Adapting *PageRank* to position events in time

Abhijit Sahoo, Swapnil Hingmire, and Sutanu Chakraborti

Department of Computer Science and Engineering
Indian Institute of Technology, Madras
{asahoo,swapnilh,sutanuc}@cse.iitm.ac.in

Abstract. In this paper, we order events in time by using evidence present in their partial orders. We propose an algorithm named *TimeRank*, a variant of *PageRank*, for this task. *PageRank* operates on the hyperlink graph and orders the web pages according to their importance. We identify limitations of *PageRank* in the context of temporally ordering the nodes. We draw an analogy between the notion of importance in *PageRank* to the notion of *recency* in *TimeRank*. We evaluate *TimeRank* using the *Citation Graph* of scientific publications of physics and propose a baseline method to compare *TimeRank* and *PageRank*. The baseline method ranks the nodes according to their number of immediate predecessors without considering the higher order transitive relations among the events. Evaluation results suggest that *TimeRank* outperforms both the baseline method and *PageRank* in this task.

1 Introduction and Related works

In the real world, we often encounter situations where we need to order things temporally but we only have partial information about them. For example, to order past events of our life in time, we often relate them with other pivot events for which the date is known, using AFTER or BEFORE relations. We try to combine the evidence from the partial orders to arrive at an ordering of all the events. We devise an algorithm named as *TimeRank*, a variant of *PageRank* [6], that does the same by assigning a *TimeRank* score to each event. *TimeRank* can be operated on an *Event Graph* where the events are used as nodes and the temporal ordering between the events are represented as the directed edges.

TimeRank can have interesting applications in estimating the occurrence time of events in history, automatic biography compilation and creating a timeline of events from history documents. In history, there are uncertainties associated with the occurrence of events. For example, the birth year of *Gautama Buddha*, the founder of Buddhism, is uncertain[1]. The occurrence time of an event is estimated based on evidence of its associated events that are known in the history. Thus, the new evidence can potentially reorder certain events in time and the ordering can be more accurate as more evidence is used. *TimeRank* can model this progressive reduction of uncertainty. It can

[1] https://en.wikipedia.org/wiki/Gautama_Buddha

combine the evidence from higher order associations of events to arrive at an ordering of events. *TimeRank* can be applied to order biographical events extracted from the digital trails of a person like emails, search logs and his information from social medias. Similarly, we can extract events from the text of a history document and can represent them as the nodes of an *Event Graph*. The history document should present the events in a chronological order which can be exploited to obtain the partial temporal orders among the events. As the events are presented using plain text in the document, there is a high chance of presenting some events in out of chronological order. Earlier events can be referred at a later part of the document which introduces uncertainty in the partial orders of the events and results in circularity among the events in the *Event Graph*. *TimeRank* can potentially resolve the temporal order using the available evidence in the event's higher order associations.

TimeRank is built upon the *PageRank* algorithm. To best of our knowledge, no extension of *PageRank* has been proposed so far to realize temporally ordering of events. Some related works are mentioned below. Mani et al. [4] anchors the clauses from text and orders them in a sentence level. Vrotsou [7] mines the sequence of daily events to observe the evolution and trends of the events. The notion of events in [4] and [7] are entirely different from our notion of notable events. O'Madadhain et al. [5] ranks individuals on a social network using the event sequences of interactions among users in the network, Berberich et al. in [1] and Jiang et al. in [2] try to improve *PageRank* using time as a component, unlike our algorithm which orders the events in time.

