Using Time-Sensitive Rooted PageRank to Detect Hierarchical Social Relationships

Mohammad Jaber[1], Panagiotis Papapetrou[2], Sven Helmer[3], Peter T. Wood[1]

[1]Department of Comp. Sci. and Info. Systems, Birkbeck, University of London, UK
[2]Department of Computer and Systems Sciences, Stockholm University, Sweden
[3]Faculty of Computer Science, Free University of Bozen-Bolzano, Bolzano, Italy

Abstract. We study the problem of detecting hierarchical ties in a social network by exploiting the interaction patterns between the actors (members) involved in the network. Motivated by earlier work using a rank-based approach, i.e., Rooted-PageRank, we introduce a novel time-sensitive method, called T-RPR, that captures and exploits the dynamics and evolution of the interaction patterns in the network in order to identify the underlying hierarchical ties. Experiments on two real datasets demonstrate the performance of T-RPR in terms of recall and show its superiority over a recent competitor method.

1 Introduction

Interactions between groups of people and the patterns of these interactions are typically affected by the underlying social relations between the people. In social networks such social relationships are usually implicit. Nonetheless, analysing patterns of social interactions between the members of a social network can help to detect these implicit social relations. For example, consider a social network where the members declare explicitly some type of social relationship with others, such as x is a *colleague* of y. Now, suppose that we also have available the communication patterns between x and y, e.g., how often they exchange e-mails in a month. Using this information we may be able to infer additional relationships between these two members, such as x is the *manager* of y.

In this paper, we study the problem of detecting implicit *hierarchical* relationships in a social network by exploiting the interactions between the members of the network. We mainly focus on two key features that play a central role in our problem: (a) the structure of the interaction network, and (b) the evolution of the interactions, or in other words, the "dynamics" of the interactions over time. Given the interactions, we are interested in finding for each member of the social network their parent in the hierarchy, which we call their *superior*.

Figure 1 illustrates the problem we are addressing. The input graph (a) is the interaction network, where nodes represent actors and edges represent interactions between the actors. By analysing this network, we can infer a hierarchical relationship network (b), where nodes represent the same group of actors and edges represent the hierarchical relationships detected between them.

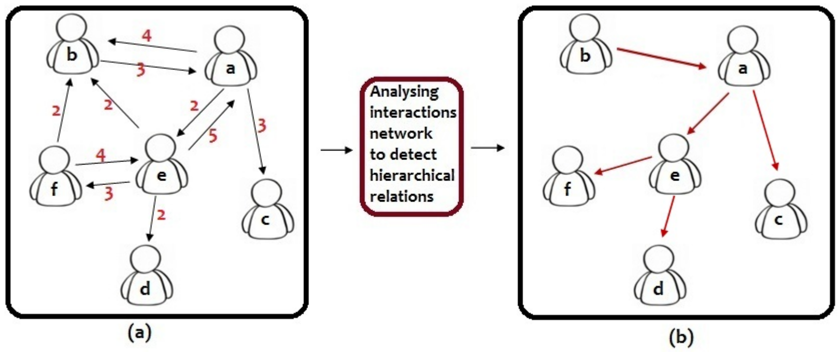

Fig. 1: (a) An interaction graph, where each arc is weighted by the total number of interactions. (b) Inferred hierarchical relationships between actors in (a).

Most related work has focussed on either the relationship structure of the network or the interactions between the nodes, but not on both. In our previous work [9], we proposed two methods for detecting hierarchical ties between a group of members in a social network. The first one, *RPR*, exploits the interaction graph of the network members and employs the Rooted-PageRank algorithm [17], whereas the second, *Time-F*, studies the interaction patterns between the network members over time.

In this paper, we propose a novel method, *Time-sensitive Rooted-PageRank* (*T-RPR*), to capture the interaction dynamics of the underlying network structure. The method proves to be more effective in detecting hierarchical ties, especially when the period over which the interactions occur is long enough.

