



BIROn - Birkbeck Institutional Research Online

De Meo, P. and Levene, Mark and Messina, F. and Provetti, Alessandro (2019) A general centrality framework-based on node navigability. IEEE Transactions on Knowledge and Data Engineering , ISSN 1041-4347. (In Press)

Downloaded from: <https://eprints.bbk.ac.uk/id/eprint/30769/>

Usage Guidelines:

Please refer to usage guidelines at <https://eprints.bbk.ac.uk/policies.html>
contact lib-eprints@bbk.ac.uk.

or alternatively

A general centrality framework based on node navigability

Pasquale De Meo, Mark Levene, Fabrizio Messina, and Alessandro Provetti

Abstract—Centrality metrics are a popular tool in Network Science to identify important nodes within a graph. We introduce the *Potential Gain* as a centrality measure that unifies many walk-based centrality metrics in graphs and captures the notion of node navigability, interpreted as the property of being reachable from anywhere else (in the graph) through short walks. Two instances of the Potential Gain (called the *Geometric* and the *Exponential Potential Gain*) are presented and we describe scalable algorithms for computing them on large graphs. We also give a proof of the relationship between the new measures and established centralities. The geometric potential gain of a node can thus be characterized as the product of its Degree centrality by its Katz centrality scores. At the same time, the exponential potential gain of a node is proved to be the product of Degree centrality by its Communicability index. These formal results connect potential gain to both the “popularity” and “similarity” properties that are captured by the above centralities.

Index Terms—Centrality, Node Ranking in Graphs, Graph Navigability.

1 INTRODUCTION

Centrality metrics [1] provide a ubiquitous Network Science tool for the identification of the “important” nodes in a graph. They have been widely applied in a range of domains such as early detection of epidemic outbreaks [2], viral marketing [3], trust assessment in virtual communities [4], preventing catastrophic outage in power grids [5] and analysing heterogeneous networks [6].

Some centrality metrics define the importance of a node i in a graph \mathcal{G} as function of the distance of i to other nodes in \mathcal{G} : for instance, in *Degree Centrality*, the importance of i is defined as the number of the nodes which are adjacent to i , i.e. which are at distance one from i . Analogously, *Closeness Centrality* [7], [8] classifies as important those nodes which are few hops away from any other node in \mathcal{G} .

Another class of centrality metrics looks at walk/path structures in \mathcal{G} : for instance, the *Betweenness Centrality* [7] of a node depends on the number of shortest paths crossing it and, thus, nodes with largest betweenness centrality scores are those which intercept most of the shortest paths in \mathcal{G} . A further popular metric is the *Katz’s Centrality Score* [9], which is understood as the weighted number of walks terminating in the node. In Katz’s centrality metric the weighting factor is inversely related to walk length and, thus, long (resp., short) walks have a small (resp., large) weight. For a suitable choice of the weighting factor, the Katz centrality score

converges to the *Eigenvector Centrality* [8], [10] and to the *PageRank* [8], [11].

To the best of our knowledge, however, there is no previous work in which the centrality of a node is closely related to the notion of *navigability*, defined as the ease at which it is possible to reach a target node i regardless of the node j chosen as source node. Navigability is one of the most important features for a broad range of natural and artificial systems which have the transportation of information (e.g. a computer network) or the trade of goods (e.g. a road network) as their primary purpose.

Early studies on graph navigability were inspired by the seminal work of Travers and Milgram [12] on the “small world” property for social networks: in a celebrated experiment, randomly-chosen Nebraska residents were asked to send a booklet to a complete stranger in Boston. Selected individuals were required to forward the booklet to any of their acquaintances whom they deemed likely to know the recipient or at least might know people who did. In some cases, the booklet actually reached the target recipient by means, on average, of 5.2 intermediate contacts, thus suggesting an intriguing feature of human societies: in large, even planetary-scale, social networks, pairs of individuals are connected through *shorts chains of intermediaries* and ordinary people are able to uncover these chains [13], [14], [15]. Several empirical studies have revealed the small-world effect in diverse domains such as metabolic and biological networks [16], the Web graph [17], collaboration networks among scientists [18] as well as social networks [19], [20].

So far, centrality metrics and navigability have been investigated in parallel, yet their research tracks are disconnected. Thus, an important (and still unanswered) direction of inquiry is the introduction of centrality metrics that are related to the *navigability* of a node.

Our main contribution is to tackle the questions above by extending previous work by Fenner et al. [21] who studied navigability in the context of web surfing. The main output of our research is a general framework in which the

- P. De Meo is with the Department of Ancient and Modern Civilizations, University of Messina, Messina, Italy, 98166.
E-mail: pdemeo@unime.it
- M. Levene and A. Provetti are with the Department of Computer Science and Information Systems, Birkbeck, University of London, London WC1E 7HX, UK.
E-mail: {mark,ale}@dcs.bbk.ac.uk
- F. Messina is with the Department of Computer Science, University of Catania, Catania, Italy.
E-mail: messina@dmi.unict.it

Manuscript received December 18th, 2018; revised October 4th 2019.

Potential Gain unifies several walk-based centrality metrics and captures the notion of navigability.

The potential gain of a node i depends on the number of walks $w_k(j, i)$ of length k that connect i with any other node j . The underlying idea is that, for a fixed k , the larger $w_k(j, i)$, the higher the chance that j will reach i , regardless of the specific navigation strategy. In our model, the contribution of each walk to the potential gain shall decrease with its length k . This intuition is formalized by the introduction of a weighting factor $\phi(k)$ which monotonically decreases with k to penalize long walks.

We have developed two variants of the potential gain of [21], namely:

- the *Geometric Potential Gain*, in which $\phi(k)$ decays as δ^k , where δ is a parameter ranging between 0 and the inverse of the spectral radius λ_1 of \mathcal{G}^1 , and
- the *Exponential Potential Gain*, in which $\phi(k)$ decays in exponential fashion.

Both the geometric and exponential gain of i can be thought as the product of one index (Degree Centrality) related to the *popularity* of i and another (Katz Centrality score, for the geometric potential gain, and the Communicability Index [22], [23] for the exponential potential gain) which reflects the degree of *similarity* of i with all other nodes in the network. The combination of popularity and similarity has proven to closely resemble the way humans navigate large social networks [24] or attempt to locate information in large information networks such as Wikipedia [25], [26], [27].

Our formalisation applies the Neumann series expansion [28] to efficiently, yet accurately approximate both the geometric and exponential gain. Both theoretical and experimental analysis show that our approach is appropriate for accurately computing the geometric and exponential potential gain in large real-life graphs consisting of millions of nodes and edges, even with modest hardware resources.

We validated our approach on three large datasets: FACEBOOK (a graph of friendships among Facebook users), DBLP (a graph describing scientific collaboration among researchers in Computer Science) and YOUTUBE (a graph mapping friendship relationships among YouTube users). The main findings of our study can be summarized as follows:

- 1) The amount of time needed to compute the geometric or the exponential potential gain *does not depend* on the number of nodes/edges of a graph; instead, it depends on the spectral radius λ_1 : the larger λ_1 , the better connected the graph and, thus, the larger the number of walks needed to get a good approximation of the geometric/exponential potential gain.
- 2) For small values of δ , the geometric potential gain is highly correlated with Degree Centrality, while for large values of δ it displays a strong and positive correlation with Eigenvector Centrality.
- 3) In the case of the geometric potential gain, walks of small length (i.e., up to ten) are sufficient to obtain

a good approximation. In contrast, to compute the exponential potential gain our algorithm needed to construct longer random walks, in some cases up to ten times longer than those required for the computation of the geometric potential gain.

- 4) As a consequence of the above point, the geometric potential gain seems to be the most efficient solution for large graphs.

This article is organized as follows: in Section 2 we provide basic definitions that will be used throughout the paper. In Section 3 we review related work. Section 4 introduces the geometric and exponential potential gain and illustrates their main properties. In Section 5 we discuss how to efficiently calculate the geometric and exponential potential gain, while Section 6 details the experiments we have performed. Finally, in Section 7 we draw our conclusions.

2 BACKGROUND

In this section we introduce some basic terminology for graphs that will be largely used throughout this article.

Let a graph \mathcal{G} be an ordered pair $\mathcal{G} = \langle N, E \rangle$ where N is a set of *nodes*, here also called *vertices*, and $E = \{\langle i, j \rangle : i, j \in V\}$ is the set of *edges*. As usual, \mathcal{G} is *undirected* if edges are unordered pairs of nodes and *directed* otherwise. In this article we will consider only undirected graphs.

Also, let $n = |V|$ be the number of nodes, $m = |E|$ the number of edges of \mathcal{G} . For any given node i its neighborhood $\mathcal{N}(i)$ is the set of nodes directly connected to it; its *degree* d_i is the number of edges incident onto it, i.e., $d_i = |\mathcal{N}(i)|$.

A *walk* of length k (with k a non-negative integer) is a sequence of nodes $\langle i_0, i_1, \dots, i_k \rangle$ such that consecutive nodes are directly connected: $\langle i_\ell, i_{\ell+1} \rangle \in E$ for $\ell \in [0..k-1]$. Also, we use the term *path* for walks that do not have repeated vertices. A walk will be *closed* if it starts and ends at the same node.

We will represent graphs by their associated *adjacency matrix*, \mathbf{A} , defined as $a_{ij} = 1$ if $\langle i, j \rangle \in E$ and 0 otherwise. Sometimes we may slightly simplify notation with $a_{ij} = \mathbf{A}_{ij}$. The adjacency matrix provides a compact formalism to describe many graph properties: for instance, the matrix \mathbf{A}^2 where $a_{ij}^2 = \sum_{r=1}^n a_{ir} a_{rj}$, gives the number of walks of length two going from i to j . Inductively, for any positive integer k , the matrix \mathbf{A}^k will give the number of closed (resp., distinct) walks of length m between any two nodes i and j if $i = j$ (resp., if $i \neq j$) [29].