2 Overview

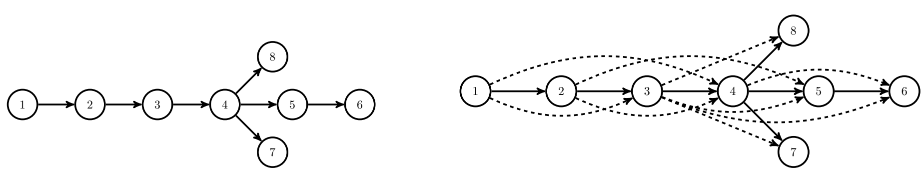
An *Event Graph*, $G = (V, E)$ where v_i is an event and $e_{ij} = 1$, if v_i has a directed edge to v_j. An *Event Graph* is similar to the hyperlink graph of the web where the web pages and their hyperlinks are analogous to the events and their order of occurrence respectively. *PageRank* [6] orders the web, according to the importance of web pages using the hyperlink graph. In the context of *PageRank*, a web page is important if it is pointed to by other important pages. Analogously, an event in an *Event Graph* is recent if it is pointed to by several recent events. In this setting, can we deploy *PageRank* on G to order the events in a chronology using the evidence present in the directed edges of G? Although an *Event Graph* is similar to a hyperlink graph, there is a significant difference. Unlike a hyperlink graph, it is important to respect transitivity in an *Event Graph*. For example, in a hyperlink graph, if a node A points to node B and B points to node C then A does not necessarily intend to point to C. This is not true in an *Event Graph*. In an *Event Graph* if an event A occurs before an event B and B occurs before an event C then one can infer that A must occur before C. The idea of *TimeRank* is motivated by this observation. The approach is presented in Section 2.1. We evaluate *TimeRank* using *Citation Graphs*, the details are in Section 2.2.

Time Marker →	Least Recent	− − − − − − →						Most Recent
Expected Rank Order (Sorted in ascending order of recency)	1	2	3	4	5	7	8	6
***PageRank* Rank Order**	1	2	5	7	8	3	6	4
***TimeRank* Rank order**	1	2	3	4	5	7	8	6

Table 1: Ranking of nodes using *TimeRank* and *PageRank* on the graph shown in Figure 1(a)

2.1 Approach

As discussed above, *PageRank* has limitations when applied to the task of ordering nodes of an *Event Graph* in time due to the implicit transitive relations among events. To address this problem, *TimeRank* explicitly adds virtual edges to the graph. The virtual edges capture transitivity of events. With the addition of virtual edges, all the indirect predecessors of any node become its direct predecessors and can pass their *recency* scores directly to it. We analyze the situation in detail using the toy example as shown in Figure 1(a). To apply *TimeRank* on the graph in Figure 1(a), we augment the graph with second order virtual edges (as shown in 1(b)) and apply *PageRank* on the augmented graph. Table 1 shows that the rank order obtained by *PageRank* for the graph as shown in Figure 1(a), varies from the expected rank order whereas *TimeRank* gives the correct order.



(a) *Event Graph* before adding virtual edges

(b) *Event Graph* after adding second order virtual edges

Fig. 1: *Event Graphs* with and without virtual edges

2.2 Evaluation

We empirically evaluate *TimeRank* on a *Citation Graph*. *Citation Graph*s are constructed using research articles as nodes and their citations as directed edges. The reason for choosing *Citation Graph*s for evaluating *TimeRank* is that they contain the required temporal properties as in the *Event Graph* i.e, the later articles cite older articles. The temporal ordering information is present in the directed edges of the *Citation Graph* as partial orders. *TimeRank* combines the evidence present in the directed edges and orders all the nodes in time. To perform the evaluation, we pretend that actual years of publications are not available, only the citation information is available regarding the *Citation Graph*. The order is obtained by *TimeRank*. The time of appearance of each node in the network is known which we use as the ground truth to compare and evaluate the rank order output of *TimeRank*.

The *Citation Graphs* are directed acyclic graphs, unlike real-world *Event Graphs* that can contain cycles. The noise and uncertainty present in the order of occurrence of the events that are extracted from plain text result in those cycles to appear, as illustrated in Section 1. To simulate the properties of real-world *Event Graph*, we introduce random noise in the *Citation Graph*. The random noise maps to the arbitrary edges which introduce uncertainty and circularity into the *Event Graph*. With adding different percentages of such noise, we operate and evaluate *TimeRank* on the *Citation Graphs*. We compare the output of *TimeRank* with a basic inlink counting (BIC) method as described below in Section 3. The BIC method orders the nodes only by the number of immediate predecessors and does not take the higher order associations into consideration. We use *PageRank* to considers the higher order associations, but it does not consider the information from the implicit transitive relations. *TimeRank* considers both higher order associations and transitive relations. The empirical results suggest that *TimeRank* outperforms both BIC and *PageRank*, in the context of ordering events in time.

