Reducing the History in Decentralized Interaction-Based Reputation Systems

Dimitra Gkorou, Tamás Vinkó, Nitin Chiluka, Johan Pouwelse, Dick Epema

Reducing the History in Decentralized Interaction-Based Reputation Systems

Dimitra Gkorou, Tamás Vinkó, Nitin Chiluka, Johan Pouwelse, and Dick Epema

Delft University of Technology, Mekelweg 4, 2628CD, The Netherlands
{D.Gkorou, T.Vinko, N.J.Chiluka, J.A.Pouwelse, D.H.J.Epema}@tudelft.nl

Abstract. In decentralized interaction-based reputation systems, nodes store information about the past interactions of other nodes. Based on this information, they compute reputations in order to take decisions about future interactions. Computing the reputations with the complete history of interactions is inefficient due to its resource requirements. Furthermore, the complete history of interactions accumulates old information, which may impede the nodes from capturing the dynamic behavior of the system when computing reputations. In this paper, we propose a scheme for reducing the amount of history maintained in decentralized interaction-based reputation systems based on elements such as the age of nodes, and we explore its effect on the computed reputations showing its effectiveness in both synthetic and real-world graphs.

Keywords: decentralized reputation systems, P2P networks, computational cost

1 Introduction

A family of reputation systems useful in many Internet applications consists of interaction-based systems (also called content-driven systems [8]). These systems are based on algorithms analyzing all interactions among users and computing the reputations without using any explicit feedback from users, such as PageRank [18] for ranking web pages and Bartercast [17] for computing reputations of users in P2P systems. In interaction-based systems, the amount of historical information on the interactions maintained by each node affects the performance and the characteristics of the reputation mechanism. Networks such as popular online markets and social networks consist of hundreds of thousands or even millions of active users and thus, using the complete history for computing the reputation of nodes is prohibitive due to its resource requirements. Particularly in decentralized systems, such as file-sharing P2P systems, the available resources at nodes are limited and thus, only scalable solutions can be applied. Furthermore, a long-term history allows previously well-behaved nodes to exploit their good reputations by acting maliciously [9, 16, 21]. In this paper, we propose a scheme for reducing the amount of history maintained in decentralized interaction-based reputation systems. We experimentally explore its effect on the computed reputations using synthetic and real-world graphs.

In order to reduce the history of interactions, we use only a subset of the complete history to approximate reputations. We model the interactions of the *complete history* of a network as a growing graph with the nodes of the network as its vertices and the

interactions between pairs of nodes as its edges, and the corresponding *reduced history* as a subgraph of the complete history. The reduced history is derived from the complete history by deleting the least important edges and nodes. We define the importance of a node according to its age, its activity level, its reputation, and its position in the graph, while the importance of an edge is defined according to its age, its weight, and its position in the graph. Then we evaluate our approach using synthetic random and scale-free graphs, and two real-world graphs, one derived from the Bartercast reputation system of our BitTorrent-based P2P client Tribler [20] and the other from the author-to-author Citation network of Physical Review E journal[1]. The main difference between the Bartercast and the Citation graphs, besides their structural properties, is that the former is derived from a deployed distributed system with personalized reputations while the latter is derived from a centralized system with global reputations.

On these networks, we apply two different computations for reputations, one based on the max-flow algorithm [6] and the other based on eigenvector centrality [4]. We evaluate our approach according the following two observations: (i) for the vast majority of reputation systems, the rank of reputations is more important than the actual reputation values themselves; and (ii) in most cases the identification of the highest ranked nodes is enough. We demonstrate that the performance of the reduced history depends on the topology of the complete history. Furthermore, we show that the performance of the reduced history depends on the reputation algorithm. Finally, we conclude that reduced history can be applied in a large range of networks. Although our approach focuses on decentralized reputation systems, it can be applied to centralized systems as well.

2 Motivation and Problem Statement

Our main motivation for reducing the history of interactions in a network is the computational cost and the storage requirements of decentralized reputation algorithms. Reputation systems, such as those of eBay or Google, cover hundreds of thousands of active nodes while reputation algorithms (e.g., Eigentrust [15], PageRank [18] and max-flow based ones [6]) have a high computational complexity. In decentralized systems, like BarterCast, where each node stores and analyzes data locally using, e.g., the max-flow algorithm (with complexity $O(nm^2)$ where n is the number of nodes and m and the number of edges), even much smaller graphs of 10^6 nodes make the computation of reputations prohibitive. Taking into account that the contributions of nodes in the computation of reputations are not equal in quality and quantity [8], thus we aim to delete the least important contributions and compute reputations using only a subset of the complete history. In this way, we can reduce the computational cost significantly without decreasing the accuracy very much.

In addition to the computational cost, the dynamic behavior of many reputation systems makes the use of the complete history ineffective. In systems with a high population turnover such as P2P networks, only a few nodes remain for a long period in the system while the majority of nodes enters the system performing some interactions and then leaves it. Also a node behaving properly for a long time can build a good reputation

[1] Data available to us after request to American Physical Society

and become a traitor [16] by exploiting other nodes. Preserving only short-term history forces all nodes in the system to behave continuously according to the protocol. For these reasons, several widely used reputation mechanisms, such as those of eBay and eLance, allow the use of historical information of a 1 or 6-month window. Although using a time window is useful for such feedback-based reputation systems, it is not effective in interaction-based reputation systems since important information of highly reputed nodes is deleted.

