Information Processing and Management xxx (xxxx) xxxx

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Boosted seed oversampling for local community ranking

Emmanouil Krasanakis[*,a,b], Emmanouil Schinas[a], Symeon Papadopoulos[a], Yiannis Kompatsiaris[a], Andreas Symeonidis[b]

[a] CERTH-ITI, 6th km Charilaou-Thermi Rd, 57001 Thermi, Thessaloniki, Greece
[b] AUTH, Department of Electrical and Computer Engineering, 54124 Thessaloniki, Greece

ARTICLE INFO

Keywords:
Network ranking
Local communities
Seed oversampling
Unsupervised rank boosting

ABSTRACT

Local community detection is an emerging topic in network analysis that aims to detect well-connected communities encompassing sets of priorly known seed nodes. In this work, we explore the similar problem of ranking network nodes based on their relevance to the communities characterized by seed nodes. However, seed nodes may not be central enough or sufficiently many to produce high quality ranks. To solve this problem, we introduce a methodology we call *seed oversampling*, which first runs a node ranking algorithm to discover more nodes that belong to the community and then reruns the same ranking algorithm for the new seed nodes. We formally discuss why this process improves the quality of calculated community ranks if the original set of seed nodes is small and introduce a boosting scheme that iteratively repeats seed oversampling to further improve rank quality when certain ranking algorithm properties are met. Finally, we demonstrate the effectiveness of our methods in improving community relevance ranks given only a few random seed nodes of real-world network communities. In our experiments, boosted and simple seed oversampling yielded better rank quality than the previous neighborhood inflation heuristic, which adds the neighborhoods of original seed nodes to seeds.

1. Introduction

Detecting communities in graphs (Fortunato, 2010; Fortunato & Hric, 2016) such as social media networks (Leskovec, Lang, & Mahoney, 2010; Papadopoulos, Kompatsiaris, Vakali, & Spyridonos, 2012; Yang & Leskovec, 2012) is a well-known network analysis problem that aims to identify groups of structurally similar network nodes. Communities in real-world networks are often overlapping (Palla, Derényi, Farkas, & Vicsek, 2005; Raghavan, Albert, & Kumara, 2007; Reid, McDaid, & Hurley, 2011; Xie, Kelley, & Szymanski, 2013), meaning that a network node may belong to more than one communities. In this case, the classification problem of which nodes belong to which community can be solved for each community independently.

Besides identifying members of a community, it is often important to rank network nodes based on their relevance to that community. This task is particularly important in large social networks, where boundaries between communities are less clearly defined (Leskovec, Lang, Dasgupta, & Mahoney, 2009). Generally, node ranking algorithms aim to produce ranks that inform users of how central or influential nodes are with respect to each community. This way, one can glean insights whose intricacies are lost to binary predictions.

Most community ranking approaches (see Section 1.1) follow unsupervised schemes to extract network nodes that are expected to

* Corresponding author at: CERTH-ITI, 6th km Charilaou-Thermi Rd, 57001 Thermi, Thessaloniki, Greece.
E-mail addresses: maniospas@iti.gr (E. Krasanakis), manosetro@iti.gr (E. Schinas), papadop@iti.gr (S. Papadopoulos), ikom@iti.gr (Y. Kompatsiaris), asymeon@eng.auth.gr (A. Symeonidis).

https://doi.org/10.1016/j.ipm.2019.06.002
Received 15 October 2018; Received in revised form 17 April 2019; Accepted 3 June 2019

be central to community structure. These are commonly referred to as *seed nodes*. Community ranking algorithms then produce ranks pertaining to the communities characterized by these nodes. However, seed nodes can also be manually defined according to real-world needs. For example, they may be user queries or expert opinions on which network nodes are most closely related to an unknown community.

Unfortunately, in many settings where seed nodes that characterize the communities of interest are manually selected, they comprise only a small fraction of the community. This is often caused by the multifaceted nature of real-world networks, where too many overlapping communities lead to a lot of network nodes missing a significant portion of community labels. Additionally, the cost of manually discovering central community nodes and/or labeling adequately many community nodes (e.g. through consulting an expert or a computationally or monetarily expensive label retrieval interface) can be prohibitive in larger networks.

For example, let us consider a network analyst whose task is to rank authors based on their relevance to a large scientific journal given a co-authorship network or rank music artists based on their relevance to a music genre given an artist co-performance network. The analyst does not necessarily know which authors have published their research in the examined journal or which artists belong to the examined genre, but after a brief manual inspection of the respective network, a quick query in an online search engine and/or personal experience they manage to locate five network nodes (i.e. authors or artists) that have the required property. Although these five nodes may not necessarily be the most prominent ones in the examined community (i.e. journal or music genre), they are then used as seed nodes that community ranking algorithms can use to help quantify how much each author is related to that journal or how much an artist is related to a genre.

In this work, we focus on problems like these, where the seed nodes form a very small (e.g. 0.1%) random subset of community nodes, for which the ranking algorithm may be unable to discover high quality community ranks (Avrachenkov, Kadavankandy, & Litvak, 2018). In particular, the goal of our research is to improve the quality of ranks calculated by community ranking algorithms using these small sets of -not necessarily high quality- seed nodes. It must be pointed out that our approach does not require further interaction with the process that discovers seed nodes and is hence particularly useful for avoiding additional data gathering.

1.1. Background

To discover communities in networks, researchers usually try to optimize certain structural characteristics that combine high internal connectivity and low number of outgoing edges. For example, two frequently used unsupervised measures that index well-connectedness of communities are conductance (Chalupa, 2017), which compares the number of links from a community to its complement within a network are linked with the number of links within the community, and modularity (Newman, 2006), which compares the number of links from a community to its complement within a network with the number of statistically expected links.

Unfortunately, the community detection problem cannot be universally solved, since there are multiple ways to organize the same network into different communities (Peel, Larremore, & Clauset, 2017). Therefore, if some information about the ground truth is known, it should be used to guide the process of forming communities instead of evaluating the efficacy of detection algorithms (Peel et al., 2017). To this end, researchers have tried to discover local communities around sets of seed nodes that work as ground truth (Ma et al., 2013; Papadopoulos, Skusa, Vakali, Kompatsiaris, & Wagner, 2009; Reichardt & Bornholdt, 2006; Wu & Huberman, 2004; Wu, Jin, Li, & Zhang, 2015; Wu, Huang, Hao, & Chen, 2012; Zhang & Wu, 2012). These communities optimize structural characteristics, such as the aforementioned conductance and modularity. A recurring idea across approaches is to discover which core nodes lead to the seed node community memberships (Wu & Huberman, 2004; Zhang & Wu, 2012).

Two very successful methods that rank network nodes based on their relation to a set of seed nodes are *personalized PageRank* (Andersen, Chung, & Lang, 2006; Lofgren, Banerjee, & Goel, 2016; Whang, Gleich, & Dhillon, 2013; 2016; Yang & Leskovec, 2015; Yin, Benson, Leskovec, & Gleich, 2017) and *heat kernels* (Chung, 2007; 2009; Kloster & Gleich, 2014). Personalized PageRank expands well-selected seed nodes using a random walk with restart strategy; for a normalization W of the adjacency matrix that reflects traversal probabilities, a binary seed vector s indicating seed nodes and a probability $1 - a$ of randomly restarting from one of these seed nodes, personalized PageRank aims to produce ranks r that are proportional to the static probabilities of visiting each node. These ranks adhere to the following equation:

$$r = aWr + (1 - a)s \tag{1}$$

Since W is often sparse, a simple algorithm to compute ranks r is by iterating the above equation until convergence, similarly to the power method for finding eigenvectors. Previous analysis (Kloumann, Ugander, & Kleinberg, 2017) has shown that personalized PageRank can be approximated by the stochastic block model, where graphs are considered to comprise partitions of disjoint sets of node blocks with fixed probability for links between those sets. Therefore, as long as the seed vector s is well-selected (e.g. contains central nodes of the community), thresholding strategies on the calculated ranks r can discover well-separated communities of low conductance.

Heat kernels are similar to personalized PageRank but also penalize longer random walks. Since this practice makes random walks focus more on the area around the seed nodes (Kloster & Gleich, 2014), it may not be able to rank the relevance of all nodes in networks of larger diameters. Since our goal is ranking the whole graph and not performing community detection, we hereby focus on applications of personalized PageRank.

As mentioned before, in this work we consider a less explored community ranking setting, where seed nodes form a very small random subset of community nodes. In this scenario, seed nodes are not necessarily central members of each community. This could reduce the quality of the seed vector and consequently of the ranks produced by personalized PageRank. Additionally, methods that utilize node attributes to improve graph representation clustering (Hamilton, Ying, & Leskovec, 2017; Tian, Gao, Cui, Chen, & Liu,

2014) cannot explore community-defining attributes, since, to do so, they would need to treat them as soft and not hard constraints, i.e. they would be allowed to slightly violate known community memberships.

To improve the outcome of community ranking algorithms for very sparse seed vectors, previous approaches (Gleich & Seshadhri, 2012; Whang, Gleich, & Dhillon, 2016) have found success by performing *neighborhood inflation* on the seed vectors. This process consists of introducing new seed nodes from the neighborhoods of the original seed nodes. Although these approaches procure seed nodes in an unsupervised manner that guarantees a high density of their neighborhoods, the same inflation strategy can also be used when seed nodes form small random subsets of large communities, since the likelihood of selecting boundary nodes as seeds becomes negligible. Unfortunately, sometimes seed vectors are too sparse and this strategy fails to discover adequately many of them to help calculate high quality ranks. But it is also not possible to iterate this scheme because it exponentially grows the number of seed nodes, depending on the network's average node degree.