3 Experiments

Data
We use the *Citation Graph* of *High-energy physics theory citation network* from *Stanford Network Analysis Project* [2]. The complete *Citation Graph* has 27770 nodes, 352807 edges, with a diameter of 13 and 90-percentile effective diameter of 5.3. We took a subgraph of the *Citation Graph* with 11444 nodes and 81088 edges for our experiments.

Baseline Method
We use basic inlink counting (BIC) as our baseline method. In this method, the nodes are ranked according to their numbers of inlinks. In our *Citation Graph*, a directed edge from node A to node B exists if B cites A. Here, the assumption is that the recent nodes will have a higher number of inlinks than older nodes. In the context of *TimeRank* the statement of circularity is: a node is recent if it is pointed to by several recent nodes. The obtained rank orders using *TimeRank*, *PageRank* and BIC are compared against the actual rank order that is obtained using the actual publication dates of articles in the *Citation Graph*. It can be seen that *TimeRank* and *PageRank* capture higher order temporal relations that BIC

[2] `https://snap.stanford.edu/`

fails to capture, and *TimeRank* can capture transitive relations that *PageRank* fails to capture.

Measure

We use *Kendall rank correlation coefficient* [3] to compare two rank orders. *Kendall rank correlation* measures the similarity between two rank orders as follows. In a rank order $(x_1, y_1), (x_2, y_2)....(x_n, y_n)$, any two pairs of observation (x_i, y_i) and (x_j, y_j) are said to be *concordant* if $x_i > x_j$ and $y_i > y_j$ or if $x_i < x_j$ and $y_i < y_j$. Any two pairs of observations (x_i, y_i) and (x_j, y_j) are said to be discordant if $x_i > x_j$ and $y_i < y_j$ or if $x_i < x_j$ and $y_i > y_j$. The *Kendall tau rank correlation coefficient* is defined as the difference between the number of concordant and discordant pairs divided by the total number of possible pairs. The range of the correlation coefficient is from -1 to $+1$. In our experiment, x_i is the obtained rank and y_i is the correct rank of an event. The range of the correlation coefficient is from -1 to $+1$.

4 Results

TimeRank(TR) and *basic inlink count (BIC)* method are operated on the *Citation Graph* with the addition of different order of transitive virtual edges. TR^k and BIC^k refer to *TimeRank* and BIC of k^{th} order respectively. TR^0 and BIC^0 do not add virtual edges to the graph and TR^0 is same as *PageRank*. BIC^1 and TR^1 add first order transitive virtual edges, and BIC^2 and TR^2 add second order transitive virtual edges to the graph. All the methods are operated on the *Citation Graph* with the addition of different percentages of random noise to it.

Kendall's tau coefficients between rank orders of different methods are compared. The results as shown in Table 2 is divided into three parts. The first part shows the comparison between BIC^0 and *PageRank* or TR^0, the second part shows the comparison between BIC^1 and TR^1 and the third part shows the comparison between BIC^2 and TR^2. We can see in all the 6 cases the performance of $TR^1 > PR$, $TR^2 > TR^1$, $TR^1 > BIC^1$ and $TR^2 > BIC^2$. Interestingly, the BIC^0 performs better than *PageRank* in 2 out of 6 cases. Some selected results of BIC^0, TR^0 and BIC^2, TR^2 are presented using the histogram as shown in Figure 2. Note that *PageRank*(PR) is same as TR^0.