We model the interactions of a network as a directed weighted graph $G = (V, E)$, where the vertices V represent the nodes and the edges E the interactions among the nodes. The weight of an edge represents its importance; for instance, in Bartercast, the weight of an edge between nodes represents the amount of data transferred in the corresponding direction, and in a citation graph, it represents the number of references to an author by another. The graph is dynamically growing over time and allows not only new nodes to join but also existing nodes to create new edges. The graph G represents the *complete history* (CH) of interactions in the network.

Given the growing graph G, our target is to create a subgraph of G, denoted by G', which preserves the highest ranked nodes in G and keeps the ranking of the reputations similar to that in G. The subgraph G' has to be dynamically maintained as the complete history grows while its size has to be almost fixed. The graph G' will be used for the computation of reputations, and represents the *reduced history* (RH) of interactions in the network.

3 Creating the Reduced History

The basic idea of creating the reduced history G' consists of removing the least important elements, either nodes or edges, from G. We use a node removal process in conjunction with edge removal. The ratio of removed nodes versus removed edges depends on the dynamics of the network. Nevertheless, edge removal implies node removal and vice versa. More precisely, edge removal can lead to disconnecting a node from the graph and node removal results in deleting the adjacent edges of the removed node.

The parameters for **removal of a node** consist of its age, its activity level, its reputation, and its position in the graph.

The *age of node* i is expressed as $\tau_i = t - t_i$ where t is the current time and t_i is the time instance node i joined the system. In most networks, the age of a node i affects its behavior in a non-linear way (e.g. [1, 14]). Thus, instead of its age, we consider its aging factor $f(\tau_i)$, where f is a decreasing function with $f(0) = 1$ (e.g., $f(\tau) = e^{-b\tau}$, where τ represents the age of a node and b is a constant). Keeping fresh information allows the reputations system to capture the dynamic behavior of nodes.

The *activity level* d_i *of a node* i represents its degree. Nodes with a high activity level participated in many interactions, and so, they provide much information.

The *reputation of node* i is denoted by r_i. Our aim is to preserve the information of nodes with high reputations, since these nodes are the most reliable in the network. Moreover, allowing nodes with high reputations to contribute to the computation of reputations longer is a kind of rewarding the most trusted nodes.

For node i the *importance of its position in the graph* is expressed by its betweenness centrality (BC), denoted by $C_B(i)$, which measures the sum of the fractions of the

numbers of shortest paths among all pairs of vertices that pass through node i [10]. Removing nodes from the graph can result in destroying its structure by creating many disconnected components and thus, we need to maintain the nodes that keep the graph connected.

The first three factors represent the behavior of node i while the fourth factor is added for preserving the structure of the graph during the deletion process. Therefore, in our method, the *priority score* $P_n(i)$ of deleting node i is defined as

$$P_n(i) = \alpha P_A(d_i, r_i, \tau_i) + (1-\alpha) P_B(C_B(i)), \tag{1}$$

where $P_A(d_i, r_i, \tau_i)$ expresses the priority score of deleting node i based on its activity level, aging factor and reputation, and $P_B(C_B(i))$ represents the priority score of deleting node i according to its position in the graph. The parameter α takes values in $[0, 1]$ and can be chosen according to the graph properties. We define the priority score P_A as

$$P_A(d_i, r_i, \tau_i) = \frac{n - d_i r_i f(\tau_i)}{n^2 - \sum_j d_j r_j f(\tau_j)}, \tag{2}$$

where n is the number of nodes in the graph, and the denominator acts as a normalization so that the sum of the priority scores sum to 1. Clearly, a node with a higher age, a lower activity level, or a lower reputation will be removed. Although the maximum value of $d_i r_i f(\tau_i)$ is equal to $n-1$ (corresponding to $d_i = n-1, r_i = 1$ and $f(\tau_i) = 1$), for simplicity, we approximate it to n. Similarly, P_B is expressed as $P_B(C_B(i)) = (n^2 - C_B(i))/(n^3 - \sum_j C_B(j))$. Again, even though the maximum value of $C_B(i)$ is equal to $(n-1)(n-2)$, we approximate it by n^2. When considering a single parameter for node removal, Eq.2 can be adapted in a straightforward way (similarly as P_B for parameter $C_B(i)$).

The **removal of an edge** is determined by its age, its weight, and its position in the graph.

The *age of edge* e_{ij} connecting nodes i and j, is defined similarly to the age of a node, and is denoted by $\tau_{ij} = t - t_{ij}$, where t is the current time and t_{ij} is the time of its creation. The aging factor of edge e_{ij} is a decaying function $f(\tau_{ij})$ and can be, e.g., an exponential function.