1.2. Our approach

To solve this conflicting situation, we introduce a new method, which we call *seed oversampling*. This method also procures more seed nodes before running the node ranking algorithm and is similar to a previously proposed semi-supervised network learning practice (Lin & Cohen, 2010), where an expert discovers the most relevant seed nodes to be assigned as seed vectors. In our case however, the expert is replaced by an unsupervised automated process that utilizes ranks calculated by the ranking algorithm on the initial set of seed nodes to find more important community nodes.

The advantage of seed oversampling compared to neighborhood inflation is two-fold. Firstly, it can be used to set up an iterative boosting scheme that can improve the quality of community ranks calculated by ranking algorithms when the original seed vector is too sparse or of low quality. To do so though, the ranking algorithm must meet certain criteria pertaining to a symmetric structure. Secondly, seed oversampling does not depend on the structural centrality of seed nodes but instead relies on the ranking algorithm's definition of which nodes it considers as more central or relevant.

In this work we assess the effectiveness of boosted and simple seed oversampling on two different personalized PageRank algorithms, which employ different normalizations of the adjacency matrix. For both of these algorithms, we aim to improve their predicted node rank quality when ranking communities of real-world networks using only a few random community members as seed nodes. In our experiments, we find that boosted and simple seed oversampling can be used to procure higher node rank quality compared to neighborhood inflation, which already greatly improves the rank quality of the base ranking algorithms.

2. Boosted seed oversampling

In this section we devise a novel strategy that aims to improve the quality of ranks calculated by community ranking algorithms, such as implementations of personalized PageRank, using seed vectors featuring few nodes. Before starting our analysis, we observe that finding community nodes through available seed vectors is similar to the one-class classification problem (Leng, Qi, Miao, Zhu, & Su, 2015; Manevitz & Yousef, 2001), where only members of one class are known. The well-known class imbalance problem (Longadge & Dongre, 2013; Wang, Minku, & Yao, 2015; Weiss, 2004), where a class is underrepresented in available data, is also closely related to one-class classification (Pan et al., 2008). Therefore, the task of finding community nodes can be approached from a class-imbalance perspective. To avoid affecting network structure (e.g. through removing nodes or inserting new ones), a possible approach is to lessen the imbalance arising from too few seed nodes by assigning more network nodes as seeds. To do so, a common assumption is that one-class boundaries must be tight around the original seed nodes. In our network node ranking setting, this means that new seed nodes should necessarily be members of the community as long as the original seed nodes are members of the community.

Motivated by the above observations, we propose a seed oversampling method that discovers more seed nodes that are robust with respect to rerunning the ranking algorithm for local community detection. In particular, if the original seed nodes are members of the community, then new seed nodes should be members of the community too and this property should be preserved after recalculating ranks by enriching the set of seed nodes with the newly found ones. If these conditions are satisfied, the new seed nodes are expected to be at least equally good community candidates as the original seed nodes and could intuitively be considered sources from which the original seed nodes' community memberships stem from. We explicitly analyze this property from the standpoint of personalized PageRank, where the relation between ranks and random walks suggests that more random walks visit new seed nodes compared to the original seed nodes.

Unfortunately, the seed oversampling methodology outlined above sometimes fails to produce adequately many new community nodes in the first iteration, resulting to poor rank quality for extremely sparse seed vectors. At the same time, blindly repeating it many times to find more nodes tends to eventually assign a large portion of network nodes as seeds. To circumvent this problem, we also propose a boosting scheme that combines several iterations to further improve rank quality for certain types of ranking algorithms.

2.1. Seed oversampling

In this subsection we propose a seed oversampling methodology that can be used to proliferate seed nodes with the goal of improving a ranking algorithm $\mathcal{R}(s)$ that ranks network nodes based on the network's structure while being biased towards a binary seed vector s. To detect a community C using this algorithm, an overlapping community detection process requires a seed vector

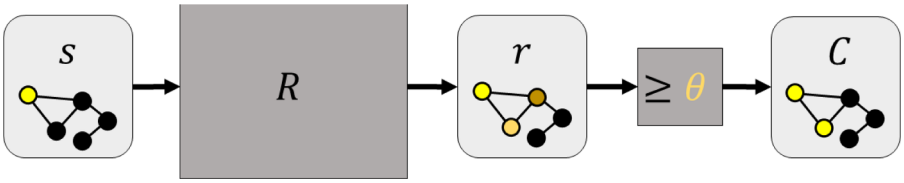


Fig. 1. Utilizing ranking algorithm $\mathcal{R}$ to estimate community C for seed vector s. If a high enough threshold θ is selected, the estimated community can be used non-iteratively as a new vector to produce more accurate ranks.

s: $s[u] = 1 \Rightarrow u \in C$, where u are either automatically generated or queried seed nodes. In other words, seed vector elements that are known members of the community have value of 1. If their membership is unknown, they have value of 0.

To perform community detection based on the ranks $r = \mathcal{R}(s)$ derived from the seed vector, the community detection process would discover an appropriate threshold θ that determines community membership, i.e. $r[u] \geq \theta \Leftrightarrow u \in C$, as demonstrated in Fig. 1.

In this community detection setting, any decision threshold θ that assigns all seed nodes as part of the community can be at most equal to the lesser rank among seed nodes. This means that seed nodes provide an upper bound for the community detection threshold $\min_{u:s[u]=1} r[u] \geq \theta$, since nodes with ranks equal to or greater than at least one seed rank u: $r[u] \geq \min_{u:s[u]=1} r[u] \geq \theta$ are identified as part of the community arising from the seed vector s for any decision threshold θ that also categorizes all seed nodes as part of that community. For example, in the network of Fig. 2a, using the least of seed node ranks as the community detection threshold helps identify more important nodes of the left community.

Another way to look at this property is that, since the ranking algorithm assigns higher ranks to nodes more important to the community's structure, the previously identified seed nodes are at least as important as seed nodes within that structure. Intuitively, a new seed vector that includes these nodes would instruct the ranking algorithm to place even more emphasis on the newly discovered more important community nodes. Therefore, the new seed vector can be constructed through the following procedure:

$$s_1\text{: } s_1[u] = \{1 \text{ if } r[u] \geq \min_{u:s[u]=1} r[u],\ 0 \text{ otherwise}\} \tag{2}$$

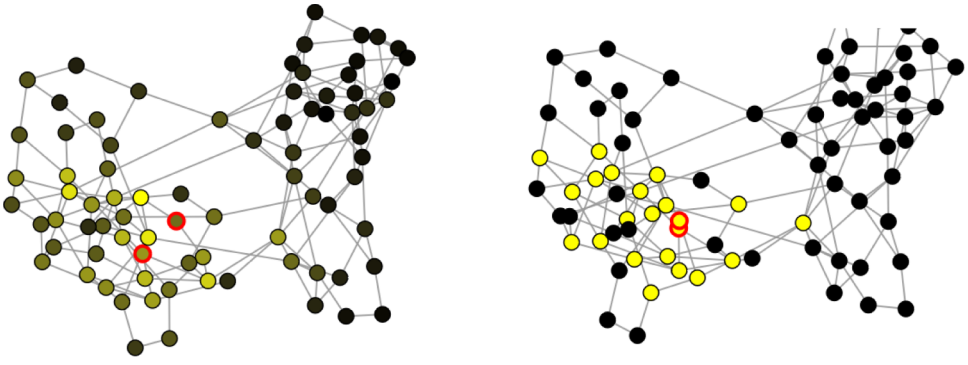
Since this seed vector presumably contains more important seed nodes than the original ones, it can be provided again as input to the ranking algorithm to discover higher quality ranks $r_1 = \mathcal{R}(s_1)$, as demonstrated in Fig. 3. Theorem 1is a formal statement of the above scheme. In particular, it shows that the process of adding not too many new seed nodes that are more important than the least important of the previous ones approximately retains their membership to the community for any valid new threshold. In practice, this approximation is very tight, since successful ranking algorithms often exhibit a strong linear behavior ($\epsilon \to 0$).

Theorem 1. *Let $\mathcal{R}(s)$ be a non-decreasing differentiable ranking algorithm with $\mathcal{R}(0) = 0$ whose linear approximation on s yields absolute rank error at most ϵ. Then:*

$$r[i] \geq \min_{u:s[u]=1} r[u] \Rightarrow r_1[i] \geq \min_{u:s[u]=1} \left(r_1[u] - 2\epsilon \frac{r_1[u] + \epsilon}{r[u] - \epsilon} \right) \tag{3}$$

where $r = \mathcal{R}(s)$, $r_1 = \mathcal{R}(s_1)$ and s_1 is procured by (2).

Proof. Let us use the Jacobian matrix $\mathbb{J}_{\mathcal{R}}(s)$ to produce a first-order approximation of $\mathcal{R}(s_1)$ around s with error vector $\epsilon(s_1)$ such that $|\epsilon(s_1)[u]| \leq \epsilon$ for any node u. Subtracting the element j from element i of this approximation yields:



(a) Algorithm ranks. (b) New seed nodes.