The **contributions** of this paper include: (1) a novel time-sensitive method (T-RPR) which builds upon Rooted-PageRank and captures how the structure of the interaction network changes over time, (2) two approaches for aggregating scores from each of the time slots over which T-RPR is run, one based on a simple weighted average and the other based on voting, (3) an extensive experimental evaluation of the performance of these methods in terms of recall on two large real datasets, the Enron e-mail network and a co-authorship network. Our experiments show that T-RPR achieves considerably better results than the competitor RPR: in the Enron network, T-RPR detects up to 58% of manager-subordinate relationships, compared to only 29% by RPR, while in the co-author network it detects about 65.5% of PhD advisor-advisee relationships, a significant improvement over the 39.5% achieved by RPR.

2 Related Work

Few researchers have focussed on finding implicit ties in social networks. In our previous paper [9], we started an investigation into how the time dimension of interactions between actors could improve the detection of hierarchical ties. We defined two methods, Time-F and FiRe, which are based on predefined time functions. However, both Time-F and FiRe are ineffective in detecting hierarchical ties when there are relatively few or no interactions between the actors connected by a hierarchical relationship, a problem we address in the current paper.

Gupte at al. [6] propose an algorithm to find the best hierarchy in a directed network. They define a global measure of hierarchy which is computed by analysing the direction of interaction edges. They do not consider the temporal dimension of the interactions nor do they infer the superior of each actor, as we do. Buke et al. [3] focus on child-parent relationships at many life stages and how communication varies with the age of child, geographic distance and gender. In contrast to our approach, they model the language used between users to generate text features. Backstrom et al. [1] developed a new measure of tie strength which they termed "dispersion" to infer romantic and spouse relationships. However, dispersion does not seem relevant to our problem of detecting hierarchical relations.

On the other hand, many methods have been developed in social-network analysis to assess the importance of individuals in implicitly- or explicitly-defined social networks. Measures of importance in social networks include in-degree, degree centrality, closeness centrality, betweenness centrality and eigenvector centrality [8, 10, 13, 14, 16].

The PageRank [2] and HITS [11] algorithms have been used and adapted to address a range of problems. For example, Xiong et al. [18] evaluate a user's influence based on PageRank. They considere three factors: the number of the user's friends, the quality of their friends and the community label, i.e. the similarity between the user and the community. Fiala et al. [5] employ and adapt PageRank to analyse both co-authorship and citation graphs to rank authors by their influence. Their results are improved in [4] by introducing time-aware modifications in which citations between researchers are weighted according to a number of factors, such as the number of common publications and whether or not they were published before a citation was made.

In the same context, Yan and Ding [19] used weighted PageRank to discover author impact on a community. In the area of search engines, Li et al. [12] investigated how time-based features improve the results of retrieving relevant research publications. They consider both the structure of the citation network and the date of publication, giving older papers lower weight.

Predicting future link formation in networks has attracted much research. Huang and Lin [7] implemented an approach that considers the temporal evolution of link occurrences within a social network to predict link occurrence probabilities at a particular time in the future. Sun [15] proposed a meta path-based model to predict the future co-author relationships in a bibliographic network. Different types of objects (e.g., venues, topic) and the links among them were analysed. However, our approach differs from these two studies in that it detects *hierarchical* social ties.

3 Problem Setting

Let V denote the set of *actors* (members) of a social network. We consider two types of graphs defined over V: the *interaction graph* and the *hierarchy graph*.

Definition 1 (Interaction Graph). An *interaction graph* is defined as $G_I = (V, E^c, W)$, where E^c is the set of edges (directed or undirected) representing the interactions between the actors in V and W is a vector of edge weights, where $w_{uv} \in W$ corresponds to the weight of the edge connecting nodes u and v.

We note that G_I can be modeled both as a directed or undirected graph, as well as weighted or unweighted, depending on the nature of the interactions and the application domain at hand.