5 Conclusion and Discussion

We propose an algorithm named *TimeRank* that can temporally order the nodes of an *Event Graph*. We use *PageRank* for the task, but it does not consider the information present in the implicit transitive relations among the events. To make use of this information, *TimeRank* adds virtual edges to the graph. We evaluated *TimeRank* on *Citation Graph* and compared it against *PageRank* and our proposed baseline method named as BIC. BIC does not take the higher

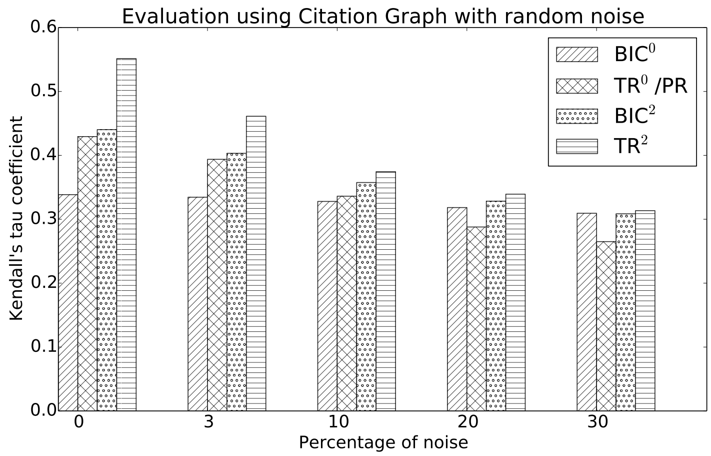


Fig. 2: Histograms showing the comparison between *TimeRank* (TR^k), *PageRank* (PR) and Basic Inlinks Count (BIC^k) method.

Percentage of Noise	**Methods**					
	BIC^0	TR^0/PR	BIC^1	TR^1	BIC^2	TR^2
0	0.3385	0.4294	0.3745	0.4941	0.4403	0.5514
3.0	0.3345	0.3938	0.3613	0.4489	0.4032	0.4612
5.0	0.3314	0.3714	0.3526	0.4209	0.3799	0.421
10.0	0.3279	0.3361	0.3418	0.3738	0.3575	0.3745
20.0	0.3183	0.2879	0.323	0.3307	0.3283	0.3393
30.0	0.3094	0.265	0.3086	0.3092	0.3085	0.3135

Table 2: Presenting the comparison between *TimeRank* (TR^k), *PageRank* (PR) and Basic Inlink Count (BIC^k) method.

order associations into consideration. *TimeRank* outperforms both *PageRank* and BIC.

TimeRank can be applied on text data of a history book to order the historical events in time. The index of the book can be used to construct an *Event Graph* as the events of a history book are mentioned on the index page along with their page number of occurrence. The events must be presented in a chronological order in the book from whose index the partial orders of the events are extracted. *TimeRank* can be used to order the events in the *Event Graph*. We can evaluate the ordering of *TimeRank* using the actual chronology of events.

References

1. K. Berberich, M. Vazirgiannis, and G. Weikum. T-rank: Time-aware authority ranking. In *Algorithms and Models for the Web-Graph*, pages 131–142. Springer, 2004.
2. H. Jiang, Y.-X. Ge, D. Zuo, and B. Han. Timerank: A method of improving ranking scores by visited time. In *2008 International Conference on Machine Learning and Cybernetics*, volume 3, pages 1654–1657, 2008.
3. M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.
4. I. Mani, B. Schiffman, and J. Zhang. Inferring temporal ordering of events in news. In *Proceedings of North American chapter of the ACL on human language technology (HLT-NAACL 2003)*, pages 55–57. ACL, 2003.
5. J. O'Madadhain and P. Smyth. Eventrank: A framework for ranking time-varying networks. In *Proceedings of the 3rd international workshop on Link discovery*, pages 9–16. ACM, 2005.
6. L. Page, S. Brin, R. Motwani, and T. Winograd. The Pagerank citation ranking: Bringing order to the web. 1999.
7. K. Vrotsou. Everyday mining: Exploring sequences in event-based data. 2010.