Fig. 2. Node ranks of personalized PageRank with row-based adjacency matrix normalization (see Section 3.2 for details) on a network constructed through a block model that sparsely links the left and right community. Lighter yellow fills indicate higher ranks and red strong outlines indicate the original seed nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

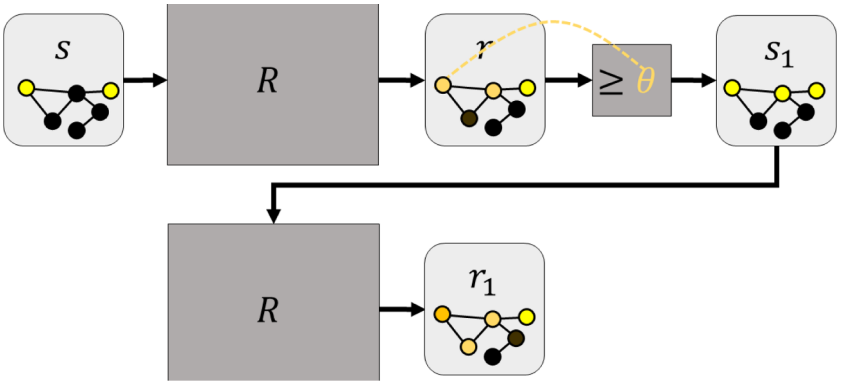


Fig. 3. Utilizing ranking algorithm $\mathcal{R}$ with seed oversampling of original seed nodes s to better estimate ranks r_1 using oversampled seed nodes s_1.

$$\mathcal{R}(s_1) = \mathcal{R}(s) + \mathbb{J}_{\mathcal{R}}(s)(s_1 - s) + \epsilon(s_1)$$
$$\Rightarrow \mathcal{R}(s_1)[i] - \mathcal{R}(s_1)[j] \geq \mathcal{R}(s)[i] - \mathcal{R}(s)[j] + (\mathbb{J}_{\mathcal{R}}(s)[i] - \mathbb{J}_{\mathcal{R}}(s)[j])(s_1 - s) - 2\epsilon$$

where $[\cdot]$ indicates the rows of tables and elements of vectors. For $j = \arg\min_{u:s[u]=1} r[u]$ and i: $r[i] \geq r[j]$ we hence obtain:

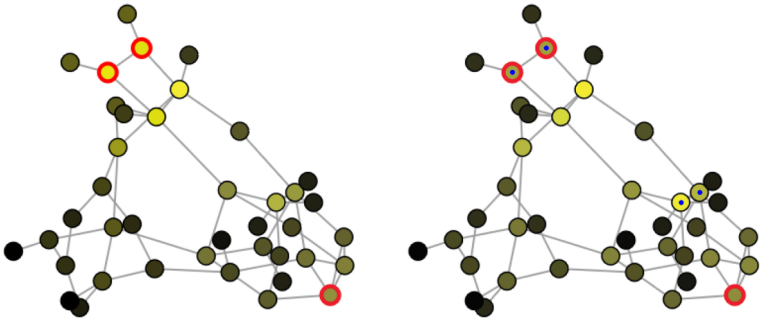
$$0 \leq r[i] - r[j] \leq \sum_{u:s[u]=1} \left(\frac{\partial \mathcal{R}(s)[i]}{\partial s[u]} - \frac{\partial \mathcal{R}(s)[j]}{\partial s[u]} \right) + 2\epsilon \Rightarrow \left(\frac{\partial \mathcal{R}(s)[i]}{\partial \mathcal{R}(s)[j]} - 1 \right) \sum_{u:s[u]=1} \frac{\partial \mathcal{R}(s)[j]}{\partial s[u]} \geq -2\epsilon$$
$$\Rightarrow r_1[i] - r_1[j] \geq r[i] - r[j] - 2\epsilon - 2\epsilon \frac{\sum_{u:s_1[u]=1, s[u]=0} \frac{\partial \mathcal{R}(s)[j]}{\partial s[u]}}{\sum_{u:s[u]=1} \frac{\partial \mathcal{R}(s)[j]}{\partial s[u]}} \Rightarrow r_1[i] - r_1[j] \geq -2\epsilon \frac{\partial r_1[j] + \epsilon}{\partial r[j] - \epsilon}$$

□

2.2. Seed oversampling for personalized PageRank

In this subsection we examine the behavior of personalized PageRank algorithms that implements (an approximation of) Eq. (1) and explain why their linear structure arising from their random walk with restart formulation favors the usage of the proposed seed oversampling scheme.

Before analyzing such algorithms, it must be noted that seed nodes may not be very central ones and instead lie near the perimeter of the respective community. For example let us consider the network in Fig. 4a, where the red strong outlines indicate seed nodes, one of which is a considerable distance apart from the others. For this network a personalized PageRank scheme places higher relevance to the nodes surrounding the seed nodes, including the two upper-left leaf nodes. However, under the knowledge of the third seed node being part of the community, these two nodes are not necessarily central enough to play a significant role in the structure of the ranked community. Therefore, a node ranking scheme should also be able to discover more central nodes from which the community membership of seed nodes arises. This happens in Fig. 4b, where the seed oversampling scheme has placed more



(a) Algorithm ranks. (b) Ranks after seed oversampling.

Fig. 4. Node ranks of personalized PageRank with symmetric adjacency matrix normalization (see Section 3.2 for details) on a network constructed through a block model that sparsely links the upper-left, down-left and down-right communities. Lighter yellow fills indicate higher ranks red strong outlines indicate the original seed nodes. Seed nodes span two of the block communities, thus identifying a super-community. To help spot which node ranks changed the most, they are marked with a dot. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

importance in a more central community node and correctly treats the two close seed nodes and their respective leaf nodes with less importance, since they can be considered to lie in the perimeter of the respective community. This scheme also maintains the otherwise intuitively correct ranks for the rest of the network nodes.

For personalized PageRank to accurately calculate ranks for all graph nodes when the number of seed nodes is much smaller than the size of the community $||s||^2 \ll |C|$, random walks need be comparable in length to the diameter of this community so that they frequently traverse all of its nodes. If random walks infrequently visit a lot of community nodes, the latter would be assigned a much lower score. However, frequently visiting enough nodes requires too low a restart probability, effectively eliminating the impact of the seed vector in the final ranks (see Appendix A). A viable alternative, that is also the main motivation behind neighborhood inflation strategies (Gleich & Seshadhri, 2012), is to add more central nodes in the seed vector.

Moreover, for personalized PageRank algorithms where the adjacency matrix is symmetrically normalized, $r[u]$ is an index of the probability that a random walk which randomly restarts from prior seeds ends on node u. Since random walks between a prior seed and u are bidirectional, $r[u]$ is a lower bound of the probability that a random walk from u would end up to a prior seed. Hence, in such algorithms, ranks higher than $\min_{u:s[u]=1} r_1[u]$ indicate more information flow between u and the seed nodes compared to the lowest ranked seed node with the rest of the seed nodes.

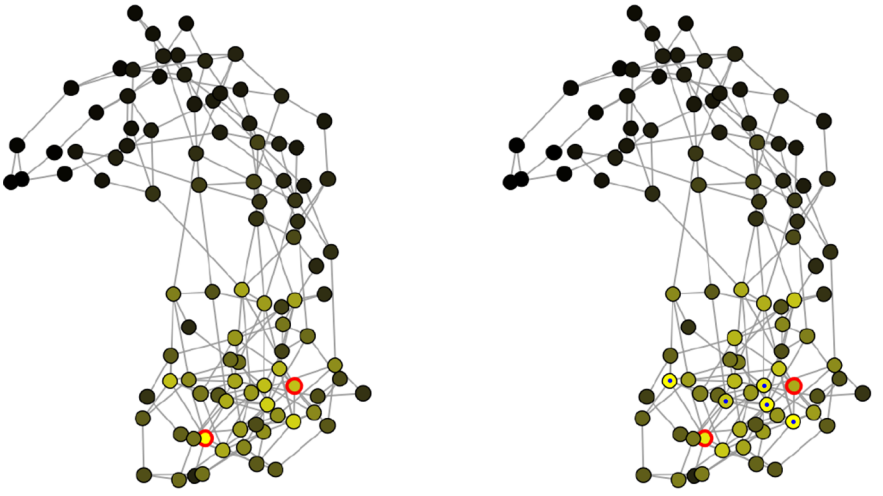
Applying Theorem 1 for personalized PageRank, we can see that the latter theoretically yields perfect adherence to the desired property, since its solution $r = (1-a)(I-aW)^{-1}s$ is fully linear. In practice, its calculated ranks exhibit a very small error ϵ determined by the halting conditions of the algorithm calculating it. It must be noted that, when designing personalized PageRank implementations, this error should be selected to be very small, as we have observed a tendency of large real-world networks to exponentially increase the ratio of new seed nodes to new ones for larger numbers of original seed nodes and $\frac{r_1[j]}{r[j]}$ grows proportionately to this ratio.

Finally, we stress that seed oversampling does not explicitly account for community-defining structural characteristics and that detecting relevant nodes could hence take priority over detecting community nodes. The selection of the node ranking algorithm usually does favor certain structural characteristics that define communities, but it is possible that some relevant nodes could lie outside the community and some irrelevant nodes inside the community (e.g. they may lie close to network leaves). In both cases, it is important for nodes closer to the center of the community to be assigned higher ranks. For example, in Fig. 5 seed oversampling helps assign higher ranks to some of the more central nodes (i.e. well-connected with other high-rank nodes) of the community identified through the two original seed nodes, even if the identified community is the same.

2.3. Boosting scheme for seed oversampling

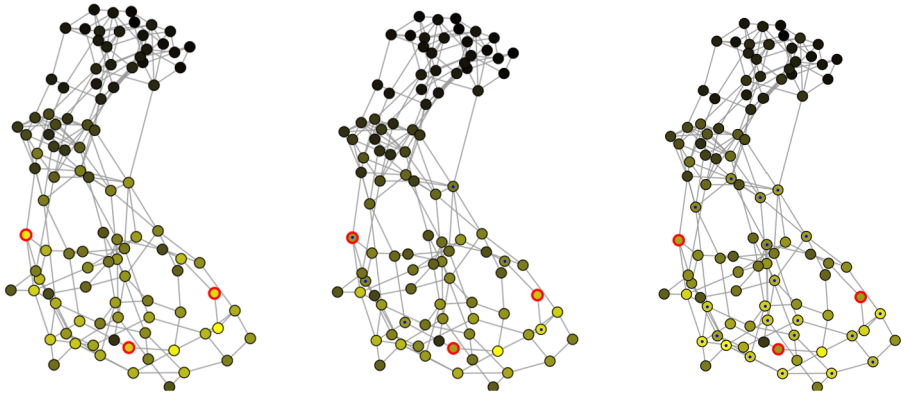
Although our previous analysis shows the success of the seed oversampling scheme in preserving community memberships for new decision thresholds, maintaining this property is not necessarily desirable once we end up with adequately many central seed nodes. For example, repeatedly oversampling seed nodes tends to continuously expand the community, as rankings of community border nodes tend to increase while ranks of the original seed nodes tend to decrease. This often has the adverse effect of over-extending the community seeds to a large portion of the network.