It is a well-know fact that the adjacency matrix of any undirected graph is *symmetric* and, hence, all its eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ are real. The largest eigenvalue λ_1 of \mathbf{A} is also called its *principal eigenvalue* or *spectral radius* of \mathcal{G} . Moreover, the corresponding eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ will form an orthonormal basis in \mathbb{R}^n [30]. Eigenpairs $\langle \lambda_i, \mathbf{v}_i \rangle$ are formed by the eigenvalue λ_i and the corresponding eigenvector \mathbf{v}_i .

3 RELATED WORK

The task of searching for and navigating in large networks has been extensively studied in the past for a broad range of domains such as routing in small world networks [12], locating pages in the World Wide Web [17], [31], finding the

1. Recall that the spectral radius of \mathcal{G} is defined as the largest eigenvalue of the adjacency matrix of \mathcal{G} .

most knowledgeable individual in an enterprise/academic social network, discovering resources in a P2P network and building recommender systems [32]. Inspired by the classification scheme introduced by Helic *et al.*, [27], we divide search and navigation into two main classes: (a) *Endogenous Search*. In this class, there are *multiple* agents embedded in the network and the navigation task is depicted as a decentralized decision process in which agents collaborate to discover a path in the network. Agents are assumed to have only a local knowledge of the network topology and, in addition, querying a neighboring node (e.g., to route a message) may have a non-negligible cost. (b) *Exogenous Search*. This class occurs whenever a user aims at navigating the Web [21], [33] or an information network such as Wikipedia [26], [34]. In exogenous search, there is *only one* agent involved in navigation task and it does not belong to the network. As in endogenous search, the agent (either human and artificial) only possesses local knowledge about the network topology. Unlike the endogenous search, however, the cost for visiting a node is generally low.

In the following two sections we review methods in endogenous and exogenous search (see Sections 3.1 and 3.2 below).

3.1 Endogenous search approaches

Many mathematical models have been proposed to explain why networks are, in an informal sense, *navigable*. Some of the best-known models are described in [13], [35], [36], [37]. The original Watts-Strogatz (WS) model [20] generated random graphs in which pairs of nodes belonging to distant parts of the graph may be connected through random edges, called *long-range weak ties*. The WS model was thus effective in forcing the graph to be “small,” i.e., to assure the presence of paths consisting of few edges between any pair of nodes. Nevertheless, the WS model alone is unable to explain why people are capable of discovering such paths.

One class of approaches to search large networks relies on the notion of *popularity*. Adamic *et al.* [38] is perhaps the best-known approach in that category: here the search task is modelled as a random walk on the graph \mathcal{G} . In a specific step, if the walker occupies a node l which is not the target one (and none of the neighbours of l is the target one), then the walker chooses the unvisited neighbour with largest degree.

Another class of approaches – called *similarity-based approaches* – exploit homophily to speed up search tasks [13], [37]. Kleinberg [35] described a generalization of the WS model to explain why decentralized search is effective in real networks. In Kleinberg’s model, nodes of a social network are arranged to form a bi-dimensional grid (called a *lattice*); each node is connected to its neighbours in the lattice and, in particular, the distance $d(u, v)$ between two nodes equals the number of grid steps separating them. As a result, each node v is connected to its four local contacts (i.e., nodes at distance one from v). In addition, a random edge—called a *long range edge*—connecting v with a node w is generated with probability proportional to $d(v, w)^{-q}$, q being the so-called clustering exponent of the model. Kleinberg proved that if $q = 2$, then the performance of decentralized search is optimal, i.e. there exists an algorithm that, on average, is

able to deliver a message from an arbitrary source node to an arbitrary target in $O(\log^2 n)$ time.

Another contribution is due to Watts *et al.* [37] who proposed a model to explain network navigability in which nodes aggregate into *groups* on the basis of some shared attributes such as job or geographic location. For each attribute, a population can be split into a hierarchical set of layers \mathcal{H} in which the top layer describes the entire population, while layers at increasing depth define a cognitive division of the population into more specific groups. Individuals can manage two kinds of information to decide to whom a message should be forwarded to: first, *social distance*, which accounts for the similarity of two nodes and, second, *network distance*, i.e., the number of network paths that can be detected by looking at the neighbors.

Social distance between two individuals i and j can be estimated by considering the groups i and j belong to and how distant these groups are in the layers of \mathcal{H} . As such, social distance is a kind of global metrics but, unfortunately, it is not a true distance in the sense that individuals belonging to close groups may be separated by long paths in the social network graph.

Network distance, on the other hand, is a true distance but a node only has access to a local portion of the network and, thus, it can correctly calculate network distances only for nodes which are separated from it by a few hops. By means of simulations [37] demonstrated that social distance is effective in approximating network distance and, thus, to successfully in directing messages across the network.

More recently Csirmecsek *et al.* [24] suggest to combine popularity- and similarity-based methods and, to this end, they describe a graph search algorithm that, at each step, forwards a message from the current node i to one of its neighbours, say j , such that the product $d_i \times q_{ij}$ is maximum; here q_{ij} quantifies homophily between i and j .

Unlike the approaches above, which assume that some attributes are available at each node, our approach only makes use of connectivity patterns to calculate the centrality of a node.

3.2 Exogenous search approaches

Exogenous search is mostly related to search in information networks such as a collection of Web pages or Wikipedia.

One of the most common search strategies on the World Wide Web (WWW) is *surfing*, in which a user moves from a Web page to another one by following hyperlinks. Huberman *et al.* [39] introduced a probabilistic model to describe surfing. In this model, the sequence of Web pages a user visits is regarded as the realisation of a random process and each Web page is associated with a value to the user. A user will stop surfing if the estimated cost of accessing a new Web page is bigger than the expected value of the information the user may get from accessing it.

More recently, West and Leskovec [40] analysed how people navigate an information network such as Wikipedia in order to reach a specific target. To this end, they used an online computation game, called *Wikispeedia* [25], in which Wikipedia information seekers are given two random articles and they are required to navigate from one to the other by clicking as few hyperlinks as possible. In

a subsequent paper [26], they compared the accuracy of several decentralized search algorithms and benchmarked them against the human navigation paths. Such a study highlighted two main phases of human navigation in information networks: (i) *Zoom-Out*: here, users strive to reach the network *core* (or a hub in the network core); such a core consists of a Wikipedia page with many links to other pages in Wikipedia. In this step, humans would prefer pages with many outgoing links (*high degree pages*). (ii) *Zoom-in*, in which users leave the core to get closer to a topic. Specifically, if we think of segmenting Wikipedia pages into clusters on the basis of their topics, such a phase would consist of entering into a cluster. In the zoom-in phase, users prefer to look for similar nodes in order to orient their search.

A nice approach to combine decentralized search was described by Helic *et al.* [27] who applied decentralized search algorithms such as those described in [35] to model human navigation in information networks. They considered an online navigation game (called *WikiGame*); in this game, a user starts from a random Wikipedia page and navigates to a target page. More than 250,000 click paths were collected and studied to determine the factors influencing players' decisions. The main finding in [27] is that two mechanisms regulate the way humans seek for information in large networks: i) *exploitation*, i.e., humans follow specific hyperlinks whenever they are confident enough that those will get them closer to the target they want, and ii) *exploration*, i.e., users navigate at random an information network, when their knowledge about how current links relates to a target Web page is insufficient. The quantitative analysis showed that exploration steps account, on average, for 15 – 20% of collected links, while exploitation accounted for the remaining 80 – 85% of collected links.

4 A MODEL OF NETWORK NAVIGABILITY

In this section we introduce our notion of network navigability, in Subsection 4.1 along with the Potential Gain as a new centrality measure and a new general framework that unifies many walk-based centrality indices and captures our notion of navigability. In Section 4.2 we describe two versions of the potential gain, called the *geometric* and the *exponential* potential gains. In Subsection 4.3 we compare the two versions of potential gain with other, well-known, centrality metrics from the literature. In Subsection 4.4 we investigate the relationship between the geometric and the exponential potential gain. Finally, 4.5 outlines our approach to calculating the geometric and exponential potential gain.

4.1 Network navigability and the potential gain

Starting from the Travers-Milgram's experiment [12], several studies have sought to characterize under which conditions a graph is deemed *navigable*. According to Kleinberg [35], a graph \mathcal{G} is *navigable* if i) its diameter is bounded by a polynomial in $\log n$ (here n is the number of nodes in \mathcal{G}) and ii) it has a strongly-connected component which contains almost all of the nodes of \mathcal{G} . Kleinberg's navigability is a property of the graph, whereas our purpose is to define the notion of navigability at the *node level*.

For navigability at the node level we leverage the main findings of Lamprecht *et al.* [41] on the navigability of a recommender network built on top of the Internet Movie Database (IMDB). Their work introduced some topological indices to quantify how difficult it is to navigate the recommender network. Specifically, they suggested to compute the *eccentricity* $ecc(j)$ of each node j , defined as the length of the longest shortest-path converging to j from any other node belonging to the same connected component of j . Thus, nodes with small eccentricity (also termed *efficient reachability*) can be easily reached from any other node of the graph.

Eccentricity may seem a good starting point for the formalisation of network navigability but "suffers" from some known issues. First, the eccentricity $ecc(j)$ is dominated by the distance, from j to the farthest node: as a consequence, j could have a large eccentricity even if it is close to almost all of nodes in \mathcal{G} . Second, any time the length of the shortest path connecting node i to j is above a threshold θ , the navigation from i to j is considered "hard," regardless of the topology.

The starting point for our work is the framework proposed by Fenner *et al.* [21] who studied the problem of identifying "good pages" from which to start exploring the Web. They classify a page p as a good starting point if it satisfies the following criteria: (1) *it is relevant*, i.e. the content of p closely matches user's information goals, (2) *page p is central*, i.e., the distance of p to other Web pages in the Web graph is as low as possible and (3) *page p is connected*, in the sense that p is able to reach a maximum number of other pages via its outlinks.

A key difference between Fenner *et al.* and our work is that they defined the navigability score of a page/a node as its ability to act as the *source node* for reaching all the other nodes. In our setting, instead, we consider the node as the *target* of search, as described next.