The *weight of edge* e_{ij}, denoted by w_{ij}, is one of the parameters for edge removal, since interactions with a high cost are more important for the computation of reputations, edges with high weights have to be preserved in the graph.

The *importance of the position of edge* e_{ij} in the graph is expressed by its edge betweenness centrality (BC), denoted by $C_E(e_{ij})$, which is defined as the sum of the ratios of shortest paths between all pairs of nodes containing this edge [10]. The aging factor and the weight of an edge represent its contribution to the computation of reputations, while its C_E helps in preserving the structure of the graph.

Similarly to node removal, we express the priority score of removing an edge e_{ij} as

$$P_e(e_{ij}) = \alpha P_S(w_{ij}, \tau_{ij}) + (1-\alpha) P_F(C_E(e_{ij})), \tag{3}$$

where α is the parameter used in the definition of P_n to control the topology of the derived graph. The scores P_S and P_F are defined similarly to P_A and P_B, respectively. Therefore, edges with lower age, lower weight, and lower betweenness centrality will be removed.

The basic computational components of reducing the history consist in the computation of BC (we do not distinguish between node and edge BC because the algorithm is the same). Computing the degree, the aging factor of nodes, the weight, and the aging factor of edges has a linear cost on the number of nodes and edges respectively and can be computed incrementally. However, the computational cost of BC is high (for unweighted networks it is $O(mn)$ where n is the number of nodes in the network and m the number of edges). The cost can be significantly reduced by using approximations [11] and exploiting the structure of the network. In particular in scale-free networks, the BC values do not have to be updated very often with the network growth [12] and in networks without community structure, the BC of a node shows a strong correlation with its degree. Note that the reputations of nodes are computed by the core reputation mechanism.

4 Datasets

In order to assess our method for creating the complete history, we consider both synthetic graphs and graphs derived from real networks. In our synthetic complete history graphs we consider two processes occur simultaneously: first, new nodes enter the system, and secondly, the already existing nodes interact, thus creating new links. Thus, we define the probability p_c which represents the probability of adding a new node at each time step to the graph, and the probability $1 - p_c$ which represents the probability of adding new links between existing nodes. In highly dynamic systems, the appearance of new nodes is dominant, and so the value of p_c is high. In our models for synthetic graphs, we allow the occurrence of multiple edges between a pair of nodes and we consider the number of multiple edges as the weight of that edge.

For our experiments, we create the complete history G and the corresponding reduced history G' in parallel. In the complete history, we store all the new information. For the construction of the reduced history we keep its size (almost) constant to a maximum number of nodes n_{max}, which represents the computational or memory limitation of the system. We control the size of the reduced history by removing nodes or edges from the graph as new information is stored as described in the previous section. Below, we describe in detail our models for random graphs and scale-free graphs, the properties of the Bartercast and Citation graph, and the construction of the corresponding reduced histories.

A **random graph**, denoted by $R(n, p_r)$, is composed of n nodes, and each potential edge connecting two nodes occurs independently with probability p_r. Based on this model, we generate a growing directed random graph $R(n_t, p_r)$ representing the complete history of interactions.

To create the graph $R(n_t, p_r)$ with n_t nodes at time t, starting from a single node, we perform the following two operations at each time step:

- With probability p_c we add a new node with each of its potential directed edges existing with probability p, for some value of p.
- With probability $1 - p_c$ we add pn_t new directed edges adjacent to chosen existing nodes uniformly at random.

For the proof of $p_r \sim p/2p_c$ the reader is referred to Appendix A. In accordance with R, we create the reduced history graph R'. The reduced history R' is equal to R up to the

maximum number of nodes n_{max}. After having reached n_{max} nodes, R' is maintained by performing the following operations at each time step:

- When a new node is added to R, we also add this node to R' along with its edges, and then we remove one node together with its edges with the highest priority score (Eq. (1)).
- When new edges are added to R, we add the same edges to R'. Then we remove from R' the same number of edges with the highest priority score (Eq. (3)).

Note that some edges in R may be adjacent to nodes that have been removed from R'; in this case, these edges are not added to R'.

Scale-free graphs are characterized by their degree distribution following a power law. We create a growing directed scale-free graph based on the preferential attachment model [3]. Similarly to the procedure for random graphs, we generate two directed graphs S and S' corresponding to the complete history and the reduced-history.

We create $S(n_t)$ by starting with a small seeding graph with m_0 nodes connected by $m_0 - 1$ edges and then performing the following steps:

- With probability p_c we add a new node with m directed edges, with $m \leqslant m_0$. Each edge is adjacent to an already existing node i with probability $\Pi(i) = d_i / \sum_j d_j$, where d_i is the degree of node i.
- With probability $1 - p_c$ we add m new directed edges. Each of these edges are adjacent to an existent node i with probability $\Pi(i)$.

One can show that S is scale-free with power-law exponent equal to $\gamma = 1 + 2/(2 - p_c)$ (see Appendix B for the proof). In line with S, we build the reduced history S' using the same procedure as for random graphs.