On the other hand, though s_1 may encompass more nodes pertaining to the community compared to s, a single oversampling step



(a) Algorithm ranks. (b) Ranks after seed oversampling.

Fig. 5. Node ranks of personalized PageRank with row-based adjacency matrix normalization (see Section 3.2 for details) on a network constructed through a block model that sparsely links the top and bottom communities. Lighter yellow fills indicate higher ranks and red strong outlines indicate the original seed nodes. To help spot which node ranks changed the most, they are marked with a dot. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Algorithm ranks. (b) After seed oversampling. (c) After boosting.

Fig. 6. Node ranks of personalized PageRank with symmetric adjacency matrix normalization (see Section 3.2 for details) on a network constructed through a block model that sparsely links the top, middle and bottom communities. Lighter yellow fills indicate higher ranks and red strong outlines indicate the original seed nodes. To help spot which node ranks changed the most, they are marked with a dot. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

may fail to reach the best community ranks that can be inferred through the original seed nodes. For example, even when 5 out of 5,000 class seed nodes are oversampled to 81 seed nodes (which happens for a community in our experiments in Section 3), it is possible that more could be required for better community detection. This is demonstrated in the network of Fig. 6a, where the bottom community identified by the three seed nodes with red strong outlines originally places high ranks on their neighbors. In Fig. 6 seed oversampling causes ranks to be spread more evenly across the community, but nodes that are far from all seeds, such as the bottom-left ones, are still not assigned very high ranks despite being well-connected with the rest of the community. To solve this problem, the boosting scheme (7) we develop below repeatedly oversamples the seed vectors of previous iterations; its outcome in Fig. 6c yields ranks that better conform to the intuition that the most bottom nodes are more important for the bottom community's structure. It also discovers which nodes from the middle community are better related to the bottom one.

From the above we can see that there is a need for an iterative seed oversampling scheme that is robust against the increasingly lower confidence of the new seed vector correctness, in that it overstates less the importance of new seed nodes as iterations increase. To devise such a scheme, we observe that it would procure increasingly weaker community ranks over seed oversampling iterations. Hence, we choose to combine these ranks using boosting, which has already been successful for handling class imbalance in other machine learning domains (Błaszczyński & Stefanowski, 2015; Galar, Fernandez, Barrenechea, Bustince, & Herrera, 2012; Krawczyk, Woźniak, & Schaefer, 2014; Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010). In particular, we attempt to set up an iterative boosting-like scheme $\mathcal{B}_N$ that combines ranks calculated by a community ranking algorithm $\mathcal{R}$ for different seed vectors s_i procured through seed oversampling:

$$\mathcal{B}_N(s) = \sum_{n=0}^{N} w_n \mathcal{R}(s_n) \tag{4}$$

where w_n are the weights of the respective ranks, $s_0 = s$ and binary seed vectors s_n: $n \geq 1$ are obtained through seed oversampling of either s or s_{n-1} using the previously calculated ranks $\mathcal{B}_{n-1}(s)$ (for a discussion on selecting these ranks to perform oversampling see Appendix C). This process is more clearly demonstrated in Fig. 7.

Theorem 2 theoretically justifies why we expect a boosting scheme that oversamples s_{N-1} to overestimate less the ranks of the seed nodes in later iterations (which are of lower quality) compared to an iterative scheme. Empirically, this theorem states that, if there exists a good enough (i.e. $\epsilon \to 0$) symmetric linearization of the ranking algorithm and the binary seed vectors remain sparse (i.e. $\|s_N\|^2 \ll U \Rightarrow \delta \to 0$), then the boosting scheme tends to increase less the ranks of the seed nodes. Intuitively, this happens because the boosting scheme preservers a portion of the higher-quality ranks of previous iterations.[1]

Theorem 2. *Let $\mathcal{R}(s)$ be a differentiable non-decreasing ranking algorithm. If it has a symmetric semi-positive definite Jacobian on $s = s_N$ of maximum eigenvalue $\lambda_{\max}$ that approximates it with absolute error at most ϵ across $R(s_n)[u]$, $n \leq N$, $u \in$ nodes, then:*

$$\frac{1 - s_N}{\|1 - s_N\|^2} \cdot \left(\frac{\mathcal{B}_N(s)}{\sum_{n=0}^{N} w_n} - \mathcal{R}(s_N) \right) - \frac{s_N}{\|s_N\|^2} \cdot \left(\frac{\mathcal{B}_N(s)}{\sum_{n=0}^{N} w_n} - \mathcal{R}(s_N) \right) \geq -3\epsilon - \delta \tag{5}$$

where $\cdot$ is the dot product, $\delta = \frac{\lambda_{\max} \| s_N \| \sqrt{U} + \epsilon \| s_N \|^2}{U - \| s_N \|^2}$ and U is the number of network nodes.

[1] Theorem 2does not cover the cases of oversampling s in every step or of discovering negative weights, but those cases exhibit the same behavior when the seed vector s_N is approximately a superset of a previous iteration and boosting weights sum to a non-negative value.

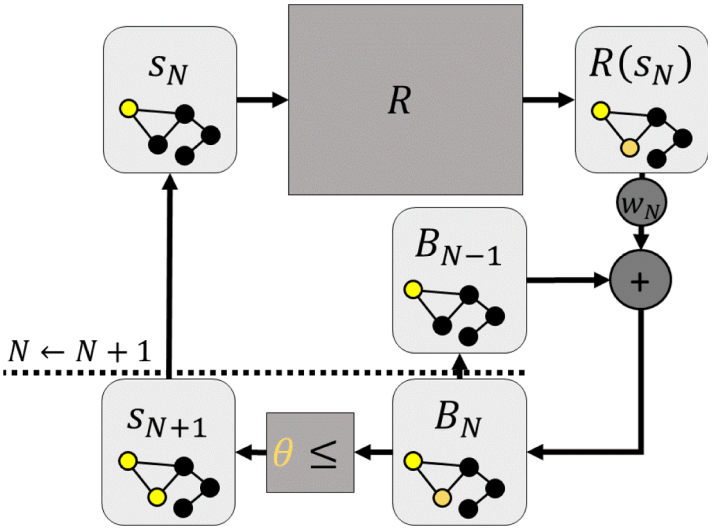


Fig. 7. Single iteration of the boosting scheme for seed oversampling of s_N using the ranks $\mathcal{B}_N(s)$. θ_N is the selected threshold parameter for creating the new seed vector based on s.

Proof. Let $\mathbb{J}_N$ be the Jacobian of $\mathcal{R}(s_N)$. The Jacobian provides the following linear approximation for $x = s_0, \cdots, s_N$ with error $\epsilon(x)$ such that $|\epsilon(x)| \leq \epsilon$:

$$\begin{aligned}\mathcal{R}(x) &= \mathcal{R}(s_N) + \mathbb{J}_N(x - s_N) + \epsilon(x)\\ &\Rightarrow \frac{1 - s_N}{||1-s_N||^2}\cdot\left(\frac{\mathcal{B}_N(s)}{\sum_{n=0}^N w_n} - \mathcal{R}(s_N)\right) - \frac{s_N}{||s_N||^2}\cdot\left(\frac{\mathcal{B}_N(s)}{\sum_{n=0}^N w_n} - \mathcal{R}(s_N)\right)\\ &\geq \left(\frac{1 - s_N}{||1-s_N||^2} - \frac{s_N}{||s_N||^2}\right)\cdot\frac{\sum_{n=0}^N w_n \mathbb{J}_N(s_n - s_N)}{\sum_{n=0}^N w_n} - 2\epsilon\end{aligned}$$

Since the ranking algorithm is non-decreasing and $s_N[u] \geq s_n[u]$ for all nodes u, then $s_n\cdot(R(s_N) - R(s_n)) \geq 0 \Rightarrow s_n\cdot\mathbb{J}_N(s_N - s_n) \geq -\epsilon\,||s_n||^2$. Furthermore, since the Jacobian $\mathbb{J}_{\mathcal{R}}(s_N)$ is symmetric semi-positive definite, if $0 \leq \lambda_{\min}, \lambda_{\max}$ are its minimum and maximum eigenvalues respectively then:

$$\begin{aligned}&\left(\frac{1 - s_N}{||1-s_N||^2} - \frac{s_N}{||s_N||^2}\right)\cdot\mathbb{J}_N(s_n - s_N)\\ &= (s_N - s_n)\cdot\mathbb{J}_N(s_N - s_n)\left(\frac{1}{||s_N||^2} + \frac{1}{||1-s_N||^2}\right)\\ &\quad + s_n\cdot\mathbb{J}_N(s_N - s_n)\left(\frac{1}{||s_N||^2} + \frac{1}{||1-s_N||^2}\right) - \frac{1}{||1-s_N||^2}\cdot\mathbb{J}_N(s_N - s_n)\\ &\geq ||s_N - s_n||^2\lambda_{\min}\left(\frac{1}{||s_N||^2} + \frac{1}{||1-s_N||^2}\right) - \epsilon\left(1 + \frac{||s_N||^2}{||1-s_N||^2}\right) - \lambda_{\max}\sqrt{U}\frac{||s_N - s_n||}{||1-s_N||^2}\end{aligned}$$

Since s_N is binary yielding $||1-s_N||^2 = U - ||s_N||^2$ and $||s_N - s_n|| \leq ||s_N||$ we thus obtain the desired outcome by substituting these inequalities in the previous one. □

A critical aspect that determines the effectiveness of the boosting scheme in real-life networks is the selection of weights w_N that optimize an appropriate boosting goal. However, in our setting, the ground truth that could be used to determine this goal contains too few datapoints (i.e. seed nodes) to split into training and test sets. At the same time, the small number and possible non-central position of seed nodes does not allow the usage of information retrieval measures (Järvelin & Kekäläinen, 2000; Wang et al., 2013) that ranking combination approaches (Freund, Iyer, Schapire, & Singer, 2003; Usunier, Amini, & Patrick, 2004; Valizadegan, Jin, Zhang, & Mao, 2009; Wu, Burges, Svore, & Gao, 2008; 2010) could aim to optimize.