Definition 2 (Hierarchy Graph). A *hierarchy graph* is a directed graph defined as $G_H = (V, E^s)$, where $E^s \subseteq V \times V$ is a set of of edges representing the hierarchical relationship between the actors in V. Each edge $(u, v) \in E^s$ indicates that actor $u \in V$ is the direct superior of actor $v \in V$ in the hierarchy.

For example, in the context of an e-mail network among a group of employees a hierarchy graph may represent the set of manager-subordinate relationships, where $(u, v) \in E^s$ indicates that u is the manager of v. Based on the above definitions, the problem studied in this paper can be formulated as follows:

Problem 1 *Given a set of actors V and their corresponding interaction graph G_I, infer the hierarchy graph G_H of V.*

In Figure 1 we can see an example of the problem we want to solve. Given the interaction graph (a) of the five actors a, b, c, d, e, f, we want to infer their corresponding hierarchy graph (b).

4 Static Rooted-PageRank (S-RPR)

In our previous paper [9], we proposed an approach that employs Rooted PageRank (RPR) to detect hierarchical ties between a group of actors who interact over a time period. The approach relies on the fact that RPR scores reflect the importance of nodes relative to the root node. For each actor $x \in V$ (root node) we rank each other actor $y \in V \setminus \{x\}$ according to the score $RS_x(y)$ obtained by RPR, which reflects the chance of y being the superior of x. In the ideal case, the actual superior of x should have the highest score and be ranked first. The main feature of this approach is that it considers the static structure of the interaction graph over the whole time period of the interaction.

After running RPR for all actors, a ranking list $L(x) = [y_1, y_2, \ldots, y_{|V|-1}]$, $y_i \in V$, is produced for each $x \in V$, such that $RS_x(y_i) \geq RS_x(y_{i+1})$, $1 \leq i \leq n-1$. Finally, the hierarchy graph G_H is inferred from $L(x)$ by assigning to each node $x \in V$ one of the candidate managers that ranked high in $L(x)$, e.g., within the top-K places, for some K.

5 Time-Sensitive Rooted PageRank (T-RPR)

We investigate whether "time matters" in detecting hierarchical social relationships; in other words, whether significant improvements in detecting hierarchical ties can be obtained by taking into account the temporal aspects of the interactions. We adapt Rooted PageRank (RPR), as described in the previous section, and introduce *Time-Sensitive Rooted Pagerank (T-RPR)*, which captures how the ranking scores of the interactions change over time. The proposed method consists of three parts: time segmentation, ranking, and rank aggregation.

5.1 Time Segmentation

We consider the interaction graph G_I of V. As opposed to S-RPR, now G_I is not static. Let $T = [t_1, t_m]$ be the time period of interactions in G_I, starting at time t_1 and ending at time t_m. First, T is divided into n equal-sized non-overlapping time slots $\{T_1, T_2, ..., T_n\}$, with $T_j = [t_{jk}, t_{jl}]$, $\forall j \in [1, n]$, such that $t_{jl} - t_{jk} = d$, $\forall j$, where $d \in \mathbb{Z}^+$ is the size of the time segments. Observe that a time slot can be any time unit (e.g., day, fortnight, month, or year) depending on the application. Next, we define an interaction graph for each time slot.

Definition 3 (Time-Interaction Graph). A *time-interaction graph* is defined as $G_I^k = (V_k, E_k^c, W)$, where $V_k \subseteq V$ is the set of actors who interacted with at least one other actor within time slot T_k, $E_k^c \subseteq V_k \times V_k$ is the set of edges (directed or undirected) corresponding to the interactions between the set of actors V_k which took place within T_k, and W is the vector of edge weights.

Finally, a set of time-interaction graphs $\mathcal{G}_I = \{G_I^1, \ldots, G_I^n\}$ is produced for the n time slots. The next task is to rank the nodes in each graph.