Let us fix a source node j and a target node i and provide an estimate $\tau(j, i)$ of how "easy" it will be for i to be reached if we choose j as source node. Intuitively, the larger the number of walks from j to i , the easier it would be to reach i when starting from j . In addition, assume that the task of exploring a graph is costly and such a cost increases as the length of the walks/paths we use for exploration purposes increases. Therefore, shorter walks should be preferred to longer ones. By combining the requirements above, we obtain:

$$\tau(j, i) = \sum_{k=1}^{+\infty} \phi(k) \cdot w_k(j, i) \quad (1)$$

here $w_k(j, i)$ is the number of walks of length k going from j to i and the non-increasing function $\phi(k)$ acts as penalty for longer walks. If we sum over all possible source nodes j , we obtain a global centrality index $pg(i)$ for i :

$$pg(i) = \sum_{j \in N} \tau(j, i). \quad (2)$$

In analogy to Fenner *et al.*, let $pg(i)$ be the *potential gain* of i .

In conclusion, the main differences between Potential Gain and Eccentricity are : i) the computation of PG is

grounded on walks while the computation of the eccentricity is based on paths, *ii*) the potential gain consider all walks converging to a target node j while the eccentricity consider only shortest paths reaching j and, *iii*) in the potential gain, the contribution of a walk of length k is penalized by a factor $\phi(k)$, while in the calculation of the eccentricity we take the length of the shortest path as is (i.e., with no penalization).

4.2 The geometric and exponential potential gain

Given the above specifications, we first define the potential gain in matrix notation. For the base case, consider walks of length $k=1$, i.e., direct connections. Only the neighbours of a node i will contribute to the potential gain of i , which leads to the trivial conclusion that, at $k = 1$, nodes with the largest degree are also those ones with the largest potential gain.

We define the vector \mathbf{p} such that $\mathbf{p}_i = pg(i)$ for every node i :

$$\mathbf{p} = \phi(1) \cdot \mathbf{A} \times \mathbf{1}. \quad (3)$$

If we include walks of length two, then we have to consider the squared adjacency matrix \mathbf{A}^2 . So, we add a contribution $\phi(2) \cdot \mathbf{A}^2 \times \mathbf{1}$ to the potential gain.

By induction, nodes capable of reaching from i through walks of length up to k provide a contribution to the potential gain equal to $\phi(k) \cdot \mathbf{A}^k \times \mathbf{1}$. By summing over all possible values of k we get to the following expression for \mathbf{p} :

$$\begin{aligned} \mathbf{p} &= \phi(1)\mathbf{A} \times \mathbf{1} + \phi(2)\mathbf{A}^2 \times \mathbf{1} + \dots + \phi(k)\mathbf{A}^k \times \mathbf{1} + \dots \\ &= \sum_{k=1}^{+\infty} (\phi(k)\mathbf{A}^k \times \mathbf{1}) = \left(\sum_{k=1}^{+\infty} \phi(k)\mathbf{A}^k \right) \times \mathbf{1} \end{aligned} \quad (4)$$

To attenuate the effect of the walks' length, we will consider two weighting functions, namely:

- 1) *Geometric*: $\phi(k) = \delta^{k-1}$ with $\delta \in (0, 1)$. So we define the *geometric potential gain*, \mathbf{g} :

$$\mathbf{g} = \left(\mathbf{A} + \delta\mathbf{A}^2 + \dots + \delta^{k-1}\mathbf{A}^k + \dots \right) \times \mathbf{1} \quad (5)$$

- 2) *Exponential*: $\phi(k) = \frac{1}{(k-1)!}$. So we define the *exponential potential gain*, \mathbf{e} :

$$\mathbf{e} = \left(\mathbf{A} + \mathbf{A}^2 + \dots + \frac{1}{(k-1)!}\mathbf{A}^k + \dots \right) \times \mathbf{1} \quad (6)$$

4.3 Relation to centrality measures

The geometric and the exponential potential gain introduced above yield a *ranking* of network nodes and, therefore, it is instructive to compare them with popular centrality metrics. Recall that we defined the *spectral radius* λ_1 of \mathbf{A} as the largest eigenvalue of \mathbf{A} .

As for the geometric potential gain, if we let $\delta < \lambda_1^{-1}$, the following expansion holds:

$$\begin{aligned} \mathbf{g} &= \left(\mathbf{A} + \delta\mathbf{A}^2 + \dots + \delta^{k-1}\mathbf{A}^k + \dots \right) \times \mathbf{1} \\ &= \mathbf{A} \times \left(\mathbf{I} + \delta\mathbf{A} + \dots + \delta^{k-1}\mathbf{A}^{k-1} + \dots \right) \times \mathbf{1} \quad (7) \\ &= \mathbf{A} \times (\mathbf{I} - \delta\mathbf{A})^{-1} \times \mathbf{1} \end{aligned}$$

in which we make use of the *Neuman series* [28]:

$$\left(\mathbf{I} + \dots + \delta^{k-1}\mathbf{A}^{k-1} + \dots \right) = (\mathbf{I} - \delta\mathbf{A})^{-1}. \quad (8)$$

At this point, the term $(\mathbf{I} - \delta\mathbf{A})^{-1} \times \mathbf{1}$ is exactly the *Katz centrality score* [9], [42], a popular centrality metric that defines the importance of a node as a function of its similarity with other nodes in \mathcal{G} . Hence, we can say that the geometric potential gain combines two kind of contributions: *popularity*, as captured by node degree, and *similarity* as captured by Katz's similarity score.

It is also instructive to consider what happens for extreme values of δ : if $\delta \rightarrow 0$, then the geometric potential gain tends to $\mathbf{A} \times \mathbf{1}$, i.e., it coincides with degree. In contrast, if $\delta \rightarrow \frac{1}{\lambda_1}$, then the Katz centrality score converges to *eigenvector centrality* [10], another popular metric adopted in Network Science. Boldi and Vigna [8] show that the Katz Centrality score is also strictly related to the PageRank. More specifically, the PageRank vector \mathbf{p} coincides with the Katz Centrality score provided that the adjacency matrix \mathbf{A} is replaced by its row-normalized version $\bar{\mathbf{A}}$:

$$\mathbf{p} = (1 - \alpha) \sum_{k=0}^{+\infty} \alpha^k \bar{\mathbf{A}}^k \times \mathbf{1} \quad (9)$$

Here, the parameter α is the so-called PageRank *damping factor*. Let us now concentrate on the exponential potential gain. We rewrite Equation 6 as follows:

$$\begin{aligned} \mathbf{e} &= \left(\mathbf{A} + \mathbf{A}^2 + \dots + \frac{1}{(k-1)!}\mathbf{A}^k + \dots \right) \times \mathbf{1} \\ &= \mathbf{A} \times \left(\mathbf{I} + \mathbf{A} + \dots + \frac{1}{k!}\mathbf{A}^k + \dots \right) \times \mathbf{1} \\ &= \mathbf{A} \times \exp(\mathbf{A}) \times \mathbf{1} \end{aligned} \quad (10)$$

where $\exp(\mathbf{A}) = \sum_{k=1}^{+\infty} \frac{1}{k!}\mathbf{A}^k$ is the exponential of \mathbf{A} [43].

The exponential of a matrix has been used to introduce other centrality scores such as *communicability* or *subgraph centrality* [10], [44].

Specifically, $\exp(\mathbf{A})_{ij}$ measures how easy is to send a unit of flow from a node i to a node j and vice versa. Such a parameter is known as *communicability* and it can be regarded as a measure of similarity between a pair of nodes. Communicability has been successfully used to discover communities in networks [44]. The product $\exp(\mathbf{A}) \times \mathbf{1}$ yields a centrality metric which defines the importance of a node as function of its ability to communicate with all other nodes in the network. In turn, the diagonal entry $\exp(\mathbf{A})_{ii}$ of the matrix exponential defines a further centrality metric called *subgraph centrality* [23]. As a result of the rewriting above, we clearly see the exponential potential gain as dependent on two factors: popularity of i (i.e., its degree) and similarity of i with all other nodes in the network.

The computation of the geometric (resp., exponential) potential gain for *all nodes* in \mathcal{G} needs the specification of the full adjacency matrix \mathbf{A} ; in this sense, the geometric and the exponential potential gain should be considered as *global centrality metrics*, on par with the Katz centrality score and subgraph centrality.

4.4 The relation between the geometric and the exponential potential gain

In this section we present some guidelines on how to choose the δ factor discussed in the previous section.

A straightforward choice would be to set $\delta = (2\lambda_1)^{-1}$ as in [9] or, in analogy with the Google PageRank damping factor, $\delta = 0.85\lambda_1^{-1}$ [22]. On the other hand, Foster et al. [45] suggested the following:

$$\delta = \frac{1}{\|\mathbf{A}\|_\infty + 1}$$

where $\|\mathbf{A}\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |\mathbf{A}_{ij}|$.

It is instructive to investigate the existence of a *crossover point* δ^c , i.e. to discover a value of δ at which the geometric and the exponential gain of a node i coincide. To this end, we provide the following result.

Theorem 1. *Let \mathcal{G} be a graph with adjacency matrix \mathbf{A} and eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$. For each node i and for $\delta \in (0, \lambda_1^{-1})$, the geometric and the exponential gains of i coincide if and only if one of the following holds:*

- 1) $\lambda_i = 0$, or
- 2) $\delta = \delta^c = \frac{e^{\lambda_i} - 1}{\lambda_i e^{\lambda_i}}$, provided that $\delta^c < \lambda_1^{-1}$.

Proof. Recall that for sufficiently large values of k , we can approximate the geometric and the exponential gain as follows:

$$\mathbf{g} = \mathbf{A} \times (\mathbf{I} - \delta \mathbf{A})^{-1} \times \mathbf{1} \quad \text{and} \quad \mathbf{e} = \mathbf{A} \times \exp(\mathbf{A}) \times \mathbf{1}$$

Recall that \mathbf{A} is a square and symmetric matrix. Thus, it admits the following eigendecomposition,

$$\mathbf{A} = \mathbf{D}^{-1} \times \mathbf{\Lambda} \times \mathbf{D}$$

where $\mathbf{\Lambda}$ is a diagonal matrix storing the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$ of \mathbf{A} and \mathbf{D} is an orthonormal matrix, whose columns coincide with the eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ of \mathbf{A} .