Although rank combination approaches are not directly applicable on our domain, they share the intuition that combined ranks should minimize differences between combined rank vectors. In our boosting setting, this suggests that the boosted ranks should have minimal differences with both the previous boosting iteration and the new rank candidates. Since ranks that imply important nodes lay approximately in the same order of magnitude (much lower ranks are by definition considered unimportant by the ranking algorithm), we choose to employ the Euclidean distance based on the L2 norm $\|\cdot\|$ to combine those differences into a loss function $\mathcal{L}_N$ that needs be minimized for each boosting step N:

$$\mathcal{L}_N = ||\mathcal{B}_N(s) - \mathcal{B}_{N-1}(s)||^2 + ||\mathcal{B}_N(s) - \mathcal{R}(s_N)||^2 \tag{6a}$$

Since this loss function is convex, it can be minimized in each boosting step by selecting weights w_N that satisfy:

$$\frac{\partial \mathcal{L}_N}{\partial w_N} = 0 \Leftrightarrow w_N = \frac{1}{2} - \frac{\mathcal{R}(s_N)\cdot\mathcal{B}_{N-1}(s)}{2||\mathcal{R}(s_N)||^2} \tag{6b}$$

However, it can be argued that the previous boosting iteration and the next rank candidates are not equally valid for contributing to the prediction of high quality boosting ranks for all network nodes. In particular, next rank candidates may progressively yield lower quality ranks for the nodes not included in the seed vector s_N. At the same time next rank candidates also reflect the centrality of seed nodes in the network's structure as the seed vector becomes less sparse. On the other hand, the previous boosting iteration is the best available approximation for non-seed node ranks.

Taking these into account, we refine (6a) so that it partially compares only the boosted ranks of seed nodes with the next rank candidates and only the boosted ranks of non-seed nodes with the previous boosting iteration:

$$\mathcal{L}_N = ||(1 - s_N)\bigstar(\mathcal{B}_N(s) - \mathcal{B}_{N-1}(s))||^2 + ||s_N \bigstar(\mathcal{B}_N(s) - \mathcal{R}(s_N))||^2 \tag{7a}$$

where $\star$ is the element-wise vector multiplication. Similarly to before, this loss function is minimized in each boosting step by selecting weights w_N that satisfy:

$$\frac{\partial \mathcal{L}_N}{\partial w_N} = 0 \Leftrightarrow w_N = \frac{\mathcal{R}(s_N)}{||\mathcal{R}(s_N)||^2}\cdot\left(s_N \bigstar \left(\mathcal{R}(s_N) - \mathcal{B}_{N-1}(s)\right)\right) \tag{7b}$$

Theorem 3. *For a boosting scheme that oversamples s_{N-1} on step N and adheres to either (6) or (7), $\frac{\mathcal{B}_N(s)}{\sum_{n=0}^{N} w_n}$ asymptotically converges.*

Proof. Since s_N is always produced through seed oversampling of s_{N-1}, $\|s_N\|^2$ form an upper-bounded (by the number of nodes in the network) non-decreasing sequence of N. Consequently, the seed vector reaches within a finite number of N_0 steps the equilibrium $s_N = s_{N-1} \Rightarrow \mathcal{R}(s_N) = \mathcal{R}(s_{N-1})\ \forall\ N \geq N_0$. Therefore, $\frac{\mathcal{B}_N(s)}{\sum_{n=0}^{N} w_n}$ converges for both (6) and (7), because at worst it would asymptotically be attracted to $\mathcal{R}(s_{N_0})$ as $N \to \infty$. □

3. Experiments

3.1. Networks

Our experiments aim to evaluate the merit of the seed oversampling and boosting methodologies proposed in Section 2 towards improving community ranking strategies. To do so, we experiment on several well-known annotated networks that are often used to validate clustering methodologies (Yang & Leskovec, 2015); the *Amazon* co-purchasing network (Leskovec, Adamic, & Huberman, 2007), the *DBLP* author co-authorship network (Tang et al., 2008) and the *Flickr* and *YouTube* networks of user links obtained through crawling on the respective social platforms (Mislove, Marcon, Gummadi, Druschel, & Bhattacharjee, 2007).

The Amazon network[2] comprises links between products frequently co-purchased in summer 2006. Those products form communities based on their type (e.g. Book, CD, DVD, Video). We use the 2011 version of the DBLP dataset,[3] which comprises 1,632,442 papers from the DBLP database. From this dataset, we extracted an author network based on co-authorship relations between authors. In this network, we expect authors to form overlapping communities based on academic sources (i.e. journals and/or conferences) they have published in. Finally, the Flickr and YouTube social networks comprise links between users and contain information about user-annotated groups.

To facilitate our experiments with extremely sparse knowledge of community memberships, we experiment only on the largest communities of each dataset. We expect that, when such communities are large enough, they are not in a state of expansion and thus are not lacking too many good potential candidates or missing links. When network communities are smaller, which often happens in interpersonal social networks (Leskovec et al., 2009) they can misguide the assessment of ranking algorithms towards lower effectiveness. The size of the aforementioned networks, as well as our threshold for selecting the largest communities and the subsequent number of communities used in experiments are detailed in Table 1.

3.2. Experiment setup

For each of the above communities, we select a random small subset of nodes as seeds with which to rank the rest of the network's nodes. Experiments are conducted for this subset covering 1%, 0.5% and 0.1% of each community, if those subsets are not empty. In some of those cases seed vectors comprise very few (e.g. $3 - 50$) nodes. We employ two different node ranking algorithms on the random walk with restart scheme demonstrated in (1) for network adjacency matrix M, diagonal node degree matrix D, seed vector s and random walk restart probability $1 - a = 1\%$, which a frequently suggested restart probability for good community detection (Lai,

[2] https://snap.stanford.edu/data/amazon-meta.html .

[3] DBLP-Citation-network V4 from https://aminer.org/citation .

Table 1
Networks and communities used for experiments.

Networks		Nodes	Edges	Communities	Min nodes
Amazon	Leskovec et al. (2007)	554,789	1,788,725	4	≥ 5000
DBLP	Tang et al. (2008)	978,488	11,289,510	52	≥ 5000
Flickr	Mislove et al. (2007)	1,715,255	22,613,981	42	≥ 1000
YouTube	Mislove et al. (2007)	1,138,499	4,945,382	7	≥ 300

Wu, Lu, & Nardini, 2011; Whang et al., 2013) and whose efficacy we also corroborated in Appendix A:

- Personalized PageRank with row-wise adjacency matrix normalization $W = MD^{-1}$ (Pan, Yang, Faloutsos, & Duygulu, 2004). Certain optimizations have been proposed for fast calculation of personalized PageRanks (Bahmani, Chowdhury, & Goel, 2010; Jung, Shin, Sael, & Kang, 2016; Lofgren, Banerjee, Goel, & Seshadhri, 2014; Shin, Jung, Lee, & Kang, 2015). However, since we are not focusing on performance, in this work we find the appropriate solution by iterating (1) starting from the seed vector until converging to mean squared differences of consecutive iteration ranks ≤ 0.00001. This scheme does not adhere to the symmetric property required for applying Theorem 2 and hence it is possible that boosting seed oversampling may worsen the quality of produced ranks.
- Personalized PageRank with symmetric adjacency matrix normalization $W = D^{-1/2}MD^{-1/2}$. In a random walk setting, traversal probabilities are considered to be proportional to weights of the symmetric matrix W (Zhou, Bousquet, Lal, Weston, & Schölkopf, 2004). This normalization often produces good community ranks (Whang et al., 2016), since it boasts several favorable characteristics (Tong, Faloutsos, & Pan, 2006), such as symmetric relationships between nodes. For this algorithm, we employ the same iterative implementation as before. This scheme adheres to the required properties of Theorem 2, since it is linear and its Jacobian $(I - aW)^{-1}(1 - a)$ is symmetric and semi-positive definite (the identity matrix I and the network's Laplacian $I - W$ have those properties, which are then retained through the matrix operations of multiplication with a constant, addition and inversion).

To improve the quality of ranks produced by these two node ranking algorithms, we employ three different schemes based on our proposed oversampling schemes of Section 2:

- **None.** The seed vectors are used as-is. The quality of procured ranks under this scheme can serve as a baseline that helps quantify whether the rest of these schemes introduce quality improvements.
- **Inflate.** A one-time neighborhood inflation previously used to improve seed node quality (Gleich & Seshadhri, 2012; Whang et al., 2016). This practice cannot be improved through a boosting scheme, since doing so covers the whole graph within very few iterations and yields completely random node ranks (with 50% AUC score).
- **OSample.** A one-time application of the seed oversampling scheme outlined in Section 2.1 to enrich the seed vector s before running the ranking algorithm.
- **NBoost.** The naive boosted seed oversampling scheme outlined in Section 2.3, where seed vectors s_N are obtained from seed oversampling s (which we found to yield better results than oversampling s_{N-1}) based on the ranks $\mathcal{B}_{N-1}(s)$ and boosting weights are selected according to (6b). Since this scheme may require too many boosting iterations to fully converge, in our experiments we stop when $|w_N| \leq 0.001$.
- **PBoost.** The partial boosted seed oversampling scheme outlined in Section 2.3, where seed vectors s_N are obtained from seed oversampling s_{N-1} (which we found to yield better results than oversampling s) based on the ranks $\mathcal{B}_{N-1}(s)$ and boosting weights are selected according to (7b). Since this scheme may require too many boosting iterations to fully converge, in our experiments we stop when $|w_N| \leq 0.001$.