5.2 Segment-based Ranking

For each time-slot T_k and each actor $x \in V$, we run RPR on the corresponding time-interaction graph $G_I^k = (V_k, E_k, W)$. Let $score_{x,k}(v_i)$ denote the RPR score of actor v_i when x is used as root on G_I^k. This results in a list of actors sorted in descending order with respect to their RPR scores at time slot k:

$$L(x)_k = [v_1, v_2, .., v_N] \ ,$$

where $N = |V_k| - 1$, $V_k \backslash \{x\} = \{v_1, v_2, .., v_N\}$, and $score_{x,k}(v_i) \geq score_{x,k}(v_{i+1})$ for $i = 1, \ldots, N-1$.

The rankings obtained over the n time-slots are aggregated for each root actor x and all remaining actors $v_i \in V$, resulting in an aggregate score $aggScore_x(v_i)$. Finally, the aggregate scores are sorted in descending order resulting in the following aggregate list of actor ranks:

$$L(x) = [v_1, v_2, .., v_M] \ ,$$

where $M = |V|-1$, $V/\{x\} = \{v_1, v_2, .., v_M\}$, and $aggScore_x(v_i) \geq aggScore_x(v_{i+1})$, for $i = 1, ...M-1$. More details on the aggregation techniques are given below.

Finally, as in S-RPR, the hierarchy graph G_H is inferred from $L(x)$ by assigning to each node $x \in V$ one of the candidate managers that ranked high in $L(x)$, e.g., within the top-K places, for some K.

5.3 Rank Aggregation

We explored two rank aggregation techniques, one based on averaging and one based on voting.

Average-based Time-sensitive RPR (AT-RPR). In this approach, the ranking in $L(x)$ is based on a weighted average of the individual RPR scores over all time-slots. We define a set of weights $\Omega = \{\omega_1, \ldots, \omega_n\}$, where ω_k is the weight assigned to time slot T_k. Each actor $y \in L(x)$ is ranked according to the obtained scores over all time-slots:

$$aggScore_x(y) = 1/n \cdot \sum_{k=1}^{n} \omega_k \cdot score_{x,k}(y) \ . \tag{1}$$

Assigning the values in Ω is application-dependent. For example, if the interactions between actors and their superiors are distributed regularly over the whole period T, then all weights can be equal. On the other hand, the interactions between actors and their superiors may be more intensive in earlier or later time-slots. An example of the former case is when detecting PhD advisor-advisee relationships in a co-author network; higher weights are given to scores in early time-slots when the advisees are expected to publish more papers with their advisors, decreasing in later time-slots.

Vote-based Time-sensitive RPR (VT-RPR). An alternative approach is to assign candidate actors with votes at each time-slot T_k based on their rank in that slot. The final rank of an actor is determined according to the total number of votes they win over all time-slots.

More precisely, given $L(x)_k$ at slot T_k, a vote is assigned to actor $y \in V \setminus \{x\}$, if y appears among the first c actors in $L(x)_k$. We call c the *vote-based cut-off*. Let $pos(L(x)_k, y)$ denote the position of y in $L(x)_k$. The total number of votes obtained by each candidate y is then defined as:

$$aggScore_x(y) = \sum_{k=1}^{n} \omega_k \cdot vote_{x,k,c}(y) \ , \tag{2}$$

where:

$$vote_{x,k,c}(y) = \begin{cases} 1 & \text{if } pos(L(x)_k, y) \leq c \\ 0 & \text{otherwise} \end{cases}$$

and ω_k is the weight of time slot T_k, which is set depending on the application.

5.4 Example

Let us consider the example shown in Figure 1. Assume that we want to detect the superior of actor a, namely, actor b. We will apply S-RPR and T-RPR, and compare the findings. We emphasise that the RPR scores used in the example are made up; however, we wish to illustrate how ranks are aggregated in our approach rather than how RPR scores are computed.

S-RPR. We set a to be the root and run RPR over the interaction graph, which produces $L(a)$. Specifically, the list contains the actors in the following order:

$$[(e, 0.30), (f, 0.25), (b, 0.20), (d, 0.18), (c, 0.07)] \ .$$

We observe that the position of actor b in $L(a)$ is 3 (out of 5).