The **Bartercast graph** is derived from Bartercast [17], the distributed reputation mechanism used in our BitTorrent-based client Tribler [20]. In Bartercast, when a peer exchanges content with another peer, they both store a *record* with the amount of data transferred and the identity of the corresponding peer. Regularly, peers contact another peer to exchange records using a gossip-like protocol. From the records it receives, every peer i dynamically creates a weighted, directed *subjective graph*, the nodes of which represent the peers about whose activity i has heard through Bartercast records, and the edges of which represent the total amount of data transferred between two nodes in the corresponding directions.

We have crawled the Tribler system from September 1, 2010 to January 31, 2011, collecting information from 29,716 nodes. In our experimental analysis, we will assume *full-gossip* in which peers forward the records they receive from other peers, and so all peers eventually receive all the propagated records. Thus, the graph derived from Bartercast, denoted by B, can be considered as the subjective graph of all nodes which corresponds to the complete history. The graph B is not connected and so, we proceed in the analysis using its largest weakly connected component. Bartercast presents high population turnover and thus, the derived graph consists in a dense core with very few long living and active nodes and a periphery with many loosely connected nodes of low activity (small average path length and small clustering coefficient, see Table 1). The addition of new nodes/edges in B is based on the actual timestamps of the crawled database of Bartercast. Similarly to the procedure for random and scale-free graphs, we maintain the reduced history B' by removing nodes and edges using Eqs. (1) and (3) as new nodes and edges are added according to the timestamps.

Table 1: The average path length (L) and the clustering coefficient (cc) of the largest connected component of the Bartercast and Citation graph, and of the corresponding random graphs with similar average path length.

Graph	# Nodes	# Edges	L	cc	L_{rand}	cc_{rand}
Bartercast	$10,634$	$31,624$	2.64	0.00074	2.63	0.0032
Citation	$15,360$	$365,319$	3.29	0.1098	3.31	0.0012

The author-to-author **Citation graph**, denoted by C, is derived from the citation network of 21,858 papers published in Physical Review E from January 2001 to November 2011. Its vertices represent the authors of papers and edges represent the citation relationship between two authors (or coauthors). The weight of an edge indicates multiple citations from one author to another. Unlike Bartercast, the graph C is derived from a centralized system with global reputations. In Table 1, we can see that graph C exhibits small-world behavior with small average path length and large clustering coefficient. Its degree distribution has a power-law tail with exponent $\gamma = 2.55$. As described for the Bartercast graph, we create the complete history C and the corresponding reduced history C' based on the actual timestamps in the database of the Citation graph.

5 Computation of Reputations and Evaluation Metrics

We consider two methods for computing reputations: the max-flow algorithm and the eigenvector centrality. However, our approach can be generalized to other methods for computing reputations as well.

The **max-flow algorithm** [6] computes the maximum flow passing between two nodes and is the core of many reputation systems (such as Bazaar [19], Bartercast [17], and the system proposed by Feldman et al. [9]) because it provides resilience to misreporting by nodes who may exaggerate their contributions to increase their reputations. In our study, we use the definition of reputation of Bartercast mechanism [12] since we use a graph derived from it for the evaluation of our approach. The reputation of a node j is computed as $\arctan(f_{ji} - f_{ij})/(\pi/2)$, where node i represents the node with the maximum betweenness centrality, f_{ji} represents the maximum flow from node j to node i in the network and f_{ij} is the maximum flow in the reverse direction. The function $\arctan$ in the computation of reputations emphasizes the differences of flows close to 0 (neutral reputation), so that newcomers with only a small contribution can achieve a good reputation value and participate in the system. Every reputation value is normalized with the factor $\pi/2$ so that it is in $(-1, 1)$.

Eigenvector centrality is a well-studied metric for the importance of a node in a network and its variants constitute the core of many reputation and recommendation mechanisms (such as EigenTrust [15], PageRank [18], TrustRank [13] and many others). The basic idea of eigenvector centrality is that interactions with highly reputed nodes contribute more to the reputation of a node. In our analysis, we use PageRank computed using the power iteration: $r_{t+1} = dAr_t + [(1-d)/N]\mathbf{1}$, where A represents the normalized adjacency matrix of the network, r_t the ranking vector at time step t, d the damping factor (we set it equal to its typical value 0.85 [18]), N the number of

nodes, and **1** the vector of length N containing only ones. In some networks like Bartercast, an incoming edge of a node has a negative meaning for the reputation of that node (because a weighted edge represents the amount of transferred data and so, adds to the reputation of the donator of the data). Therefore, in these networks, first we reverse the direction of links before we apply PageRank (reverse PageRank [2]).

