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
d98944006@csie.ntu.edu.tw

Cheng-Te Li
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
d98944005@csie.ntu.edu.tw

Shou-De Lin
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
sdlin@csie.ntu.edu.tw

## ABSTRACT

In current social networking service (SNS) such as Facebook, there are diverse kinds of interactions between entity types. One commonly-used activity of SNS users is to track and observe the representative social and temporal behaviors of other individuals. This inspires us to propose a new problem of *Temporal Social Behavior Search* (*TSBS*) from social interactions in an information network: given a structural query with associated temporal labels, how to find the subgraph instances satisfying the query structure and temporal requirements? In *TSBS*, a query can be (a) a topological structure, (b) the partially-assigned individuals on nodes, and/or (c) the temporal sequential labels on edges. The *TSBS* method consists of two parts: offline miningand online matching. to the former mines the temporal subgraph patterns for retrieving representative structures that match the query. Then based on the given query, we perform the online structural matching on the mined patterns and return the top-*k* resulting subgraphs. Experiments on academic datasets demonstrate the effectiveness of *TSBS*.

## Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval.

## General Terms

Algorithms, Performance, Design.

## Keywords

Structural matching, Temporal subgraph pattern mining, Social search.

## 1. INTRODUCTION

With the popularity of a social network, nowadays numerous social interactions do occur between individuals. Faced with such complicated and overwhelming social behaviors in an information network, it is critical to be able to efficiently track an individual's representative behavior or discover certain specified social topology from it.

For example, in bibliography collections, an author usually involves in the co-author social behaviors with publication activities over time. Consider the co-authorships in Figure 1, we can find the author, $A_1$, co-works with his colleagues and students for different venues at distinct timestamps. The time interval associated on "co-author-of" edges indicates the submission and notification time of the papers.

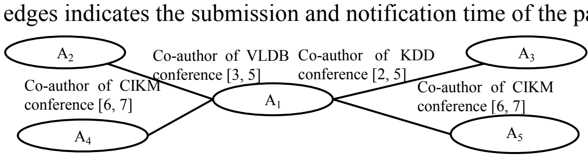


**Figure 1: A co-author relationship example.**



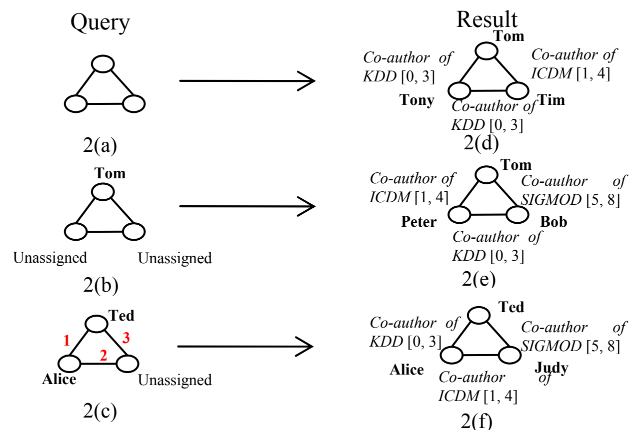


**Figure 2. Different query types and their corresponding results.**

In this work, we propose the Temporal Social Behavior Search (TSBS) problem, which aims to answer structural queries with a temporal order in an information network. The expected results are subgraph satisfying both the query topological shape and the specified temporal order on edges. The TSBS problem is related to graph matching [2][3] and graph mining[1][4][6]. However, existing methods on these works consider no temporal factors in both definitions and solutions.

We propose a two-step method, the offline and online steps, to tackle the proposed TSBS problem. In the offline step, we devise an unsupervised mechanism to identify the representative interactions, represented as frequent temporal subgraph patterns. The online step processes the user-given structural-temporal query by performing searching and matching over the discovered patterns.

From the system perspective, the proposed TSBS allow three kinds of query for tracking temporal social behaviors between individuals. The first kind is the pure structural query such as Figure 2(a) and 2(d). The second kind enables users to decide whether or not to specify the id of individuals on nodes in the query structure, Figure 2(b) and 2(e) for example. The third kind further allows users to associate the temporal order on edges of the structure query. Take Figure 2(c) and 2(f) for example, the user specifies the left edge should occur before the bottom one, and the right edge should be the latest to occur To the best of our knowledge, the proposed TSBS framework is the first attempt to answer the temporal social behaviors for the structure matching problem.