T-RPR. Suppose that T consists of four equal-sized time-slots. We generate four time-interaction graphs, $G_I^1(V_1, E_1^c), G_I^2(V_2, E_2^c), G_I^3(V_3, E_3^c), G_I^4(V_4, E_4^c)$, one for each time-slot, as shown in Table 1.

	V_k	E_k^c
T_1	a, b, e, f	$(a, b), (e, b), (e, f), (f, e)$
T_2	a, b, c, d, e, f	$(a, b), (a, c), (b, a), (e, d), (f, e),$
T_3	a, b, e, f	$(a, b), (a, e), (e, a), (e, b), (e, f), (f, b), (f, e)$
T_4	a, c, e	$(a, c), (a, e)$

Table 1: The set of time-interaction graphs: we list the set of vertices V_k and edges E_k^c for each time slot T_k.

Next, to detect the superior of a, we run RPR with a as the root for each time-slot T_k using G_I^k for $k = 1, \ldots, 4$. A ranked list $L(a)_k$ is produced for each T_k, as shown in Table 2.

$L(a)_1$			$L(a)_2$			$L(a)_3$			$L(a)_4$		
rank	actor	$score_{a,1}$	rank	actor	$score_{a,2}$	rank	actor	$score_{a,3}$	rank	actor	$score_{a,4}$
1	$\boldsymbol{b}$	**1.00**	**2**	$\boldsymbol{b}$	**0.50**	**1**	$\boldsymbol{b}$	**0.50**	2	c	0.50
5	c	0.00	2	c	0.50	2	e	0.30	2	e	0.50
5	d	0.00	5	d	0.00	3	f	0.20	**5**	$\boldsymbol{b}$	**0.00**
5	e	0.00	5	e	0.00	5	c	0.00	5	d	0.00
5	f	0.00	5	f	0.00	5	d	0.00	5	f	0.00

Table 2: Rank lists produced by T-RPR over all time-slots with root a.

Finally, the lists are aggregated using Eq. (1) (average-based approach with weights $\omega_k = 1$) and Eq. (2) (vote-based approach with weights $\omega_k = 1$ and cut-off $c = 2$). The final aggregated lists for each aggregation approach are shown in Tables 3 and 4. For example, the score for e in Table 3 is 0.20 because its average score is $(0.30 + 0.50)/4$, while its score in Table 4 is 2 because e appears in ranks 1 or 2 in 2 time slots (3 and 4). We observe that in both cases *T-RPR* places actor b at position 1, as opposed to position 3 (*S-RPR*).

position	actor	$score_a$
1	***b***	**0.50**
2	c	0.25
3	e	0.20
4	f	0.05
5	d	0.00

Table 3: Final ranked list using the average-based approach.

position	actor	$score_a$
1	***b***	**3**
3	c	2
3	e	2
5	d	0
5	f	0

Table 4: Final ranked list using the vote-based approach.

6 Results and Analysis

We evaluated the methods in terms of recall on two datasets: the Enron email dataset and a co-authorship network, both of which are available online[1].

The Enron dataset includes more than 255000 emails exchanged among 87474 email addresses between January 2000 and November 2001. However, only 155 of these email addresses belong to Enron employees. Each email in the dataset has a sender, subject, timestamp, body, and a set of recipients. The dataset also contains the hierarchical manager-subordinate relationship between employees. The co-author dataset includes more than 1 million authors who contributed to about 80000 papers in total between 1967 and 2011. Each paper has a title, date, conference where it was published and a list of co-authors. The dataset includes hierarchical relationships between PhD advisors and their advisees.

To evaluate the performance of the two methods, we compute for each subordinate/advisee x the *rank* of their correct superior/advisor x^* in $L(x)$:

$$rank(x, x^*) = |\{y : score_x(y) \geq score_x(x^*)\}| \ , \forall y \in L(x) \ . \tag{3}$$

Hence, the rank of the manager x^* of x is the number of actors in $L(x)$ who have an RPR score greater than or equal to the score of x^* (see Table 2).