The evaluation of our method is based on the observations that for the vast majority of reputation systems, the ranking of nodes according to their reputations is more important than the actual reputation values themselves, and that in many systems the identification of the highest ranked nodes is more important than of the rest of the nodes. Therefore, we define the *ranking error* as the difference between the rankings of the nodes according to their reputations in the reduced history and the complete history. More precisely, we consider the sequences of the Unique Identifiers (UIDs) of the nodes in the reduced and the corresponding complete history of our graphs, and we compute the minimum number of inversions of consecutive elements needed in the sequence of the reduced history to get all the common nodes in their correct order in the complete history. This minimum number of inversions is then normalized over the worst case, which would occur if the ranking would be completely reversed. Furthermore, to explore the ability of the reduced history to identify the highest ranked nodes, we define a second metric called the *ranking overlap* which is defined as the fraction of nodes the sequences of the top-5%, 10% and 20% ranked nodes in the reduced history and the corresponding sequences in the complete history have in common. More precisely, we compute the ranking overlap as $|\mathcal{U} \cap \mathcal{V}|/|\mathcal{U}|$, where $\mathcal{U}$ is the set of the top-5%, 10% and 20% ranked nodes in the reduced history and $\mathcal{V}$ is the set of the top ranked nodes in the complete history of size $|\mathcal{V}| = |\mathcal{U}|$.

6 Evaluation

In this section, we present our experimental evaluation. First, we explore the effect of each of the parameters for node and edge removal separately and in combination. Next, we study the effect of the size of the reduced history relative to the size of the complete history. Finally, we evaluate the effect of the growth of the complete history while the size of the reduced history is constant. In our experiments, we use the synthetic and real-world graphs introduced in Section 4. Our synthetic graphs consist of $5,000$ nodes with α and p_c neutral (both equal to 0.5), unless other initializations are mentioned. We choose the other parameters for the random graph ($p_r = 0.02$) and the scale-free graph ($m = 3$ and $\gamma = 2.2$) so that they roughly correspond to the Bartercast graph. For the synthetic graphs, our results presented in each plot are the average of 25 independent experiments, while for the Bartercast and Citation graphs, we conduct only one experiment since we have only one instance of these real-world graphs.

Experiments and Results We first explore the effect of the parameters for node and edge removal defined in Section 3 on the ranking error. To explore the effect of the parameter α, we remove 50% of the nodes and edges of the complete history according to Eqs. (1) and (3) for different values of α. Due to space limitations we omit the corresponding figure. We find that α does not affect the performance of the reduced history much. In particular, for random graphs using max-flow (or Pagerank), the ranking error

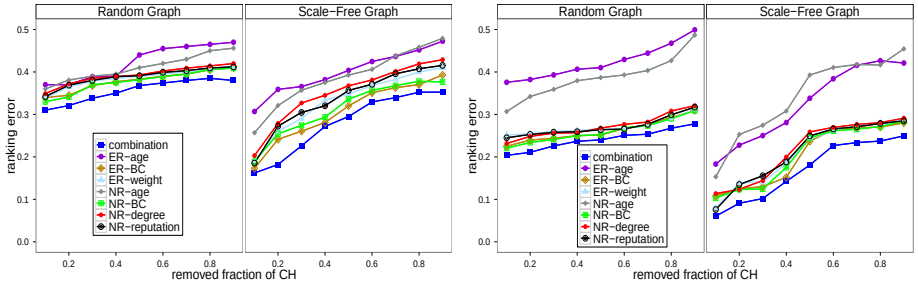

Fig. 1: The effect of the parameters for node and edge removal when removing a fraction of the nodes and edges of CH for random and scale-free graphs when the reputation algorithm is max-flow (left) and Pagerank (right). The indication ER in the legend denotes parameters for edge removal and NR parameters for node removal.

starts at 0.33 (or 0.21) for α equal to 0, and it slightly decreases by 0.02 (or 0.01) until α is equal to 0.8. As α increases further, the ranking error increases by 0.07 (or 0.06). A similar stable behavior for the ranking error is observed for the scale-free and real-world graphs. Since α doesn't affect the performance of the reduced history much we take it as neutral, equal to 0.5, for all the following experiments.

Next, we explore the effect of the parameters for node and edge removal separately, and their combination as defined by Eqs. (1) and (3). For the parameters for node or edge removal, we remove fractions nodes or edges of the complete history using only one parameter at a time. The effect of these parameters on the ranking error is plotted in Fig. 1 for the random and scale-free networks. We observe that creating the reduced history using only node removal results in similar performance as edge removal for the corresponding parameters. This is to be expected as there is a correlation between these parameters: in general, an edge with high BC is adjacent to nodes with high BC, an old edge is attached to old nodes, and an edge with a large weight is adjacent to a node with high reputation. Furthermore, the combination of all parameters in Eqs. (1) and (3) results in the smallest ranking error. The largest ranking error occurs when we remove nodes based on their age. The reputation of a node depends on the period it participates in the system and thus, when only new nodes with low reputations participate in the reduced history, the ranking error is high. All the other parameters cause quite similar ranking errors because they exhibit correlations in graphs without strong community structure, such as the random and scale-free graphs. In the real-world graphs, the parameters for node and edge removal and their combination exhibit similar relative performance as in the scale-free graphs. We omit the plot for the real-world graphs due to space limitations. Since the combination of the parameters for node and edge removal achieves the lowest ranking error, we use it to create the reduced history for all the following experiments.