Now recall [43] that, for any function f , the matrix $f(\mathbf{A})$ is still diagonalisable and, for any eigenvalue λ_i of \mathbf{A} we have that $f(\lambda_i)$ is an eigenvalue of $f(\mathbf{A})$. In addition, the matrices \mathbf{A} and $f(\mathbf{A})$ share the same eigenvectors so we have $f(\mathbf{A}) = \mathbf{D}^{-1} \times f(\mathbf{\Lambda}) \times \mathbf{D}$.

Let us consider now the application of the two functions $f_1(x) = \frac{x}{1-\delta x}$ and $f_2(x) = x e^x$ to matrix \mathbf{A} . The eigenvalues of the matrix $f_1(\mathbf{A}) = \mathbf{A} \times (\mathbf{I} - \delta \mathbf{A})^{-1}$ are

$$\frac{\lambda_1}{1 - \delta \lambda_1}, \frac{\lambda_2}{1 - \delta \lambda_2}, \dots, \frac{\lambda_n}{1 - \delta \lambda_n} \quad (11)$$

whereas the eigenvalues of the matrix $f_2(\mathbf{A}) = \mathbf{A} \times \exp(\mathbf{A})$ are

$$\lambda_1 e^{\lambda_1}, \lambda_2 e^{\lambda_2}, \dots, \lambda_n e^{\lambda_n}. \quad (12)$$

Let us introduce $\mathbf{\Lambda}_g$ and $\mathbf{\Lambda}_e$, the diagonal matrices storing the eigenvalues of the matrices $\mathbf{A} \times \exp(\mathbf{A})$ and $\mathbf{A} \times (\mathbf{I} - \delta \mathbf{A})^{-1}$, respectively. Let us now compute the difference between the geometric and potential gain:

$$\begin{aligned} \mathbf{g} - \mathbf{e} &= \mathbf{A} \times (\mathbf{I} - \delta \mathbf{A})^{-1} \times \mathbf{1} - \mathbf{A} \times \exp(\mathbf{A}) \times \mathbf{1} \\ &= \mathbf{D}^{-1} \times \mathbf{\Lambda}_g \times \mathbf{D} \times \mathbf{1} - \mathbf{D}^{-1} \times \mathbf{\Lambda}_e \times \mathbf{D} \times \mathbf{1} \quad (13) \\ &= (\mathbf{D}^{-1} \times (\mathbf{\Lambda}_g - \mathbf{\Lambda}_e) \times \mathbf{D}) \times \mathbf{1} \end{aligned}$$

We focus on the i -th component of vector $\mathbf{g} - \mathbf{e}$ and observe that its value Δ_i is given as:

$$\Delta_i = \left(\lambda_i e^{\lambda_i} - \frac{\lambda_i}{1 - \delta \lambda_i} \right) \mathbf{u}_i^T \mathbf{u}_i = \left(\lambda_i e^{\lambda_i} - \frac{\lambda_i}{1 - \delta \lambda_i} \right)$$

Here we used the fact that eigenvectors of \mathbf{A} form an orthonormal basis. If we assume that $\lambda_i \neq 0$, then $\Delta_i = 0$ if and only if:

$$\delta = \frac{e^{\lambda_i} - 1}{\lambda_i e^{\lambda_i}} \quad (14)$$

which completes the proof. \square

4.5 Calculation of geometric and exponential potential Gains

In this section we present our algorithm for the computation of the geometric potential and exponential potential gain.

Our algorithm can be implemented in few lines of code in any high-level programming language, since it applies the expansion series provided in Equations 7 and 10. Our approach provides insight on how the walk length k affects the calculation of the geometric (resp., exponential) potential gain: indeed, if we stop the expansion of Equation 7 (resp. Eq. 10) after the first k terms, then, we would only include the walks up to length k in the calculation of the geometric (resp., exponential) potential gain.

Let us consider the computational complexity of our solution. We begin with the geometric potential gain and assume that we stop expanding the Neumann series after generating walks of length k^* . In such a case, it is easy to see that cost will be in $O(k^*|E|)$. In fact, for any j such that $1 < j < k^*$, let us set $\mathbf{y}_j = \delta^{j-1} \mathbf{A}^j \times \mathbf{1}$ and suppose that we have stored the sequence $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{j-1}\}$, with $\mathbf{y}_1 = \mathbf{1}, \mathbf{y}_2 = \mathbf{A} \times \mathbf{1}$.

Hence, the following recurrence holds:

$$\begin{aligned} \mathbf{y}_j &= \delta^{j-1} \mathbf{A}^j \times \mathbf{1} \\ &= (\delta \mathbf{A}) \times (\delta^{j-2} \mathbf{A}^{j-1} \times \mathbf{1}) \\ &= (\delta \mathbf{A}) \times \mathbf{y}_{j-1} \end{aligned} \quad (15)$$

The last equality states that any term \mathbf{y}_j can be calculated as the product of a sparse matrix $(\delta \mathbf{A})$ by a vector (\mathbf{y}_{j-1}) , already computed in the previous iteration. Such an operation takes $O(|E|)$ steps which, in the case of sparse networks, is $O(n)$.

Similarly, given that \mathbf{g} can be expressed as $\mathbf{g} \simeq \sum_{j=0}^{k^*} \mathbf{y}_j$, we conclude that the cost required to compute the geometric potential gain amounts is $O(k^*n)$. As for space complexity, the cost for computing \mathbf{g} is $O(|E|)$.

The computation of the geometric potential gain requires to fix δ beforehand, which, in turn, requires to fix an approximation of the spectral radius λ_1 . The literature on the estimation of λ_1 provides some bounds on it [46], [47] but, available upper bounds are often not tight and, thus, uninformative; therefore, an alternate way to approximate λ_1 is to rely on algorithms such as the *Power Iteration Method* [48]. On the other hand, if we target very large graphs, *sampling techniques* seem the best option [49].

Analogous results for both time and space complexity hold for the computation of the exponential potential gain

as we show next. Define a sequence $\mathcal{Z} = \{\mathbf{z}_i\}$ recursively as follows:

$$\begin{aligned} \mathbf{z}_1 &= \mathbf{1} \\ \mathbf{z}_2 &= \mathbf{A} \times \mathbf{1} \\ &\dots \\ \mathbf{z}_i &= \frac{1}{i-2} \mathbf{A} \times \mathbf{z}_{i-1} \end{aligned}$$

Therefore, any term \mathbf{z}_j can be calculated as the product of a sparse matrix (\mathbf{A}) by a vector (\mathbf{z}_{j-1}), which has been already computed in the previous iteration. Such an operation takes $O(|E|)$.

Given that \mathbf{e} can be expressed as $\mathbf{e} \simeq \sum_{j=0}^{k^*} \mathbf{z}_j$, we can conclude that the (worst-case) time complexity for the calculation of the exponential potential gain is $O(k^*|E|)$; similarly the space complexity is $O(|E|)$, hence for sparse graphs both time and space complexity reduce to $O(n)$.

5 METHODS FOR COMPUTING THE GEOMETRIC AND EXPONENTIAL POTENTIAL GAIN

In this section we prove that our algorithm to calculate the geometric and the exponential potential gain is convergent, and we provide an upper bound on the rate of convergence.

Let \mathbf{g} be the true value of the geometric potential gain and let \mathbf{g}_k be the approximate value of geometric potential gain we would obtain by considering walks up to length k . We wish to estimate:

$$\varepsilon_g(k) = \frac{\|\mathbf{g} - \mathbf{g}_k\|}{\|\mathbf{g}\|} \quad (16)$$

Analogously, the approximation error associated with the calculation of the exponential potential gain is given by

$$\varepsilon_e(k) = \frac{\|\mathbf{e} - \mathbf{e}_k\|}{\|\mathbf{e}\|} \quad (17)$$

The evaluation of $\varepsilon_g(k)$ and $\varepsilon_e(k)$ requires us to evaluate the *norm* of some matrices; here we will rely on the L_2 matrix norm (also known as *the spectral norm*), which, in case of symmetric matrices coincides exactly with λ_1 [48].

Since all matrix norms defined over a space of finite-dimension matrices are equivalent, our results generalize to other matrix norms; the only requirement is that the *sub-multiplicative property* holds, i.e., $\|\mathbf{X} \times \mathbf{Y}\| \leq \|\mathbf{X}\| \|\mathbf{Y}\|$ for any pair of matrices \mathbf{X} and \mathbf{Y} .

5.1 Rate of convergence for the geometric potential gain

Regarding the assessment of $\varepsilon_g(k)$, we note that the source of error in approximating the geometric potential gain depends on the early stopping of the Neumann series, i.e., on the approximation:

$$(\mathbf{I} - \delta\mathbf{A})^{-1} \simeq \mathbf{I} + \delta\mathbf{A} + \delta^2\mathbf{A}^2 + \dots \delta^k\mathbf{A}^k \quad (18)$$

Now, if we set $\mathbf{S}_k = \sum_{i=0}^k \delta^i\mathbf{A}^i$, the error $\varepsilon_g(k)$ depends on:

$$\begin{aligned} (\mathbf{I} - \delta\mathbf{A})^{-1} - \mathbf{S}_k &= (\delta\mathbf{A})^k + (\delta\mathbf{A})^{k+1} + \dots \\ &= (\delta\mathbf{A})^k \times (\mathbf{I} + \delta\mathbf{A} + \dots) \\ &= (\delta\mathbf{A})^k (\mathbf{I} - \delta\mathbf{A})^{-1} \end{aligned} \quad (19)$$

As $k \rightarrow \infty$ we obtain:

$$\begin{aligned} \left\| (\mathbf{I} - \delta\mathbf{A})^{-1} - \mathbf{S}_k \right\|^{\frac{1}{k}} &= \left\| (\delta\mathbf{A})^k (\mathbf{I} - \delta\mathbf{A})^{-1} \right\|^{\frac{1}{k}} \\ &\leq \left\| \delta^k \mathbf{A}^k \right\|^{\frac{1}{k}} \left\| (\mathbf{I} - \delta\mathbf{A})^{-1} \right\|^{\frac{1}{k}} \end{aligned} \quad (20)$$

Moreover, as $k \rightarrow +\infty$, $\left\| \delta^k \mathbf{A}^k \right\|^{\frac{1}{k}}$ converges to λ_1 [29]. This result, however, is rather weak as it does not give us a realistic estimation of the number of iterations that are required to assure that $\varepsilon_g(k) \leq \varepsilon$, for any $\varepsilon > 0$.