## 2. METHODOLOGY

**Offline Mining.** We regard the social activities in a certain time period as a temporal snapshot. By collecting networks in a series of periods, we can construct a transaction database of networks, in which each heterogeneous network stands for a graph recording social interactions in a certain period.

In each transaction, a relationship $c$ between individual $A$ and $B$ during timestamp [1, 2] is represented by ($A$, $c$, $B$, 1, 2). We sort the edges by the associated timestamps and transform the graph into an edge sequence. For example, the graph $g_1$ in Figure 3 can be represented as an edge sequence {($A$, $c$, $B$, 0, 2) ($B$, $c$, $D$, 1, 2) ($A$, $b$, $D$, 2, 3) ($B$, $a$, $D$, 3, 4) ($A$, $c$, $B$, 4, 6) ($A$, $b$, $C$, 7, 9)}. The edges are sorted by their start time intervals and then by their end time intervals. The temporal subgraph pattern, is defined as {($u_1$, $l_1$, $v_1$, $t_{s1}$, $t_{e1}$) ($u_2$, $l_2$, $v_2$, $t_{s2}$, $t_{e2}$) … ($u_h$, $l_h$, $v_h$, $t_{sh}$, $t_{eh}$)}, where $t_{s1}$=0, and all the edges in the pattern are sorted in increasing order. To measure the importance of a pattern, the strength of a pattern is calculated by counting its *support*, which is defined as the number of graphs containing $P$ in the heterogeneous network database. A pattern $P$ is *frequent* if its *support* is not less than *minsup*, where *minsup* is a user-specified minimum support threshold. During mining social patterns from database, we build projected databases to help we discover more frequent patterns. For example, if we have a pattern $P$=($A$, $c$, $B$, 0, 2), the corresponding projected database in $g1$ is {($B$, $c$, $D$, 1, 2) ($A$, $b$, $D$, 2, 3) ($B$, $a$, $D$, 3, 4) ($A$, $c$, $B$, 4, 6) ($A$, $b$, $C$, 7, 9)}. By scanning different projected databases from all transactions contain $P$, we can find a local pattern $e$, say, {($B$, $c$, $D$, 1, 1)}. We concatenate $P$ and $e$ to form a new pattern {($A$, $c$, $B$, 0, 2) ($B$, $c$, $D$, 1, 1)}. The concatenations are recursively performed in a depth-first search manner until no more closed frequent patterns can be found.

During the mining process, we use the closure checking and pruning strategies to reduce unnecessary candidates. The first strategy is *Same projected database removal*. If $P_1$ is a super-pattern of $P_2$ and both share the same projected database, $P_2$ is not needed to be grown because the patterns generate from $P_2$ will be not closed patterns. The second strategy is *Forward checking scheme*. A pattern $P$ is not closed if there exists a frequent pattern $e$ in $P$'s projected database, whose support is equal to $P$'s support. The third strategy is *Backward checking scheme*. A pattern $P$ is not needed to be grown if there exists a frequent pattern $e$ before $P$, whose support is equal to $P$'s support. Thus, every pattern generated from $P$ is contained by the pattern generated from concatenating $P$ and $e$ and both patterns have the same support. By applying these strategies, the closed frequent social patterns can be efficiently mined.

**Online Matching.** In the online part, our system will return patterns by the following property: A query {($qu_1$, $ql_1$, $qv_1$) ($qu_2$, $ql_2$, $qv_2$) … ($qu_m$, $ql_m$, $qv_m$)} is contained by a pattern {($pu_1$, $pl_1$, $pv_1$, $pt_{s1}$, $pt_{e1}$) ($pu_2$, $pl_2$, $pv_2$, $pt_{s2}$, $pt_{e2}$) … ($pu_n$, $pl_n$, $pv_n$, $pt_{sn}$, $pt_{en}$)} if there exists a sequence of integers $j_1 < j_2 < \ldots < j_n$ so that $qu_i = pu_{ji}$, $ql_i = pl_{ji}$, $qv_i = pv_{ji}$, $i$= 1, 2,…, $n$. We can use this property to check query existence no matter whether users assign individuals or not. If a user gives a structural query with at least one individuals and sequence order, it is still quite easy to check because edges in patterns are sorted with increasing order. Another advantage of our model is that it does not have to do the isomorphism checking during the mining process due to the sequential property of edges concatenation. Besides, the TSBS framework can return the top-$k$ support results if a user does not want to return the overwhelming numbers of results.