We next evaluate the effect of the size of the reduced history relative to the size of the complete history on the ranking error and the ranking overlap. For this purpose, we construct reduced histories of different sizes for a complete history of fixed size as described in Section 4. Fig. 2 (left) plots the ranking error for different relative sizes

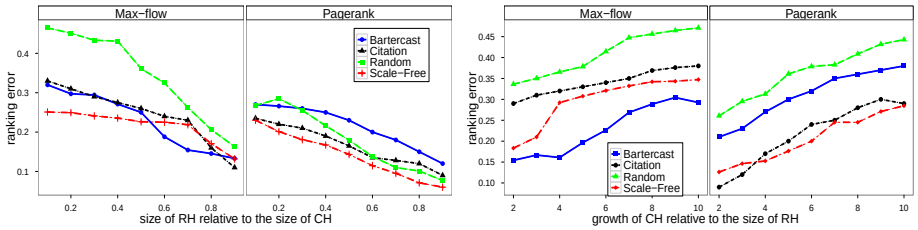

Fig. 2: The effect of the size of RH relative to the size of CH (left) and the effect of the growth of CH relative to the size of RH (right).

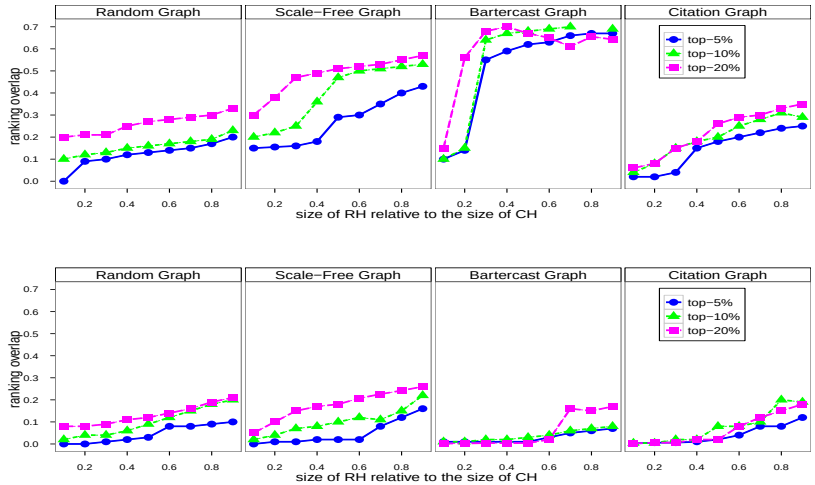

Fig. 3: The effect of the size of RH relative to the size of CH for max-flow (top) and Pagerank (bottom).

of the reduced history. We observe that when using max-flow, the scale-free, Bartercast and Citation graphs exhibit much smaller ranking error than the random graphs. For all the graphs using Pagerank, the reduced history exhibits smaller ranking error than using max-flow. Fig. 3 plots the ranking overlap for different relative sizes of the reduced history. The scale-free and Bartercast graphs exhibit much higher ranking overlap than the random and Citation graphs when using the max-flow based algorithm. Particularly, in these networks the ranking overlap decreases quite slowly with the decrease of the size of the reduced history, until the size of the reduced history is about 0.4 of the complete history. The reason is that these networks have a large amount of redundant information for approximating the highest ranked nodes when using the max-flow algorithm. When the size of the reduced history is smaller than 0.3 of the complete history, the ranking overlap degrades quickly. With Pagerank, the reduced history exhibits very low ranking overlap for all the graphs.

Finally, we evaluate the effect of the growth of the complete history while the reduced history is of constant size on the ranking error and the ranking overlap. For the synthetic graphs, we let the complete history grow from 500 to $5,000$ nodes while we

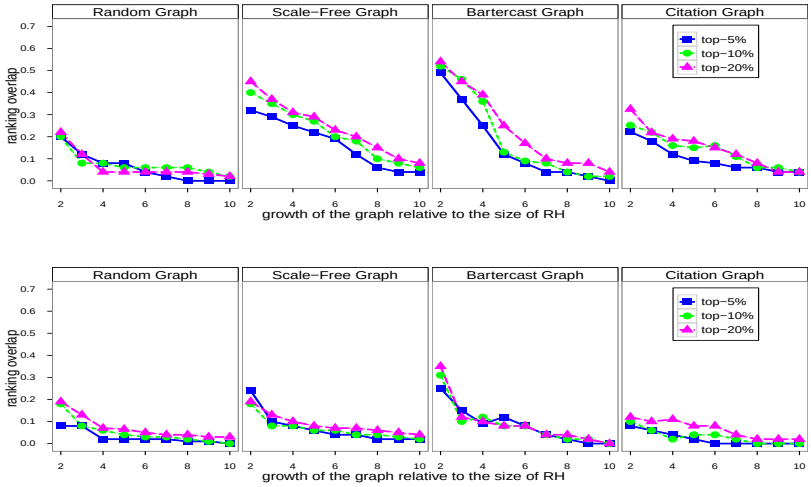

Fig. 4: The effect of the growth of CH relative to the size of RH for max-flow (top) and Pagerank (bottom).

keep the size of the reduced history constant at 500 nodes. For the real-world graphs, using the available temporal information, we have the Bartercast graph grow from $1,063$ to $10,634$ nodes with the reduced history constant at $1,063$, and the Citation graph from $1,536$ to $15,360$ nodes with the reduced history at $1,536$. Fig. 2 (right) plots the ranking error and Fig. 4 plots the ranking overlap for different relative growths of the complete history. We observe again that Pagerank achieves a smaller ranking error while the max-flow based algorithm achieves a better ranking overlap, specially for the scale-free and real-world graphs.