A more refined estimation of the rate of convergence of $\varepsilon_g(k)$ that applies to the general case of square complex matrices is due to Young [50], who provides a bound of the form $O(\lambda_1^{k-n}k^n)$; it depends on λ_1 , on the number k of iterations and, finally, on the size n of \mathbf{A} .

Since we are dealing with symmetric matrices, we can derive simpler bounds that are independent of the matrix size, as proved below.

Theorem 2. *Let \mathcal{G} be a graph with adjacency matrix \mathbf{A} and let λ_1 be its spectral radius; also let $\delta \in (0, \lambda_1^{-1})$. Then $\varepsilon_g(k) \rightarrow 0$ with convergence rate $(\delta\lambda_1)^k$.*

Proof. Recall that matrix \mathbf{A} is square and symmetric thus it admits the following eigendecomposition

$$\mathbf{A} = \mathbf{D}^{-1} \times \mathbf{\Lambda} \times \mathbf{D}$$

where $\mathbf{\Lambda}$ is a diagonal matrix storing the eigenvalues of \mathbf{A} and \mathbf{D} is an orthonormal matrix, whose columns coincide with the eigenvectors of \mathbf{A} .

In the light of the eigendecomposition of \mathbf{A} we get:

$$\begin{aligned} \left\| (\mathbf{I} - \delta\mathbf{A})^{-1} - \mathbf{S}_k \right\| &= \left\| \delta^k \mathbf{A}^k + \delta^{k+1} \mathbf{A}^{k+1} + \dots \right\| \\ &\leq \left\| \delta^k \mathbf{A}^k \right\| + \left\| \delta^{k+1} \mathbf{A}^{k+1} \right\| + \dots \\ &= \delta^k \left\| \mathbf{A}^k \right\| + \delta^{k+1} \left\| \mathbf{A}^{k+1} \right\| + \dots \end{aligned} \quad (21)$$

which can be further simplified by observing that, for any j :

$$\begin{aligned} \left\| \mathbf{A}^j \right\| &= \left\| \mathbf{D}^{-1} \mathbf{\Lambda} \mathbf{D} \right\|^j \leq \left\| \mathbf{D}^{-1} \right\| \left\| \mathbf{\Lambda}^j \right\| \left\| \mathbf{D} \right\|^j \\ &= \left\| \mathbf{\Lambda}^j \right\| \leq \left\| \mathbf{\Lambda} \right\|^j = \lambda_1^j \end{aligned} \quad (22)$$

Here we used the fact that \mathbf{D} and \mathbf{D}^{-1} are orthonormal so their L_2 norm is equal to 1. In addition, $\mathbf{\Lambda}$ has the same spectrum of \mathbf{A} hence its L_2 norm coincides with the spectral radius λ_1 of \mathbf{A} . By putting together the previous results we obtain:

$$\begin{aligned} \left\| (\mathbf{I} - \delta\mathbf{A})^{-1} - \mathbf{S}_k \right\| &\leq \delta^k \lambda_1^k + \delta^{k+1} \lambda_1^{k+1} + \dots \\ &= \delta^k \lambda_1^k (1 + \delta\lambda_1 + \delta^2\lambda_1^2 + \dots) \\ &= \delta^k \lambda_1^k \frac{1}{1 - \delta\lambda_1} \\ &= (\delta\lambda_1)^k \frac{1}{1 - \delta\lambda_1} \end{aligned} \quad (23)$$

as required. \square

5.2 Rate of convergence for the exponential potential gain

The convergence result obtained with Theorem 2 above has a counterpart for the exponential potential gain. In Theorem 3 below we give an exponential convergence result for the exponential potential gain case.

Theorem 3. *Let \mathcal{G} be a graph with adjacency matrix \mathbf{A} and let λ_1 be the spectral radius of \mathbf{A} . If k is at least $2e\lambda_1$ then*

$$\left(\frac{1}{2}\right)^{2e\lambda_1} \lambda_1^{-\frac{1}{2}}$$

is an upper bound for $\varepsilon_e(k)$.

Proof. Thanks to Equation 6 we can define:

$$\mathbf{S} = \sum_{i=1}^{+\infty} \frac{1}{i!} \mathbf{A}^i \quad \text{and} \quad \mathbf{S}_k = \sum_{i=1}^k \frac{1}{i!} \mathbf{A}^i$$

Next, we exploit the sub-multiplicativity property of the L_2 norm to obtain:

$$\begin{aligned} \|\mathbf{e} - \mathbf{e}_k\| &= \|\mathbf{A} \times \mathbf{S} \times \mathbf{1} - \mathbf{A} \times \mathbf{S}_k \times \mathbf{1}\| \\ &= \|\mathbf{A} \times (\mathbf{S} - \mathbf{S}_k) \times \mathbf{1}\| \\ &\leq \|\mathbf{A}\| \|\mathbf{R}_k\| \|\mathbf{1}\| \end{aligned} \quad (24)$$

with $\mathbf{R}_k = \mathbf{S} - \mathbf{S}_k$. Also, by repeated application of the triangle inequality we obtain:

$$\begin{aligned} \|\mathbf{R}_k\| &= \left\| \frac{\mathbf{A}^{k+1}}{(k+1)!} + \frac{\mathbf{A}^{k+2}}{(k+2)!} + \dots \right\| \\ &\leq \frac{\|\mathbf{A}^{k+1}\|}{(k+1)!} + \frac{\|\mathbf{A}^{k+2}\|}{(k+2)!} + \dots \end{aligned} \quad (25)$$

Recall that \mathcal{G} is undirected so its adjacency matrix \mathbf{A} is symmetric. Thus, the L_2 norm of \mathbf{A} coincides with its spectral radius. Also, for any $r \in \mathbb{N}$, \mathbf{A}^r is still symmetric and, due to the sub-multiplicativity of the L_2 norm, we get $\|\mathbf{A}^r\| \leq \lambda_1^r$, which allows us to simplify Equation 25 as follows:

$$\begin{aligned} \|\mathbf{R}_k\| &\leq \frac{\lambda_1^{(k+1)}}{(k+1)!} + \frac{\lambda_1^{(k+2)}}{(k+2)!} + \dots = \\ &= \frac{\lambda_1^{(k)}}{k!} \left[1 + \frac{\lambda_1}{k+1} + \frac{\lambda_1^2}{k(k+1)} + \frac{\lambda_1^3}{k(k+1)(k+2)} + \dots \right] \\ &\leq \frac{\lambda_1^{(k)}}{k!} \left[1 + \frac{\lambda_1}{k+1} + \frac{\lambda_1^2}{(k+1)^2} + \frac{\lambda_1^3}{(k+1)^3} + \dots \right] = \\ &= \frac{\lambda_1^k}{k!} \sum_{r=0}^{+\infty} \left(\frac{\lambda_1}{k+1}\right)^r \end{aligned} \quad (26)$$

Now, since $k+1 > k \simeq 2e\lambda_1 > \lambda_1$, we have that $\sum_{r=0}^{+\infty} \left(\frac{\lambda_1}{k+1}\right)^r$ converges to the constant value $\frac{k+1}{k+1-\lambda_1} \simeq \frac{2e}{2e-1}$.

The final step corresponds to applying Stirling's formula [51], which states that, for sufficiently large values of k , $k! \simeq \sqrt{2\pi k} \left(\frac{k}{e}\right)^k$, which implies $k! = 2\sqrt{\pi} \lambda_1^{\frac{1}{2}} (2\lambda_1)^{2e\lambda_1}$ if $k \simeq 2e\lambda_1$. Therefore, after some simplifications, we obtain

$$\frac{\lambda_1^k}{k!} \simeq \frac{1}{2\sqrt{\pi}} \left(\frac{1}{2}\right)^{2e\lambda_1} \lambda_1^{-\frac{1}{2}}$$

which completes the proof. \square

5.3 Computational Analysis

Many previous studies focused on the problem of efficiently calculating the product $F(\mathbf{A}) \times \mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$ and $F(\cdot)$ is an arbitrary function defined over the spectrum of \mathbf{A} [52].

In many applications \mathbf{A} is large and, thus, it is computationally prohibitive to first compute $F(\mathbf{A})$ and, then, to form the product $F(\mathbf{A}) \times \mathbf{b}$.

A clever strategy consists of projecting \mathbf{A} (resp., \mathbf{b}) onto a matrix \mathbf{H} (resp., a vector \mathbf{w}) of size $r \times r$ (resp., r) belonging to a subspace Ω such that r is much smaller than n : in this way, we estimate the product $F(\mathbf{A}) \times \mathbf{b}$ as $F(\mathbf{H}) \times \mathbf{w}$ which is much easier to compute.

The task of projecting \mathbf{A} and \mathbf{b} onto \mathbf{H} and \mathbf{w} is equivalent to constructing an orthonormal matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r]$ whose columns span Ω . If such a matrix \mathbf{V} is available, then \mathbf{H} is defined as $\mathbf{H} = \mathbf{V}^T \mathbf{A} \mathbf{V}$ and the vector \mathbf{b} is mapped onto the vector $\mathbf{w} = \mathbf{V}^T \times \mathbf{b}$.

The procedure for calculating \mathbf{V} depends on both the spectral features of \mathbf{A} as well as on the approximation accuracy we plan to obtain; in practice, the size of \mathbf{V} could be very large and, thus, optimization techniques have been extensively studied to generate good approximations of $F(\mathbf{A}) \times \mathbf{b}$. One of the most popular techniques is the *Krylov subspace* [48], [53]

Definition 1 (Standard Krylov Subspace). *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ matrix and let $\mathbf{b} \in \mathbb{R}^n$. The standard Krylov subspace of dimension t associated with \mathbf{A} and \mathbf{b} is defined as the*

$$\mathcal{K}_t(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{A} \times \mathbf{b}, \dots, \mathbf{A}^{t-1} \times \mathbf{b}\}$$

One could construct the standard Krylov subspace $\mathcal{K}_t(\mathbf{A}, \mathbf{b})$ and identify an orthonormal basis for $\mathcal{K}_t(\mathbf{A}, \mathbf{b})$; the vectors in such a basis constitute the columns of the matrix \mathbf{V} we would like to compute.