Finally, given a threshold K, we can define the *overall rank* $\rho(K)$ of V as the percentage of actors with rank at most K over all hierarchical relations that exist in G_H:

$$\rho(K) = \frac{|\{x : rank(x, x^*) \leq K\}|}{|E^s|} \cdot 100 \tag{4}$$

Enron. We excluded all email addresses of people who were not Enron employees. In cases where an employee used more than one email address, we chose one

[1] http://arnetminer.org/socialtie/

randomly. We explored two versions of the interaction graph: *directed*, where a directed edge exists from employee u to v if u sent at least one email to v; *undirected*, where any interaction (sent or received email) between u and v is represented as an edge between them. So, in both cases, all edge weights are 1.

For the T-RPR approach, each time-slot represents 1 month, giving 24 time-slots in total. In addition, since we expect to have regular interaction between a subordinate and their manager over the whole time period, each weight ω_k ($k = 1, \ldots, n$) in the aggregation functions given in Eq. (1) and (2) was set to 1.

The experimental results of the performance benchmark of AT-RPR and VT-RPR against S-RPR are shown in Figure 2(a) for the directed case and in Figure 2(b) for the undirected case. S-RPR performs considerably better for the directed graph. This becomes clear when we consider the number of managers ranked first in a subordinate's ranked list. About 30% of the managers are ranked first when using a directed graph compared to only 9.7% for the undirected graph. However, this picture changes when we consider the time dimension in T-RPR. Both AT-RPR and VT-RPR give better results on the undirected graph and especially when using vote-based aggregation. We consider the best value for the voting cut-off c for each case, i.e., 3 for the directed and 4 for the undirected graph. This finding suggests that the volume of email matters more than direction for detecting hierarchical ties in an employer-employee setting. A possible explanation is that employees may have similar communication patterns with respect to the fraction of sent vs. received emails when they communicate with other employees and also when they communicate with their boss. However, the volume of the email traffic as a whole can be a more distinctive feature of the underlying hierarchical tie.

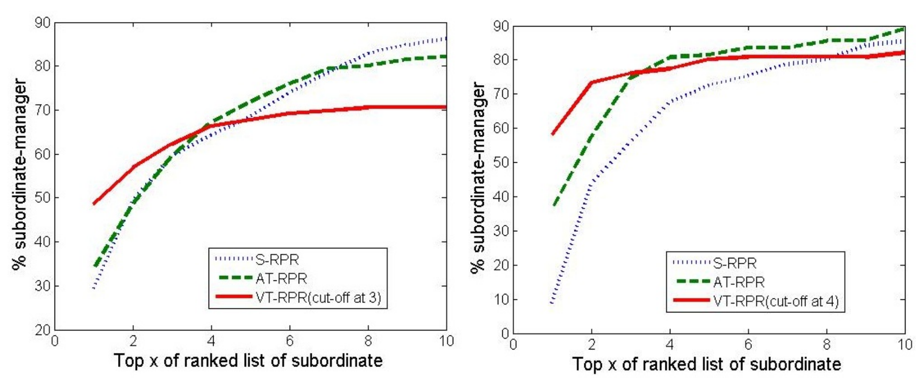

Fig. 2: Results using S-RPR and T-RPR (both aggregation strategies) on Enron using (a) the directed interaction graph and (b) the undirected interaction graph.

In Tables 5 and 6, we can see how different values of the voting cut-off c affect the performance percentages of VT-RPR. Using these tables in combination with Figure 2 we can make several observations. Firstly, for the undirected graph, we observe that VT-RPR with the voting cut-off at 4 is preferable to both S-RPR and AT-RPR with significant improvement in detecting managers who rank in the top *three* of their subordinate's lists. For example, in 58.9% of manager-subordinate relations, managers come first in the ranked lists compared to 39.8% and 9.7% detected by AT-RPR and S-RPR respectively. In addition, for the directed graph, VT-RPR is still better than both AT-RPR and S-RPR, which perform similarly. For AT-RPR and S-RPR, about 30–33% of managers are ranked first in their subordinate's lists. VT-RPR performs substantially better, detecting over 48% of managers in the top position ($K = 1$).