Discussion The observations arising from our experiments indicate that the reduced history can give a good approximation of the ranking of nodes according to their reputations when the complete history exhibits a particular structure. In this subsection, we explain and discuss our main observations.

First, we observe that constructing the reduced history using the combination of all the parameters for node and edge removal results in the lowest ranking error. Considering only parameters such as degree and reputation gives priority for removal to the newest nodes and so, new nodes will not participate in the reduced history. On the other hand, considering only the age as parameter for removal results in high ranking error because then, only new nodes participate in the reduced history and information of old important nodes has been removed. Therefore, for good performance of the reduced history, it is required to use a combination of these parameters as defined by Eqs. (1) and (3).

Secondly, the performance of the reduced history depends on the topology of the graph, and is better in the scale-free, Bartercast and Citation graphs than in the random graphs. The scale-free and our real-world graphs have only a few well connected nodes accumulating the majority of links, while the vast majority of nodes has a very low connectivity. In the reduced history, the highly connected nodes are preserved keeping their

good ranking position, while most of the loosely connected nodes have been removed. In contrast, in random graphs all nodes have stochastically similar connectivity properties. Since most real networks exhibit heterogeneity in the connectivity properties of their nodes [1], we can conclude that the reduced history can be applied in a large range of networks.

Finally, the performance of the reduced history depends on the reputation algorithm used. In particular, it causes a lower ranking error when using Pagerank, while it achieves a higher ranking overlap when using max-flow. Pagerank computes the reputation of a node by aggregating the interactions of all nodes participating in a graph. The aggregative computation of centrality by Pagerank achieves lower ranking error even if the reduced history has a relatively small size. Unlike Pagerank, the max-flow based algorithm computes the reputation of a node taking into account only the interactions between that node and the most central node. Since both the most central and the highest ranked nodes are considered as important, they are preserved in the reduced history. Therefore, we achieve a high ranking overlap when using the max-flow based algorithm.

In conclusion, our observations demonstrate the effectiveness of the reduced history in approximating the ranking of nodes with Pagerank and in identifying the highest ranked nodes with the max-flow based algorithm. This implies that the reduced history can approximate with reasonably accuracy the complete history in real world graphs, while it has much smaller resource requirements. As we stated in Section 2, this result is valuable especially for decentralized systems, such as Tribler, because of the limited resources available at each node.

7 Related work

The observations of our experiments are consistent with the findings of prior published research for the robustness of centrality measures under sampling or missing data. In particular, our finding that node and edge removal cause similar ranking errors has been discussed in the context of the robustness of centrality measures under missing data [5]. In the context of network sampling, it has been observed that ranking nodes with eigenvector centrality is highly robust [7]. Moreover, the result that the use of BC for node and edge removal does not affect the ranking error much, has been also observed under the context of edge removal for security reasons [22]. However, our approach is different from sampling techniques, since sampling techniques focus on creating a static subgraph with similar properties as the original graph. In our case, we need to maintain the reduced history dynamically with the growth of the original graph, and we are interested in producing a reduced history that preserves the reputations of nodes and not necessarily the general properties of the original graph.

8 Conclusion and Future Work

Using the complete history of interactions in a reputation system is not efficient due to its high computational cost and high memory requirements, and to the high population turnover. We have proposed the use of the reduced history instead of the complete history defining the main parameters for choosing the nodes participating in it. Next,

we have evaluated our approach experimentally exploring both theoretical graph models and real-world graphs using two reputation algorithms, a max-flow based algorithm and Pagerank. We conclude that for scale-free and real-world graphs, the reduced history is reasonably accurate while for random graphs, due to their structural properties, the reduced history causes high error. Furthermore, we have demonstrated that using the max-flow based algorithm results in better identification of the highest ranked nodes while using Pagerank results in better ranking error. An interesting direction for future work is the evaluation of the reduced history using only the local information available at each node in a decentralized environment.