To evaluate the ranks calculated by combining the aforementioned node ranking algorithms with schemes that improve rank quality, we use the Area Under Curve (AUC) (Bradley, 1997) of the Receiving Operating Characteristics (ROC) of non-seed nodes. AUC is a robust rank evaluation metric not affected by the imbalance between positive (i.e. community) and negative (i.e. non-community) memberships (Ling, Huang, Zhang et al., 2003). AUC values closer to 100% indicate that community members achieve higher ranks compared to non-community members, whereas 50% AUC indicates random classification (Hanley & McNeil, 1982). In the subsequent experiments, we report the average scheme ranks[4] across all overlapping communities in each dataset, also averaged across 5 iterations.

Finally, we compare schemes with the non-parametric Wilcoxon test for p-value < 0.015 to confidently determine whether their

[4] Average scheme ranks are not related to the ranks of network nodes. In each experiment, the schemes are sorted based on the quality (AUC) of network ranks and each scheme is assigned an integer rank based on their position in this ordering, where 1 denotes the best scheme and ties are resolved by assigning the average position of tied schemes. These new ranks are the ones averaged across experiments to summarize the performance of each scheme. Contrary to network ranks, where larger values indicate higher relation to communities, lower scheme ranks indicate better schemes.

AUC evaluations pair-wise differ across experiments. To assert whether the best schemes differ significantly from the rest, we also follow a well-known procedure for multi-scheme comparison (Demšar, 2006; Derrac, García, Molina, & Herrera, 2011; Garcia & Herrera, 2008); we make sure that differences between schemes are statistically significant according to a Friedman test with p-value < 0.015 and then use the Holm post-hoc test to discover differences with 0.05 level of significance. We also visualize the differences between key schemes by calculating their average scheme ranks across experiments and employing the Nemenyi post-hoc test[5] to determine the critical difference over which average scheme rank differences have 0.05 level of significance.

3.3. Experiment results

In Tables 2 and 3 we present the AUC scores for the two node ranking algorithms, averaged across 5 iterations of experiments conducted for non-empty sets of seed nodes. We can see that all fixing strategies always improve rank quality with statistical significance (the statistical significance of fixing strategies improving rank quality across all experiment setups passes the Wilcoxon test with p-value < 0.015). Furthermore, the Friedman test rejects the hypothesis schemes perform the same and the Holm test yields that Inflate, OSample and PBoost also induce improvements for both personalized PageRank algorithms with 0.05 statistical significance.

Empirically, we can see that these improvements are larger when the seed set is small enough to not yield high rank qualities on its own. Furthermore, approaches manage to produce more meaningful ranks on the networks comprising smaller communities, where the ranking algorithms on their own produce almost completely random ranks. Taking into account previous concerns on the small nature of communities in large social networks (Leskovec et al., 2009) that make community membership a less adequate indicator of good relations with external members to the community, rank quality improvements in those networks can be considered meaningful from a ranking perspective, even if they still do not correspond to high predictive capability.

Another interesting observation is that the NBoost scheme sometimes exhibits higher performance compared to PBoost but also often fails to even preserve the performance of seed oversampling. This outcome reflects the lack of robustness of this boosting scheme, which suggests that PBoost should be preferred when there is no way to priorly assess its effectiveness. This finding also validates our concern that devised boosting loss functions should account for the degradation of rank certainty across iterations, which the objective of NBoost ignores.

The methods arising from our analysis as more suitable for each ranking algorithm are PBoost for personalized PageRank with symmetric normalization and seed oversampling for personalized PageRank with row-wise normalization. These methods outperform neighborhood inflation with statistical significance (the Wilcox tests yield p-value < 0.015). The Holm test corroborates that PBoost is the best scheme for the first algorithm with statistical significance 0.05. For the second one, no scheme can be deemed strictly superior to the rest, since sometimes the boosting schemes manage to yield better rank quality.

These results show the viability of the boosted seed oversampling scheme when the properties of Theorem 2 are satisfied. On the other hand, for the personalized PageRank with row-wise normalization algorithm, which does not satisfy these properties, both boosting schemes often fail to maintain the rank quality obtained from seed oversampling. An intuitive explanation of this phenomenon is that the non-symmetric structure of row-wise normalization introduces a tendency to discover more nodes whose information flow does not necessarily return to previously identified community nodes.

We finally visualize the differences between the Inflate, OSample and PBoost schemes. These comparisons omit the base ranking algorithm, which is outperformed by all of them across experiments, as well as the less robust NBoost scheme. Before comparing these schemes, we again employed the Friedman test, which rejects the null hypothesis that these schemes produce similar AUC values for both ranking algorithms. Fig. 8 visually represents the average scheme ranks of schemes and the critical difference calculated by the Nemenyi test.[6] Our findings agree to a large extend with the previous ones; the PBoost scheme consistently introduces better rank quality compared to the neighborhood inflation strategy for personalized PageRank with symmetric normalization and seed oversampling introduces similar or better rank quality to the neighborhood inflation strategy for personalized PageRank with row-wise normalization. It must be noted that, for both algorithms, seed oversampling is ranked as better than the neighborhood inflation strategy, since it performs equally well or better in all but one experiments.

4. Conclusions and future work

In this paper we examined the problem of ranking network nodes based on their relevance to communities identified through a few seed nodes. To improve the quality of ranks produced by local community ranking algorithms, we proposed a seed oversampling scheme that uses an initial estimate of network node ranks to enrich the set of seed nodes with new more important ones and showed that they remain more important after rerunning the ranking algorithm. We also introduced a boosting methodology that weights the outcomes of iteratively repeating seed oversampling and showed that it tends to overestimate less the ranks of seed nodes in later iterations when the ranking algorithm satisfies certain properties.

In our experiments in real-life networks, we found that seed oversampling can greatly improve the quality of community ranks calculated by two personalized PageRank algorithms and does so better than the previously proposed neighborhood inflation

[5] The Nemenyi test is usually weaker than the Holm test, but the fact that it yields an ordered outcome makes it suitable for visualization.

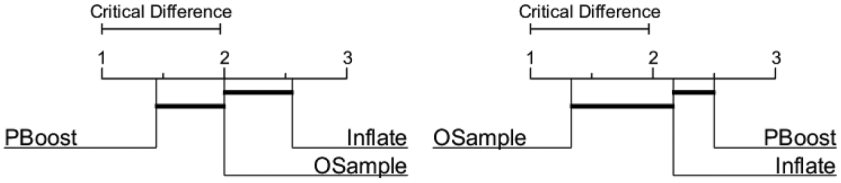
[6] Although the Nemenyi test often tends to underestimate the statistical significance of algorithm differences, in this settings our findings also happen to coincide with the Holm test when employed on these three schemes.

Table 2
AUC scores for different approaches to fixing sparsity of seed vectors used by personalized PageRank with symmetric normalization and parameter $a = 0.99$.

Network	Seeds	None	Inflate	OSample	NBoost	PBoost
Amazon	1%	90%	91%	91%	**92%**	**92%**
Amazon	0.5%	90%	89%	90%	90%	**91%**
Amazon	0.1%	67%	79%	79%	**88%**	86%
DBLP	1%	71%	80%	**81%**	**81%**	80%
DBLP	0.5%	61%	**77%**	74%	70%	**77%**
DBLP	0.1%	51%	68%	66%	64%	**70%**
Flickr	1%	52%	56%	**59%**	**59%**	58%
Flickr	0.5%	51%	56%	57%	**58%**	**58%**
YouTube	1%	50%	52%	**59%**	58%	56%

Table 3
AUC scores for different approaches to fixing sparsity of seed vectors used by personalized PageRank with row-wise normalization and parameter $a = 0.99$.

Network	Seeds	None	Inflate	OSample	NBoost	PBoost
Amazon	1%	83%	90%	**91%**	90%	89%
Amazon	0.5%	83%	**89%**	**89%**	88%	**89%**
Amazon	0.1%	61%	85%	86%	87%	**88%**
DBLP	1%	72%	78%	**79%**	69%	71%
DBLP	0.5%	62%	**77%**	**77%**	72%	71%
DBLP	0.1%	51%	68%	**69%**	67%	68%
Flickr	1%	54%	58%	**59%**	**59%**	57%
Flickr	0.5%	51%	55%	**57%**	**57%**	56%
YouTube	1%	52%	**57%**	**57%**	53%	56%



(a) Symmetric normalization. (b) Row-wise normalization.

Fig. 8. Average scheme ranks (smaller is better) of AUC evaluation across datasets for two different personalized PageRank algorithms, where lower values indicate better schemes. Average rank differences higher than the critical difference discovered through the Nemenyi test are considered statistically significant. Non-significant differences are grouped using bold horizontal lines.

heuristic. The boosting scheme can further improve rank quality for the algorithm on which it is theoretically applicable. Based on these results, we encourage using seed oversampling to improve local community ranking processes. There also exists considerable merit in further improving ranks using our boosting scheme, if the latter's objective is suitable for the ranking task.

In the future, we aim to run experiments on both smaller and larger networks, possibly adjusting our methods to supplement scalable implementations of PageRank for the latter. We are also interested in conducting a more thorough theoretical analysis on the boosting scheme that would explicitly take into account the degradation of seed quality through seed oversampling iterations.