	ρ				
c	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
1	39.72	52.05	56.84	60.27	62.32
2	43.83	55.47	62.32	67.80	68.49
3	**48.63**	**56.84**	**62.32**	**66.43**	**67.80**
4	44.52	58.90	63.01	67.12	67.80
5	42.46	58.21	63.69	65.75	67.80
6	40.41	59.58	63.69	65.75	67.80

Table 5: VT-RPR results on Enron dataset using vote cut-off $c = 1$–6 with *directed* interaction graph.

	ρ				
c	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
1	45.20	60.95	67.80	68.49	68.49
2	54.10	67.12	75.34	78.08	79.45
3	57.53	72.60	76.02	78.08	79.45
4	**58.90**	**73.28**	**76.71**	**78.76**	**80.13**
5	58.21	73.28	76.02	77.39	80.13
6	54.79	72.60	75.34	78.08	80.82

Table 6: VT-RPR results on Enron dataset using vote cut-off $c = 1$–6 with *undirected* interaction graph.

Co-author. For the purposes of our study, we excluded all single-author papers as well as papers without a publication date. Due to the symmetric nature of the co-author relationship, the interaction graph representing this dataset is undirected. Once again, each edge weight was set to 1. For the T-RPR approach, we defined 45 time-slots, one per publication year. Moreover, the weights used by both aggregation methods were defined for each $aggScore_x(y)$, as $\omega_k = 1 - \frac{N_{first}}{N_{all}}$, where N_{first} is the number of time-slots (years) between time-slot k and the slot in which the first paper co-authored by x and y appeared, and N_{all} is the total number of time-slots between the first and last papers co-authored by x and y. We defined the weights in this way since we expect more intensive interactions between an advisee and their advisor in the early stages of the advisee's publication activity. Therefore, higher weights are given to early years.

Figure 3 depicts the performance of S-RPR, AT-RPR, and VT-RPR for Co-author. Clearly, the results for both AT-RPR and VT-RPR, are substantially better than those for S-RPR. For example, for more than 65% of the advisees, both AT-RPR and VT-RPR correctly infer their advisor as the top-ranked co-author. This gives a remarkable improvement over the results of S-RPR which only detects 39.6% of advisors correctly.

On the other hand, the results of AT-RPR are 4–5% better than VT-RPR. For instance, more than 95% and 90% of advisor-advisee relationships can be detected within the top 7 authors by AT-RPR and VT-RPR respectively. Table 7 shows that the best results for VT-RPR are with voting cut-off $c = 1$.

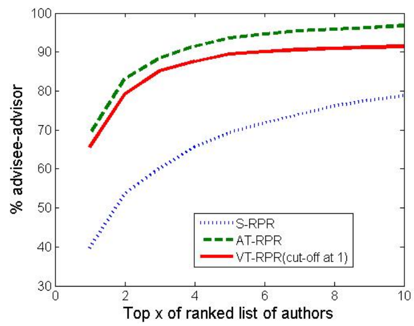

Fig. 3: Results for S-RPR, T-RPR (both aggregations) on Co-author.

	ρ				
c	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
1	**65.52**	**79.25**	**85.12**	**87.55**	**89.50**
2	60.80	78.54	84.64	88.03	89.93
3	56.84	75.58	82.30	85.78	87.88
4	53.55	72.72	79.97	83.83	86.17
5	50.69	69.48	77.25	81.78	84.21

Table 7: VT-RPR results for Co-author using vote cut-off $c = 1$–5.