References

1. Amaral, L.A.N., Scala, A., Barthélémy, M., Stanley, H.E.: Classes of small-world networks. Proc. Natl. Acad. Sci. U.S.A (2000)
2. Bar-Yossef, Z., Mashiach, L.T.: Local approximation of pagerank and reverse pagerank. In: ACM SIGIR (2008)
3. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science (1999)
4. Bonacich, P.: Eigenvector-like measures of centrality for asymmetric relations. Social Networks (2001)
5. Borgatti, S., Carley, K., Krackhardt, D.: On the robustness of centrality measures under conditions of imperfect data. Social Networks (2006)
6. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. The MIT Press (1990)
7. Costenbader, E., Valente, T.: The stability of centrality measures when networks are sampled. Social Networks (2003)
8. De Alfaro, L., Kulshreshtha, A., Pye, I., Adler, B.T.: Reputation systems for open collaboration. Commun. ACM (2011)
9. Feldman, M., Lai, K., Stoica, I., Chuang, J.: Robust incentive techniques for peer-to-peer networks. In: ACM EC (2004)
10. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry (1977)
11. Geisberger, R., Sanders, P., Schultes, D.: Better approximation of betweenness centrality. In: ALENEX (2008)
12. Gkorou, D., Pouwelse, J., Epema, D.: Betweenness centrality approximations for an internet deployed p2p reputation system. In: IEEE IPDPSW (HotP2P) (2011)
13. Gyongyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank (2004)
14. Hajra, K., Sen, P.: Aging in citation networks. Physica A: Statistical Mechanics and its Applications (2005)
15. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In: WWW (2003)
16. Marti, S., Garcia-molina, H.: Taxonomy of trust: Categorizing p2p reputation systems. Computer Networks (2006)
17. Meulpolder, M., Pouwelse, J., Epema, D., Sips, H.: Bartercast: A practical approach to prevent lazy freeriding in p2p networks. In: IEEE IPDPS (Hot-P2P) (2009)
18. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (1999)
19. Post, A., Shah, V., Mislove, A.: Bazaar: Strengthening user reputations in online marketplaces. In: NSDI (2011)
20. Pouwelse, J.A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Iosup, A., Epema, D.H.J., Reinders, M., van Steen, M.R., Sips, H.J.: Tribler: a social-based peer-to-peer system. Concurr. Comput.: Pract. Exper. (2008)

21. Xiong, L., Liu, L.: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE TKDE (2004)
22. Zeng, A., Cimini, G.: Removing spurious interactions in complex networks. arXiv:1110.5186v1 (2011)

A Proof for Probability of Connectivity for Random Graphs

We start with one initial node with no edges. Then, we start building our graph as described in Section 4, and at time $t > 0$, the expected number of nodes is $n_t = 1 + p_c t$. Since the probability of connection is p, the expected number of edges at time t is: $E(t) = \sum_t p n_t = \sum_t p(1 + p_c t) = p(t + p_c t(t-1)/2)$. Thus, the probability of connection in the random graph is equal to $E(t)/(n_t(n_t - 1)) \approx (p p_c t^2/2)/(p_c^2 t^2) = p/2p_c$ for large t, which proves that our procedure described in Section 4 creates a random graph $R(n_t, p/2p_c)$.

B Proof for the Exponent of our Scale-free Graphs

The proof of S being a scale-free graph is based on the mean-field theory proposed by Barabási and Albert [3]. With $p_c = 1$ we have the classic Barabási-Albert model, where only a new node is added and the exponent of power-law is $\gamma = 3$. We start with one initial node and then, to construct our scale-free graph, we follow the constructive process described in Section 4. With probability p_c we add a new node with m edges, and so the degree of node i, denoted by d_i, changes with rate: $\partial d_i/\partial t = m d_i / \sum_j d_j$. With probability $1 - p_c$ we add m new directed edges and the degree of node i changes with rate: $\frac{\partial d_i}{\partial t} = 2m d_i / \sum_j d_j$. Therefore, in total:

$$\frac{\partial d_i}{\partial t} = p_c m \frac{d_i}{\sum_j d_j} + (1 - p_c) 2m \frac{d_i}{\sum_j d_j} = (2 - p_c) m \frac{d_i}{\sum_j d_j}. \tag{4}$$

Moreover, $\sum_j d_j = 2E(t) = 2mt$, where $E(t)$ is the number of edges in the graph at time t, so we can solve Eq. (4) for d_i and find:

$$d_i = m \Big(\frac{t}{t_i} \Big)^{(2-p_c)/2}, \tag{5}$$

where t_i represents the time that node i joined the network. Using Eq. (5), the probability $P[d_i(t) < d]$, that a node i has a connectivity d_i smaller than d, can be written as $P[d_i(t) < d] = P\Big(t_i > (m/d)^{2/(2-p_c)} t\Big)$.

We assume that each operation of either adding a new node or a set of edges takes one unit of time, and so the probability density of t_i is $P_i(t_i) = 1/(m_0 + t_i)$. Thus,

$$P\Big(t_i > \Big(\frac{m}{d}\Big)^{2/(2-p_c)} t\Big) = 1 - P\Big(t_i \leq \Big(\frac{m}{d}\Big)^{2/(2-p_c)} t\Big) = 1 - \Big(\frac{m}{d}\Big)^{2/(2-p_c)} \frac{t}{m_0 + t}.$$

The degree distribution is the probability density for $P(d)$, thus we obtain:

$$P(d) = \frac{\partial P[d_i(t) < d]}{\partial d} = \frac{2m^{2/(2-p_c)}}{(2 - p_c)} \frac{1}{d^{2/(2-p_c)+1}} \frac{t}{(m_0 + t)},$$

and as a consequence, for large t, $P(d) \sim d^{-\gamma}$ with $\gamma = (2/(2 - p_c) + 1)$.