Acknowledgement

This work was partially funded by the European Commission under contract numbers H2020-761634 FuturePulse and H2020-825585 HELIOS.

Appendix A. The effect of random walk length on rank quality

In this appendix, we present a series of experiments that assess the necessity of seed oversampling in comparison to changing (e.g. increasing) the length of random walks. We also examine the effect of different random walk lengths on seed oversampling outcomes.

To do so, we selected both high and low random walk restart probabilities ($1 - a = 15\%$ and $1 - a = 0.5\%$ respectively) and investigate their impact on community rank quality in the networks of Section 3.1. High random walk restart probabilities produce shorter random walks, whereas low random walk restart probabilities produce longer random walks. In Tables 4 and 5 we show the results of these experiments for two personalized PageRank algorithms. Comparing these results to the ones presented in Section 3.3,

Table A1
AUC scores without and with seed oversampling for personalized PageRank with symmetric normalization and different values of the parameter a.

		a = 0.85		a = 0.995	
Network	Seeds	None	OSample	None	OSample
Amazon	1%	85%	85%	90%	91%
Amazon	0.5%	78%	75%	84%	90%
Amazon	0.1%	57%	57%	71%	82%
DBLP	1%	62%	65%	71%	81%
DBLP	0.5%	60%	60%	61%	76%
DBLP	0.1%	59%	51%	51%	65%
Flickr	1%	52%	52%	52%	57%
Flickr	0.5%	51%	51%	51%	57%
Youtube	1%	50%	50%	50%	61%

Table A2
AUC scores without and with seed oversampling for personalized PageRank with row-wise normalization and different values of the parameter a.

		a = 0.85		a = 0.995	
Network	Seeds	None	OSample	None	OSample
Amazon	1%	85%	88%	90%	90%
Amazon	0.5%	78%	57%	83%	89%
Amazon	0.1%	57%	50%	64%	82%
DBLP	1%	68%	66%	72%	80%
DBLP	0.5%	60%	61%	63%	74%
DBLP	0.1%	51%	51%	51%	69%
Flickr	1%	52%	53%	52%	57%
Flickr	0.5%	51%	51%	51%	56%
Youtube	1%	50%	50%	50%	56%

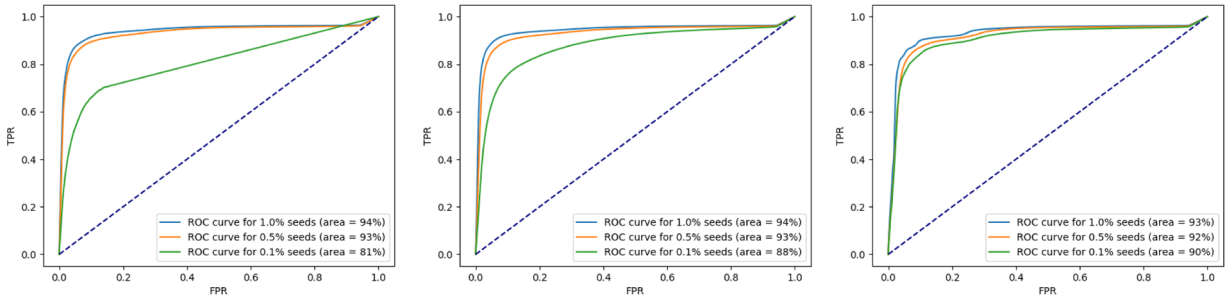
we can see that both too short and too long random walks worsen rank quality in all networks.

Furthermore, short random walks seem not to work well themselves with seed oversampling. We attribute this behavior to the fact that they heavily restrict ranking schemes to very short areas around the seed nodes, which in turn causes seed oversampling to find new seeds predominantly in those areas and thus to often yield lower rank quality. On the other hand, as previously discussed, longer random walks progressively tend to approach an eigenvector of the network's adjacency matrix and thus may relate more loosely to the examined community's structure. Our experiments also confirm that seed oversampling is a superior approach for treating low node rank quality compared to selecting different random walk lengths.

Comparing these results with the ones in Section 3.3, we confirm that the frequently proposed restart probability $1 - a = 1\%$ seems to be the better choice for community detection in large networks and that selecting shorter or longer random walks negatively impacts the quality of seed ranks. In all cases, seed oversampling manages to either retain approximately the original seed node rank quality (which happens for most of the short random walks, i.e. those arising from high restart probabilities) or to significantly improve rank quality.

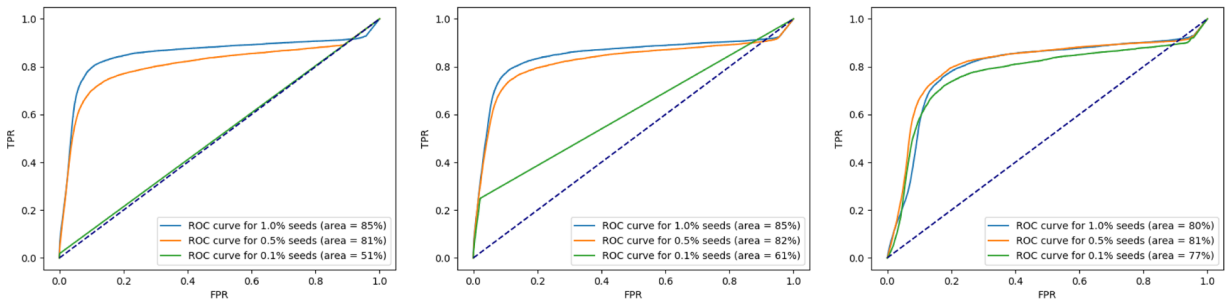
Appendix B. Movement of the ROC curve's inflection point

In Figs. 9 and 10 we demonstrate the ROC curves of two communities for the None, OSample and PBoost schemes of Section 3.2 with different seed vector sparsities. We can observe how the inflection point of the more sparse seed vectors is effectively moving to



(a) None (b) OSample (c) PBoost

Fig. B1. ROC curves of personalized PageRank with symmetric adjacency matrix normalization for the largest community of the Amazon network. The dashed diagonal represents the ROC of random ranks.



(a) None (b) OSample (c) PBoost

Fig. B2. ROC curves of personalized PageRank with symmetric adjacency matrix normalization for the smallest community of the Amazon network. The dashed diagonal represents the ROC of random ranks.

previous positions within the ROC curve of less sparse vectors. This means that, after a certain point of gradually introducing more false positives (i.e. setting more outside nodes as community members), the ranking algorithm abruptly stops discovering more true positives (i.e. community members) at an acceptable rate. This behavior suggests that the ranking algorithm is good at discovering more community members close to the seed nodes but fails to do as random walks branch away.

Appendix C. Boosting scheme selection

For the boosting scheme presented in (4), there are two possible ways to select ranks that can help perform seed oversampling as per (2) on either the original seed vector s or the previous seed vector s_{N-1} to obtain the next seed vector s_N in each boosting step N. The first way is to implement a voting mechanism between iterations of the seed oversampling strategy as demonstrated in Fig. 11. The second one, which is followed throughout this paper, oversamples the previous seed vector using the ranks produced by the previous boosting scheme $\mathcal{R}(s_{N-1})$.

The demerit of the first scheme that led us to adopt the second one comes from a methodological perspective; if we compare the diagrams of Figs. 11 and 7, which demonstrate the first and second boosting schemes respectively, we can see that in the first one there exists no flow from any boosting iteration towards the selection of the seed vectors. However, since this process does not guarantee that s_N is of high quality (in fact, the seed vector tends to eventually cover a number of nodes comparable or even greater than the size of the community), each iteration progressively introduces higher uncertainty in the calculated ranks. This way, uncertainties in earlier iterations are directly propagated to later ones without attempting to mitigate them through comparison with the previous boosting outcome.

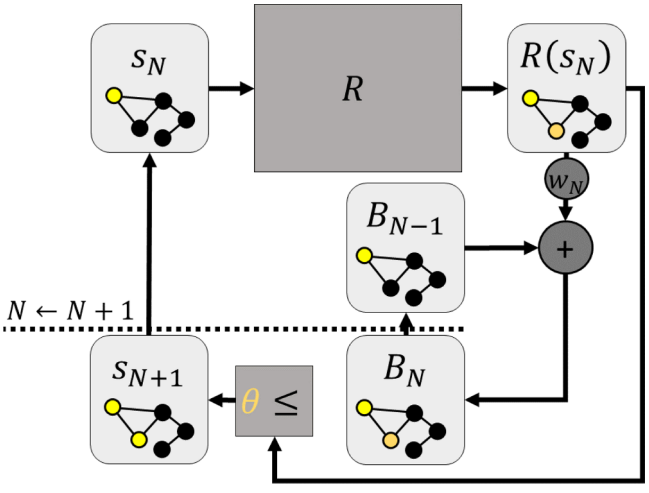


Fig. C1. Single iteration of the boosting scheme for seed oversampling of s_N or s using $\mathcal{R}(s_N)$ instead of the boosting outcome. θ_N is the selected threshold parameter for creating the new seed vector.