Since \mathbf{A} is symmetric the most efficient technique to find such an orthonormal basis for $\mathcal{K}_t(\mathbf{A}, \mathbf{b})$ is the *symmetric Lanczos algorithm* [48]. Such an algorithm performs ℓ iterations and, at each iteration, the most time-expensive step consists of calculating the product of the matrix \mathbf{A} by a vector of size n . If \mathbf{A} is large but sparse, the cost of each iteration is $O(n)$ and the overall cost is $O(\ell n)$ which is equal to the asymptotic cost of our algorithm.

A further, computationally-appealing option is to approximate $F(\cdot)$ by means of a rational function $g(x)$ defined as the ratio of two polynomials, $p_\nu(x)$ and $q_\mu(x)$, of degree ν and μ , respectively [52], [54]. In this way, the problem of calculating $F(\mathbf{A}) \times \mathbf{b}$ is equivalent to evaluate $g(\mathbf{A}) \times \mathbf{b}$ which should be hopefully easier to calculate. If we denote as $\sigma_1, \sigma_2, \dots, \sigma_\mu$ the poles of $g(x)$, i.e, the roots of $q_\mu(x)$, then we introduce the following definition:

Dataset	Nodes	Edges	λ_1	Year
1 Facebook Friendship	63 731	817 035	132.57	2009
2 DBLP co-authorship	317 080	1 049 866	115.85	2012
3 Youtube friendship	1 134 890	2 987 624	210.40	2012

TABLE 1

Key features of the datasets employed in our experimental tests. For each dataset we report the number of nodes, the number of edges, the spectral radius and the year in which data were collected.

Definition 2 (Rational Krylov Subspace). Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ matrix and let $\mathbf{b} \in \mathbb{R}^n$. The rational Krylov subspace of dimension t associated with \mathbf{A} and \mathbf{b} is defined as the

$$\mathcal{K}_t^*(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, (\mathbf{A} - \sigma_1 \mathbf{I})^{-1} \times \mathbf{b}, \dots, (\mathbf{A} - \sigma_t \mathbf{I})^{-1} \times \mathbf{b}\}$$

Rational Krylov subspace methods are often more accurate than standard ones but they require to solve, at each iteration, a linear system; such a task is equivalent to computing terms of the form $(\mathbf{A} - \sigma_i \mathbf{I})^{-1} \times \mathbf{b}$ and such a procedure is usually much more expensive than standard Krylov methods.

Theorems 2 and 3 indicate that our procedure is capable of achieving an exponentially decay error: consequently, we are able to achieve the accuracy we wish with a relatively small number of iterations.

6 EXPERIMENTAL VALIDATION

In this section we report on the experiments we carried out to assess the effectiveness of geometric and exponential potential gain on real-world datasets. Our experiments aim at answering the following questions:

- Q_1 : How sensitive are our approximations of the geometric and the exponential potential gain w.r.t. the length k of walks?
- Q_2 : Do our algorithms scale up to real graphs?
- Q_3 : How do the geometric and exponential potential gain correlate with other, popular, centrality metrics such as Degree, Katz, PageRank and Eigenvector Centrality? Are the aforementioned centrality metrics good candidates for assessing the navigability of a node in the sense defined in this paper?

To answer these questions, we considered three large, real datasets, taken from [55], whose features are described in Table 1. The first dataset – FACEBOOK – is a sample of the Facebook user connections graph: in it, a node represents a Facebook user and an edge represents a friendship between two users. The second dataset, called DBLP, is a sample of the DBLP computer science bibliography: authors correspond to nodes and two nodes are linked by an edge if the corresponding authors have published at least one paper together. Finally, YOUTUBE, is a sample of the friendship network between YouTube users.

We implemented our algorithms in Python (with the Scipy module) on a hardware platform with the following features: AMD Ryzen 5 1600 CPU, 16GB RAM and Ubuntu 17.10.

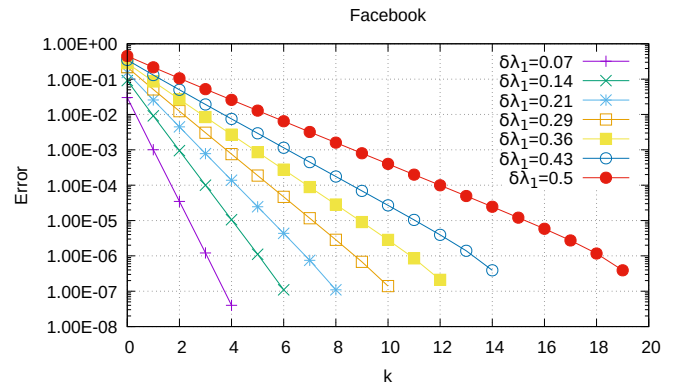


Fig. 1. Values of $\varepsilon_g(k)$ as a function of k plotted in semi-logarithmic scale for FACEBOOK

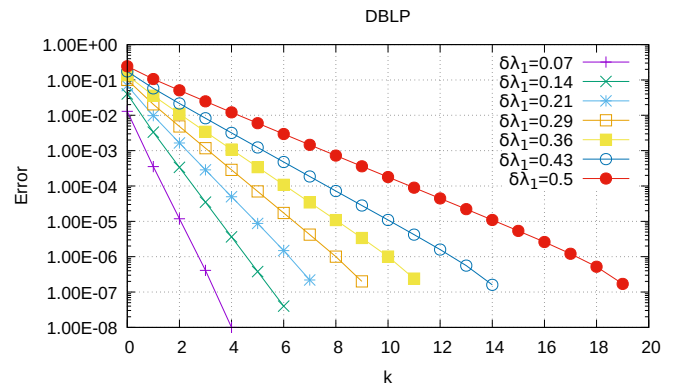


Fig. 2. Values of $\varepsilon_g(k)$ as a function of k plotted in semi-logarithmic scale for DBLP

6.1 The impact of walk length on the approximation of geometric/exponential potential gains

The aim of this section is to answer question Q_1 and, specifically, to study the quality of the approximation of the geometric and exponential potential gain in relation to walk length. Ideally, one would like geometric (resp., exponential) potential gain values to stabilize already for small values of k . It would mean that, for any node i , it suffices to consider nodes located a few hops away from i to get a satisfactory approximation of its geometric (resp., exponential) potential gain.

To perform our study we experimentally studied the decrease of $\varepsilon_g(k)$ and $\varepsilon_e(k)$ as function of k . We start by discussing the results for the geometric potential gain, in Figures 1, 2 and 3.

A notable feature of our algorithm is that convergence is very fast, independently of the size and nature of the dataset under investigation. For instance, with the YOUTUBE dataset– the largest tested here – walks up to length $k = 20$ are sufficient to achieve $\varepsilon_g(k)$ lesser than 10^{-6} .

A further observation is that, for a fixed value of k , the larger δ , the slower the convergence of $\varepsilon_g(k)$ to zero, and that our results perfectly agree with the statement of Theorem 2.

It is also instructive to study how $\varepsilon_g(k)$ varies across

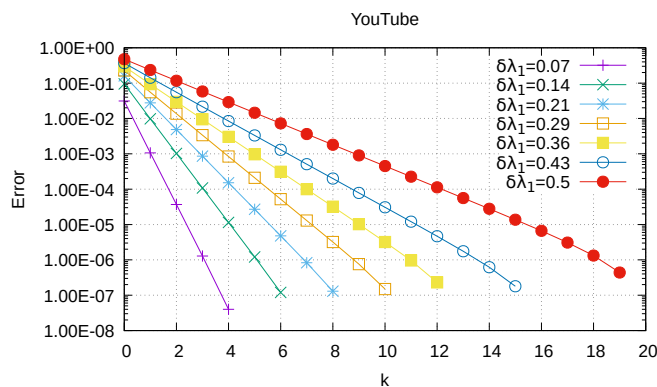


Fig. 3. Values of $\varepsilon_g(k)$ as a function of k plotted in semi-logarithmic scale for YOUTUBE

datasets: for any value of δ , $\varepsilon_g(k)$ for DBLP converges to zero faster than it does for FACEBOOK, despite the fact that DBLP is about five times larger than FACEBOOK (see Figures 2 and 3). It is also possible to appreciate small differences in the slopes of straight lines plotting $\log \varepsilon_g(k)$ as a function of k .

The important finding described above is mirrored by a similar result for the exponential potential gain, as illustrated by Figure 4. Once again we notice that graph size has a small impact on the convergence of our algorithm. From Theorem 2, in fact, λ_1 is the dominant parameter: the smaller λ_1 , the faster the algorithm converges to the true value of the exponential potential gain. Notice also how the exponential potential gain needs walks that are *longer* than those needed for the geometric potential gain: we need walks up to length 169, 189 and 279 for DBLP, FACEBOOK and YOUTUBE, respectively.

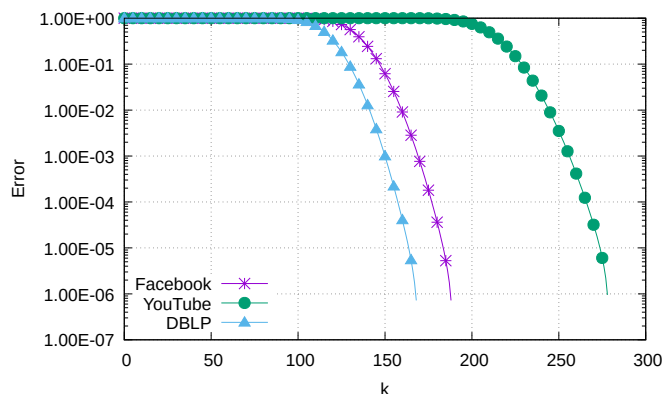


Fig. 4. Values of $\varepsilon_e(k)$ as a function of k for the FACEBOOK, DBLP, and YOUTUBE datasets plotted in semi-logarithmic scale.