Main Findings. For both the Enron and co-author datasets, the time-sensitive methods AT-RPR and VT-RPR are significantly better than S-RPR. This demonstrates that time matters when detecting hierarchical relationships in social networks. However, AT-RPR and VT-RPR perform differently on each dataset, with VT-RPR being more effective in detecting subordinate-manager relationships in the Enron data and AT-RPR being slightly better in detecting advisee-advisor relationships in the co-author network.

One interpretation of these results is that, when the interactions between actors and their superiors extend over many time-slots, then VT-RPR is more appropriate. An example of this is the Enron dataset, where the interactions occur over 24 time-slots. On the other hand, when the interactions with the superior are intensive within a few time-slots, AT-RPR is preferable to VT-RPR. This is the case for the co-author dataset where usually an advisee publishes papers with their advisor within only 4–5 time-slots while the advisee is completing their PhD. When compared to our previous work [9], our new time-sensitive methods prove to be effective in detecting hierarchical ties even when there are no, or relatively few, interactions between an actor and their superior.

7 Conclusion

We introduced T-RPR, a method for detecting hierarchical ties in an interaction graph. We investigated the impact of the temporal dimension in the ranking process and adapted Rooted-PageRank to capture the dynamics of the interactions over time between the actors in the network. We explored two variants for aggregating the rankings produced at each time slot. Experiments on two real datasets showed the superiority of T-RPR against our previous static approach, S-RPR, hence providing reasonable empirical justification for our claim that *"time matters"* in detecting hierarchical ties.

References

1. L. Backstrom and J. M. Kleinberg. Romantic partnerships and the dispersion of social ties: A network analysis of relationship status on Facebook. *CoRR*, abs/1310.6753, 2013.
2. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998.
3. M. Burke, L. Adamic, and K. Marciniak. Families on Facebook. In *ICWSM*, 2013.
4. D. Fiala. Time-aware pagerank for bibliographic networks. *J. Informetrics*, 6(3):370–388, 2012.
5. D. Fiala, F. Rousselot, and K. Jezek. Pagerank for bibliographic networks. *Scientometrics*, 76(1):135–158, 2008.
6. M. Gupte, P. Shankar, J. Li, S. Muthukrishnan, and L. Iftode. Finding hierarchy in directed online social networks. In *WWW*, pages 557–566. ACM, 2011.
7. Z. Huang and D. K. J. Lin. The time-series link prediction problem with applications in communication surveillance. *INFORMS Journal on Computing*, 21(2):286–303, 2009.
8. C. H. Hubbell. An input-output approach to clique identification. *Sociometry*, 28(4):377–399, 1965.
9. M. Jaber, P. T. Wood, P. Papapetrou, and S. Helmer. Inferring offline hierarchical ties from online social networks. In *WWW Companion '14*, pages 1261–1266, 2014.
10. Z. Katona, P. P. Zubcsek, and M. Sarvary. Network effects and personal influences: the diffusion of an online social network. *J. Marketing Research*, 48, 2011.
11. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
12. X. Li, B. Liu, and P. S. Yu. Time sensitive ranking with application to publication search. In *ICDM*, pages 893–898, 2008.
13. M. E. J. Newman. Scientific collaboration networks. II. shortest paths, weighted networks, and centrality. *Physical Review E*, 64, 2001.
14. M. E. J. Newman. The mathematics of networks. *The New Palgrave Encyclopedia of Economics*, 2007.
15. Y. Sun, R. Barber, M. Gupta, et al. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, pages 121–128, 2011.
16. V.Zlatic, G.Bianconi, et al. On the rich-club effect in dense and weighted networks. *EPJ B-Condensed Matter and Complex Systems*, 67(3):271–275, 2009.
17. S. White and P. Smyth. Algorithms for estimating relative importance in networks. In *KDD*, pages 266–275, 2003.
18. Z. Xiong, W. Jiang, and G. Wang. Evaluating user community influence in online social networks. In *IEEE TrustCom*, pages 640–647, 2012.
19. E. Yan and Y. Ding. Discovering author impact: A pagerank perspective. *CoRR*, abs/1012.4870, 2010.