References

Andersen, R., Chung, F., & Lang, K. (2006). *Local graph partitioning using PageRank vectors. 2006 47th Annual IEEE symposium on foundations of computer science (FOCS'06)*. IEEE475–486.
Avrachenkov, K., Kadavankandy, A., & Litvak, N. (2018). Mean field analysis of personalized PageRank with implications for local graph clustering. *Journal of statistical physics, 173*(3–4), 895–916.
Bahmani, B., Chowdhury, A., & Goel, A. (2010). Fast incremental and personalized PageRank. *Proceedings of the VLDB Endowment, 4*(3), 173–184.
Błaszczyński, J., & Stefanowski, J. (2015). Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing, 150*, 529–542.
Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30*(7), 1145–1159.
Chalupa, D. (2017). A memetic algorithm for the minimum conductance graph partitioning problem. arXiv preprint arXiv:1704.02854.
Chung, F. (2007). The heat kernel as the PageRank of a graph. *Proceedings of the National Academy of Sciences, 104*(50), 19735–19740.
Chung, F. (2009). A local graph partitioning algorithm using heat kernel PageRank. *Internet Mathematics, 6*(3), 315–330.
Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research, 7*(Jan), 1–30.
Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation, 1*(1), 3–18.
Fortunato, S. (2010). Community detection in graphs. *Physics Reports, 486*(3–5), 75–174.
Fortunato, S., & Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports, 659*, 1–44.
Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research, 4*(Nov), 933–969.
Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(4), 463–484.
Garcia, S., & Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research, 9*(Dec), 2677–2694.
Gleich, D. F., & Seshadhri, C. (2012). *Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. Proceedings of the 18th ACMSIGKDDinternational conference on knowledge discovery and data mining*. ACM597–605.
Hamilton, W., Ying, Z., & Leskovec, J. (2017). *Inductive representation learning on large graphs. Advances in neural information processing systems*1024–1034.
Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29–36.
Järvelin, K., & Kekäläinen, J. (2000). *IR evaluation methods for retrieving highly relevant documents. Proceedings of the 23rd annual international ACMSIGIR conference on research and development in information retrieval*. ACM41–48.
Jung, J., Shin, K., Sael, L., & Kang, U. (2016). Random walk with restart on large graphs using block elimination. *ACM Transactions on Database Systems (TODS), 41*(2), 12.
Kloster, K., & Gleich, D. F. (2014). *Heat kernel based community detection. Proceedings of the 20th ACMSIGKDDinternational conference on knowledge discovery and data mining*. ACM1386–1395.
Kloumann, I. M., Ugander, J., & Kleinberg, J. (2017). Block models and personalized PageRank. *Proceedings of the National Academy of Sciences, 114*(1), 33–38.
Krawczyk, B., Woźniak, M., & Schaefer, G. (2014). Cost-sensitive decision tree ensembles for effective imbalanced classification. *Applied Soft Computing, 14*, 554–562.
Lai, D., Wu, X., Lu, H., & Nardini, C. (2011). Learning overlapping communities in complex networks via non-negative matrix factorization. *International Journal of Modern Physics C, 22*(10), 1173–1190.
Leng, Q., Qi, H., Miao, J., Zhu, W., & Su, G. (2015). One-class classification with extreme learning machine. *Mathematical Problems in Engineering, 2015*.
Leskovec, J., Adamic, L. A., & Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Transactions on the Web (TWEB), 1*(1), 5.
Leskovec, J., Lang, K. J., Dasgupta, A., & Mahoney, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics, 6*(1), 29–123.
Leskovec, J., Lang, K. J., & Mahoney, M. (2010). *Empirical comparison of algorithms for network community detection. Proceedings of the 19th international conference on world wide web*. ACM631–640.
Lin, F., & Cohen, W. W. (2010). *Semi-supervised classification of network data using very few labels. Advances in social networks analysis and mining (ASONAM), 2010 international conference on*. IEEE192–199.
Ling, C. X., Huang, J., Zhang, H., et al. (2003). *AUC: A statistically consistent and more discriminating measure than accuracy. IJCAIVol. 3. IJCAI* 519–524.
Lofgren, P., Banerjee, S., & Goel, A. (2016). *Personalized PageRank estimation and search: A bidirectional approach. Proceedings of the ninth ACM international conference on web search and data mining*. ACM163–172.
Lofgren, P. A., Banerjee, S., Goel, A., & Seshadhri, C. (2014). *Fast-PPR: Scaling personalized PageRank estimation for large graphs. Proceedings of the 20th ACMSIGKDDinternational conference on knowledge discovery and data mining*. ACM1436–1445.
Longadge, R., & Dongre, S. (2013). Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707.
Ma, L., Huang, H., He, Q., Chiew, K., Wu, J., & Che, Y. (2013). *Gmac: A seed-insensitive approach to local community detection. International conference on data warehousing and knowledge discovery*. Springer297–308.
Manevitz, L. M., & Yousef, M. (2001). One-class SVMs for document classification. *Journal of Machine Learning Research, 2*(Dec), 139–154.
Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., & Bhattacharjee, B. (2007). *Measurement and analysis of online social networks. Proceedings of the 7th ACMSIGCOMM conference on internet measurement*. ACM29–42.
Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences, 103*(23), 8577–8582.
Palla, G., Derényi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature, 435*(7043), 814.
Pan, J.-Y., Yang, H.-J., Faloutsos, C., & Duygulu, P. (2004). *Automatic multimedia cross-modal correlation discovery. Proceedings of the tenth ACMSIGKDDinternational conference on knowledge discovery and data mining*. ACM653–658.
Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., & Scholz, M. (2008). *One-class collaborative filtering. Data mining, 2008. ICDM'08. Eighth IEEEinternational conference on*. IEEE502–511.
Papadopoulos, S., Kompatsiaris, Y., Vakali, A., & Spyridonos, P. (2012). Community detection in social media. *Data Mining and Knowledge Discovery, 24*(3), 515–554.
Papadopoulos, S., Skusa, A., Vakali, A., Kompatsiaris, Y., & Wagner, N. (2009). Bridge bounding: A local approach for efficient community discovery in complex networks. arXiv preprint arXiv:0902.0871.
Peel, L., Larremore, D. B., & Clauset, A. (2017). The ground truth about metadata and community detection in networks. *Science Advances, 3*(5), e1602548.
Raghavan, U. N., Albert, R., & Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E, 76*(3), 036106.
Reichardt, J., & Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E, 74*(1), 016110.
Reid, F., McDaid, A., & Hurley, N. (2011). *Partitioning breaks communities. Advances in social networks analysis and mining (ASONAM), 2011 international conference on*. IEEE102–109.
Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40*(1), 185–197.
Shin, K., Jung, J., Lee, S., & Kang, U. (2015). *Bear: Block elimination approach for random walk with restart on large graphs. Proceedings of the 2015 ACMSIGMODinternational conference on management of data*. ACM1571–1585.
Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008). *Arnetminer: Extraction and mining of academic social networks. Proceedings of the 14th ACMSIGKDDinternational conference on knowledge discovery and data mining*. ACM990–998.
Tian, F., Gao, B., Cui, Q., Chen, E., & Liu, T.-Y. (2014). *Learning deep representations for graph clustering. AAAI*1293–1299.
Tong, H., Faloutsos, C., & Pan, J.-Y. (2006). *Fast random walk with restart and its applications. Sixth international conference on data mining (ICDM'06)*. IEEE613–622.

Usunier, N., Amini, M., & Patrick, G. (2004). *Boosting weak ranking functions to enhance passage retrieval for question answering. SIGIR 2004 workshop on information retrieval for question answering*.
Valizadegan, H., Jin, R., Zhang, R., & Mao, J. (2009). *Learning to rank by optimizing NDCG measure. Advances in neural information processing systems*1883–1891.
Wang, S., Minku, L. L., & Yao, X. (2015). Resampling-based ensemble methods for online class imbalance learning. *IEEE Transactions on Knowledge and Data Engineering, 27*(5), 1356–1368.
Wang, Y., Wang, L., Li, Y., He, D., Chen, W., & Liu, T.-Y. (2013). *A theoretical analysis of NDCG ranking measures. Proceedings of the 26th annual conference on learning theory (COLT 2013)8. Proceedings of the 26th annual conference on learning theory (COLT 2013)* 6–32.
Weiss, G. M. (2004). Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter, 6*(1), 7–19.
Whang, J. J., Gleich, D. F., & Dhillon, I. S. (2013). *Overlapping community detection using seed set expansion. Proceedings of the 22nd ACM international conference on conference on information & knowledge management.* ACM2099–2108.
Whang, J. J., Gleich, D. F., & Dhillon, I. S. (2016). Overlapping community detection using neighborhood-inflated seed expansion. *IEEE Transactions on Knowledge and Data Engineering, 28*(5), 1272–1284.
Wu, F., & Huberman, B. A. (2004). Finding communities in linear time: A physics approach. *The European Physical Journal B, 38*(2), 331–338.
Wu, Q., Burges, C. J., Svore, K. M., & Gao, J. (2008). *Ranking, boosting, and model adaptationTechnical Report*. Technical report, Microsoft Research.
Wu, Q., Burges, C. J., Svore, K. M., & Gao, J. (2010). Adapting boosting for information retrieval measures. *Information Retrieval, 13*(3), 254–270.
Wu, Y., Jin, R., Li, J., & Zhang, X. (2015). Robust local community detection: On free rider effect and its elimination. *Proceedings of the VLDB Endowment, 8*(7), 798–809.
Wu, Y.-J., Huang, H., Hao, Z.-F., & Chen, F. (2012). Local community detection using link similarity. *Journal of Computer Science and Technology, 27*(6), 1261–1268.
Xie, J., Kelley, S., & Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (CSUR), 45*(4), 43.
Yang, J., & Leskovec, J. (2012). *Community-affiliation graph model for overlapping network community detection. 2012 IEEE 12th international conference on data mining.* IEEE1170–1175.
Yang, J., & Leskovec, J. (2015). Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems, 42*(1), 181–213.
Yin, H., Benson, A. R., Leskovec, J., & Gleich, D. F. (2017). *Local higher-order graph clustering. Proceedings of the 23rd ACMSIGKDDinternational conference on knowledge discovery and data mining.* ACM555–564.
Zhang, T., & Wu, B. (2012). *A method for local community detection by finding core nodes. Proceedings of the 2012 international conference on advances in social networks analysis and mining (ASONAM 2012)*. IEEE Computer Society1171–1176.
Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2004). *Learning with local and global consistency. Advances in neural information processing systems*321–328.