The main results of our experimental validation can be now summarized as follows: (i) For the geometric potential gain, small values of δ should be used. Walks of length between 4 and 10 are sufficient to get a very good approximation. (ii) The time needed to compute the geometric and the exponential potential gain does not depend on the graph size but it only depends on λ_1 : the larger λ_1 the more dense/connected the graph is and, thus, a larger number of walks is needed to achieve a good approximation. (iii)

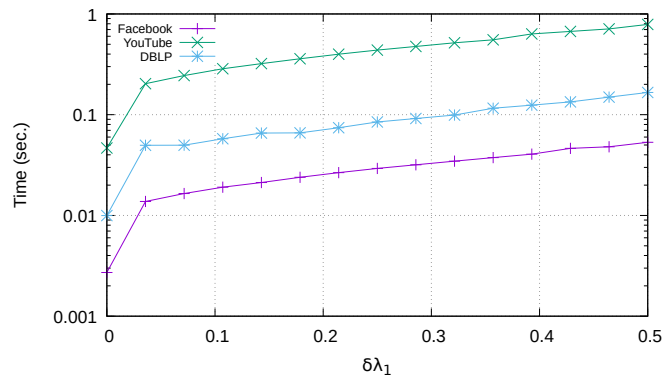


Fig. 5. Computation times vs. $\delta\lambda_1$ for the geometric potential gain

Computing the exponential potential gain is slower than computing geometric potential gain and, experimentally, it might require walks whose length is ten times larger. This finding suggests that we should consider as future work to introduce a decay factor of the form $\frac{\delta^k}{k!}$ to penalize long walks.

6.2 Scalability Analysis

In this section we address question Q_2 by studying how our algorithms scale over FACEBOOK, DBLP and YOUTUBE. We measured the execution times as a function of the walk length k . In particular, for the geometric potential gain we were concerned with understanding how δ affected the performance of our approach. The results we obtained are plotted in Figure 5.

Clearly, an increase of δ yields an increase in computational time. This is due to the fact that larger values of δ force our algorithm to explore the graph in more depth. Such effect can be clearly seen in Figure 5; the increase is approximately linear in δ so we can conclude that the computational impact of increasing δ is limited. Of course, larger datasets still require greater computational resources since we will have multiply the adjacency matrix \mathbf{A} by a vector. However, for sparse adjacency matrices the calculation is still fast, e.g., it takes less than one second for the three datasets considered here.

Table 2 reports the computation times of the exponential potential gain. Again we may notice how exponential potential gain is computationally more demanding than the geometric potential gain as we need many more iterations. In addition, notice how the computational time for DBLP is about three times slower than for FACEBOOK despite the fact that the latter needed 20 iterations more. Such difference depends on the difference in size between the two datasets.

Dataset	Time (sec.)
Facebook	0.525
DBLP	1.568
YouTube	11.878

TABLE 2

Computational times for the exponential potential gain

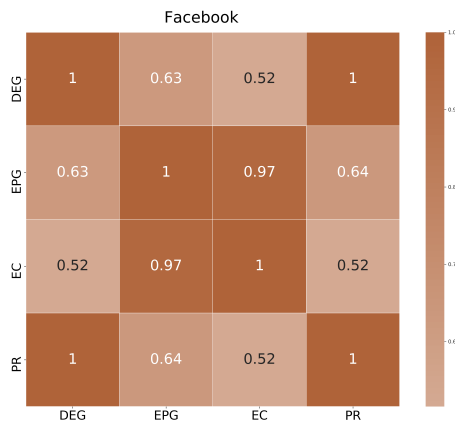


Fig. 6. Spearman's ρ correlation between *DEG*, *EPG*, *EC* and *PR* on the FACEBOOK dataset.

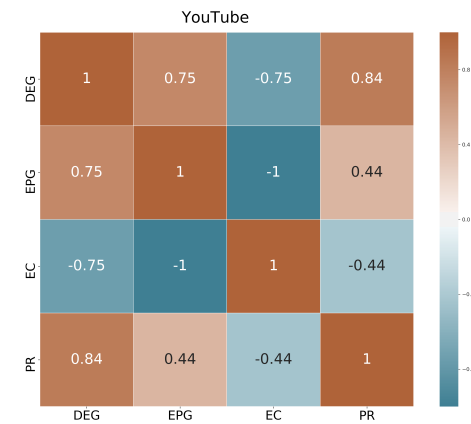


Fig. 8. Spearman's ρ correlation between *DEG*, *EPG*, *EC* and *PR* on the YOUTUBE dataset.

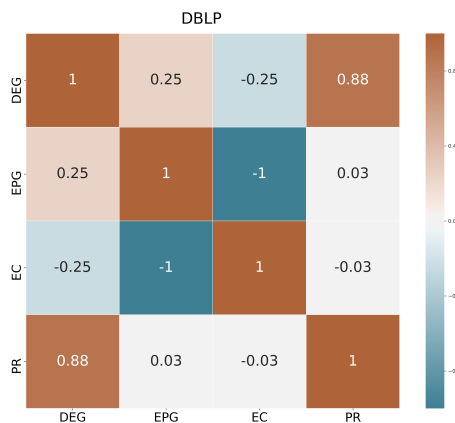


Fig. 7. Spearman's ρ correlation between *DEG*, *EPG*, *EC* and *PR* on the DBLP dataset.

6.3 Relation with other centrality metrics

In this section we investigate how the geometric and the exponential potential gain are correlated with some popular centrality metrics. Specifically, we compared the Geometric (*GPG*) and the Exponential Potential Gain (*EPG*) with Degree Centrality (*DEG*), Eigenvector Centrality (*EC*), PageRank (*PR*) and Katz Centrality (*Katz*). As for *PR*, we fixed the damping factor equal to 0.85. For a fair comparison, we choose the same value of δ for *GPG* and *Katz*. We used Spearman's ρ coefficient to calculate the correlation between the ranked lists of nodes that each of these centrality metrics generates on the datasets above.

Let us now analyse how *DEG*, *PR* and *EC* correlate each other and how they are related with *EPG*. (*EPG* is discussed later as it does not depend on any parameter). The main findings of our analysis are plotted in Figures 6 - 8 and can be summarized as follows:

- 1) *DEG* and *PR* are strongly positively correlated: we found Spearman's ρ correlation coefficients ranging from 0.84 (on YOUTUBE) to 1 (on FACEBOOK). Such

correlation is well-known in the literature because, in the case of undirected graphs, the *PR* scores of nodes are almost proportional to their degrees.

- 2) On FACEBOOK *EPG* and *EC* are highly correlated while at the same time display a perfect negative correlation on the other two datasets. This result is related to the property that the *EPG* of a node is always proportional to its communicability, as we have shown in Section 4.2. When the spectral gap between the largest and the second-largest eigenvalue, i.e., $\lambda_1 - \lambda_2$, is large, then communicability is, up to a constant factor, exactly equal to *EC*; see [22] for a detailed proof. As a consequence, for graphs with a large spectral gap the *EPG* of a node is proportional to its *EC*; this explaining the large correlation values we observed in our tests.
- 3) The topological features of the given graph have a huge impact on ρ : for instance, *DEG* and *EC* are positively correlated on FACEBOOK ($\rho = 0.52$) but they are negatively correlated on DBLP and YOUTUBE. Analogously, *EPG* and *PR* are positively correlated on FACEBOOK and YOUTUBE ($\rho = 0.64$ and 0.44 , respectively) but almost unrelated on DBLP.

Let us now consider the correlation between *DEG*, *EC*, *PR* and *EPG* and *GPG*; such correlation varies upon different choices of the parameter δ . Figures 9, 10 and 11, show how ρ varies as function of $\delta^* = \delta/\delta_{\max}$, where $\delta_{\max} = \frac{1}{\lambda_1}$. We can draw the following conclusions:

- 1) The *GPG* and *Katz* have an almost perfect correlation, as one would expect (red line). In fact, *GPG* and *Katz* differ by a multiplicative and constant factor given by the degree.
- 2) We note some interesting facts about the correlation of *GPG* and *EPG* (orange line). Firstly, if δ^* increases, then we generally have a (sometimes slight) increase in ρ . The correlation coefficient ρ is always positive but it ranges from 0.45 in DBLP to 1 (on FACEBOOK and YOUTUBE).

To explain the differences emerging across our datasets we observe the difference of the geometric

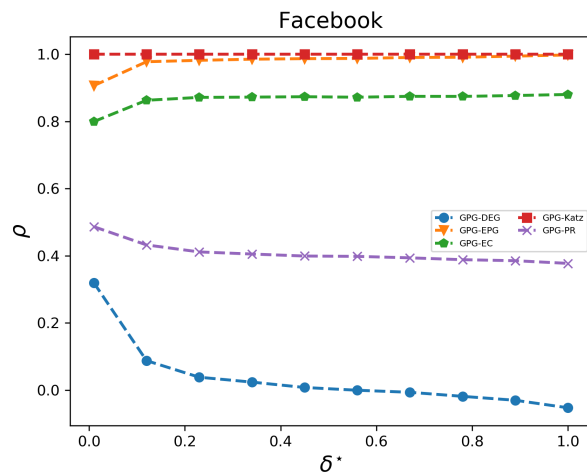


Fig. 9. Spearman's ρ correlation between GPG and DEG, EC, PR, Katz as function of δ^* on the FACEBOOK dataset.

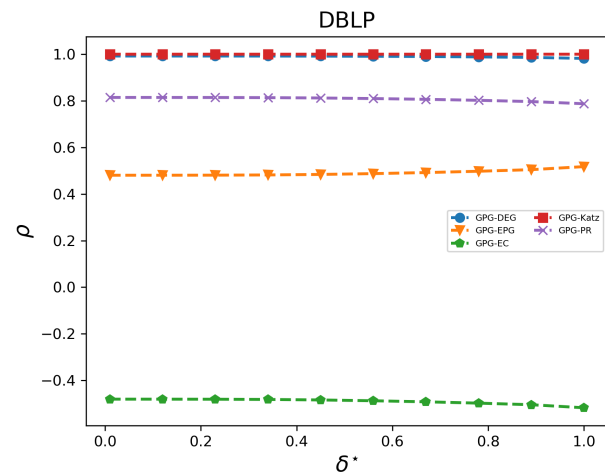


Fig. 10. Spearman's ρ correlation between GPG and DEG, EC, PR, Katz as function of δ^* on the DBLP dataset.

and the exponential potential gain of a node i depends on $\frac{\lambda_i}{1-\lambda_i} - \lambda_i e^{\lambda_i}$ (see Equation 13). Therefore, on the basis of the distribution of eigenvalues, we conclude that for some datasets *GPG* and *EPG* might be strongly and positively correlated while they are negatively correlated elsewhere.