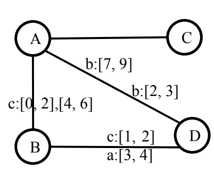


**Figure 3. The graph $g_1$.**

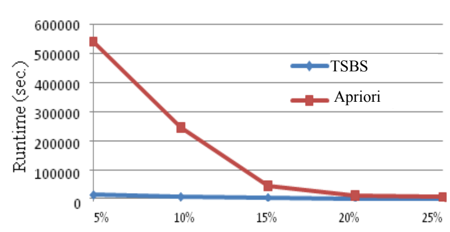


**Figure 4. Runtime versus *minsup* graph.**

## 3. EXPERIMENTAL RESULTS

We conduct the experiments using real academic datasets to show the efficiency of our framework. We modify the Apriori algorithm [5] and compare its execution time with ours. The dataset is extracted from the DBLP bibliography data, which contains multi-type "co-author-of" relationships in the conferences of data mining and database from 1970 to 2010, including SIGMOD, VLDB, ICDE, KDD, ICDM, and PAKDD. Figure 4 shows the runtime where the *minsup* varies from 5% to 25%. The TSBS's method runs faster than the modified Apriori. When the *minsup* is getting lower, the modified Apriori generates a large amount of candidates, and thus the support counting procedure is time-consuming. Moreover, the TSBS's method requires only one database scan and removes unnecessary candidates in the projected databases. Therefore, our method can outperform the modified Apriori in efficiency.

Here we demonstrate several structural queries as well as the corresponding search results in Figure 5. The following structural queries contain linear, triangle, tree, tree plus linear and double triangles.

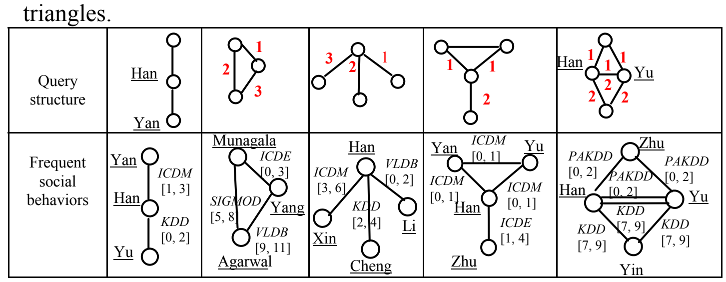


**Figure 5. The test queries and corresponding results.**

## 4. CONCLUSION

This paper presents a novel temporal social behavior search (TSBS) in an information network. The TSBS framework offline mine the temporal subgraph patterns as representative user behaviors, and then online search and match the structural query over such mined patterns. Experimental results and case studies show the efficiency and effectiveness of our TSBS framework. We believe TSBS can not only allow performing advanced social network analysis but also help people manage social circles in social networking platforms.

## 5. ACKNOWLEDGEMENTS

This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC 100-2911-I-002-001, and 10R70501.

## 6. REFERENCES

[1] A. Inokuchi, T. Washio, and H.Motoda. An Apriori-based algorithm for mining frequent substructures from graph data, *In Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, Lyon, France, 2000, pp. 13-23.

[2] J. Cheng, J. X. Yu, Bolin. Ding, P.S. Yu, and H.Wang. Fast Graph Pattern Matching. *In Proc. of IEEE International Conference on Data Engineering*, pp. 913-922, 2008.

[3] H. Tong, B. Gallagher, C. Faloutsos, and T. Eliassi-Rad. Fast Best-Effort Pattern Matching in Large Attributed Graphs. *In Proc. of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.737-746, 2007.

[4] M. Kuramochi, and G. Karypis. Frequent subgraph discovery, *In Proc. of IEEE International Conference on Data Mining*, California, USA, 2001, pp. 313-320.

[5] R. Agrawal and .R Srikant. Fast algorithms for mining association rules in large databases. *In Proc. of ACM VLDB Conference on Very Large data bases*, pp. 487-499, 1994.

[6] X. Yan, and J. Han. gSpan: Graph-based substructure pattern mining, *In Proc. of IEEE International Conference on Data Mining*, pp.721-724, 2002.