- 3) As for the correlation between *EC* and *GPG* (green line), we find a positive correlation on FACEBOOK and a negative correlation on DBLP and YOUTUBE. In fact, for a fixed node, the *GPG* is proportional to the *Katz* score of that node; if $\delta^* \rightarrow 1$, then the *Katz* score converges to *EC* and, in this case, the *GPG* of a node is proportional to its *EC*. Similar considerations hold for *GPG* and *PR* (magenta line).
- 4) Consider now *DEG* (see blue line). An increase in δ^* generally causes a decrease in ρ , with the exception of the DBLP dataset where the correlation between *DEG* and *GPG* is almost perfect. As expected, when $\delta^* \rightarrow 0$ the *GPG* is well approximated by *DEG*. Vice versa, as δ^* increases, walks of length greater than one increasingly contribute to *GPG*, thus amplifying the difference between *DEG* and *GPG*.

The detailed analysis of the experimental results strongly suggests that the potential gain is a general framework, which unifies some of the main walk-based centralities.

7 CONCLUSIONS

We have introduced the potential gain, a general framework which captures the ability of a node to act as a target point for navigation within a graph and unifies several walk-based centrality indices in graphs. We have defined two variants of the potential gain, the geometric and exponential potential gain and proposed two iterative algorithms for each of them. The convergence of the algorithms was also proved; their scalability was tested against three real large datasets.

The present results provide a starting point to a better understanding of the theoretical properties of the potential gain. Previous research has sought to axiomatise centrality

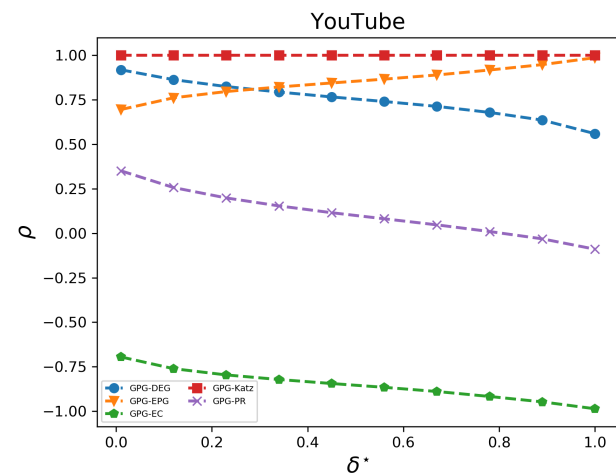


Fig. 11. Spearman's ρ correlation between GPG and DEG, EC, PR, Katz as function of δ^* on the YOUTUBE dataset.

and its metrics, and it would be extremely interesting to study which of these axioms are satisfied by the Potential gain. For instance, Boldi and Vigna [8] proposed three axioms, called *size*, *density* and *score monotonicity*, which may play a role in axiomatising the Potential Gain itself.

Another topic for future work is investigation of the relationship between network robustness and network navigability. To this end, we intend to design an experiment in which graph nodes are ranked on the basis of their geometric/exponential potential gain and then are progressively removed from the graph. Basic properties about graph topology, such as the number and size of connected components shall be re-evaluated upon node deletion. We also plan to study how adding edges can increase the geometric/exponential potential gain of a target group of nodes.

REFERENCES

[1] L. Lü, D. Chen, X. Ren, Q. Zhang, Y. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol.

- 650, pp. 1–63, 2016.
- [2] F. Chung, P. Horn, and A. Tsias, “Distributing antidote using Pagerank vectors,” *Internet Mathematics*, vol. 6, no. 2, pp. 237–254, 2009.
- [3] J. Leskovec, L. Adamic, and B. Huberman, “The dynamics of viral marketing,” *ACM Transactions on the Web*, vol. 1, no. 1, p. 5, 2007.
- [4] S. Agreste, P. De Meo, E. Ferrara, S. Piccolo, and A. Provetti, “Trust Networks: Topology, Dynamics, and Measurements,” *IEEE Internet Computing*, vol. 19, no. 6, pp. 26–35, 2015. [Online]. Available: <https://doi.org/10.1109/MIC.2015.93>
- [5] R. Alber, I. Albert, and G. Nakarado, “Structural vulnerability of the North American power grid,” *Physical review E*, vol. 69, no. 2, p. 025103, 2004.
- [6] S. Agreste, P. De Meo, E. Ferrara, S. Piccolo, and A. Provetti, “Analysis of a heterogeneous social network of humans and cultural objects,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 4, pp. 559–570, 2015.
- [7] M. Newman, *Networks: an introduction*. Oxford University Press, 2010.
- [8] P. Boldi and S. Vigna, “Axioms for centrality,” *Internet Mathematics*, vol. 10, no. 3–4, pp. 222–262, 2014.
- [9] L. Katz, “A new status index derived from sociometric analysis,” *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
- [10] M. Benzi and C. Klymko, “On the limiting behavior of parameter-dependent network centrality measures,” *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 686–706, 2015.
- [11] S. Brin and L. Page, “The anatomy of a large-scale hypertextual Web search engine,” *Computer networks and ISDN systems*, vol. 30, no. 1–7, pp. 107–117, 1998.
- [12] J. Travers and S. Milgram, “The small world problem,” *Psychology Today*, vol. 1, no. 1, pp. 61–67, 1967.
- [13] J. Kleinberg, “The small-world phenomenon: An algorithmic perspective,” in *Proc. of the ACM symposium on Theory of computing (STOC 2000)*. ACM, 2000, pp. 163–170.
- [14] S. Goel, R. Muhamad, and D. Watts, “Social search in “small-world” experiments,” in *Proc. of the International Conference on World Wide Web (WWW 2009)*, Madrid, Spain, 2009, pp. 701–710.
- [15] J. Leskovec and E. Horvitz, “Planetary-scale views on a large instant-messaging network,” in *Proc. of the International Conference on World Wide Web, WWW 2008*, Beijing, China, 2008, pp. 915–924.
- [16] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabasi, “The large-scale organization of metabolic networks,” *Nature*, vol. 407, no. 6804, p. 651, 2000.
- [17] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph structure in the Web,” *Computer Networks*, vol. 33, no. 1–6, pp. 309–320, 2000.
- [18] M. Newman, “The structure of scientific collaboration networks,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.
- [19] P. Dodds, R. Muhamad, and D. Watts, “An experimental study of search in global social networks,” *Science*, vol. 301, no. 5634, pp. 827–829, 2003.
- [20] D. Watts and S. Strogatz, “Collective dynamics of small-world networks,” *Nature*, vol. 393, no. 6684, p. 440, 1998.
- [21] T. Fenner, M. Levene, and G. Loizou, “Modelling the navigation potential of a Web page,” *Theoretical Computer Science*, vol. 396, no. 1–3, pp. 88–96, 2008.
- [22] M. Benzi and C. Klymko, “Total communicability as a centrality measure,” *Journal of Complex Networks*, vol. 1, no. 2, pp. 124–149, 2013.
- [23] E. Estrada and J. Rodriguez-Velazquez, “Subgraph centrality in complex networks,” *Physical Review E*, vol. 71, no. 5, p. 056103, 2005.
- [24] O. Simsek and D. Jensen, “Navigating networks by using homophily and degree,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, pp. 12758–12762, 2008.
- [25] R. West, J. Pineau, and D. Precup, “Wikispeedia: An Online Game for Inferring Semantic Distances between Concepts,” in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI 2009)*, Pasadena, California, USA, 2009, pp. 1598–1603.
- [26] R. West and J. Leskovec, “Automatic versus human navigation in information networks,” in *Proc. of the International Conference on Weblogs and Social Media, (ISWMC 2012)*, Dublin, Ireland, 2012.
- [27] D. Helic, M. Strohmaier, M. Granitzer, and R. Scherer, “Models of human navigation in information networks based on decentralized search,” in *Proc. of the ACM conference on Hypertext and Social Media*. Paris, France: ACM, 2013, pp. 89–98.
- [28] R. Horn and C. Johnson, *Matrix analysis*, 2nd ed. Cambridge Univ. Press, 2013.
- [29] D. Cvetkovic, P. Rowlinson, and S. Simic, *Eigenspaces of graphs*. Cambridge University Press, 1997.
- [30] G. Strang, *Introduction to linear algebra*. Wellesley-Cambridge Press Wellesley, MA, 1993, vol. 3.
- [31] F. Menczer, “Growing and navigating the small world web by local content,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 22, pp. 14014–14019, 2002.
- [32] D. Lamprecht, M. Strohmaier, and D. Helic, “A method for evaluating the navigability of recommendation algorithms,” in *Proc. of the International Workshop on Complex Networks and their Applications*. Springer, 2016, pp. 247–259.
- [33] M. Levene and R. Wheeldon, “Navigating the World Wide Web,” in *Web Dynamics*, M. Levene and A. Poulouvassilis, Eds. Springer, 2004, pp. 117–151.
- [34] D. Lamprecht, K. Lerman, D. Helic, and M. Strohmaier, “How the structure of Wikipedia articles influences user navigation,” *New Review of Hypermedia and Multimedia*, vol. 23, no. 1, pp. 29–50, 2017.
- [35] J. Kleinberg, “Navigation in a small world,” *Nature*, vol. 406, no. 6798, p. 845, 2000.
- [36] —, “Small-world phenomena and the dynamics of information,” in *Proc. of the International Conference on Advances in Neural Information Processing Systems (NIPS 2001)*, Vancouver, British Columbia, Canada, 2002, pp. 431–438.
- [37] D. Watts, P. Dodds, and M. Newman, “Identity and search in social networks,” *Science*, vol. 296, no. 5571, pp. 1302–1305, 2002.
- [38] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman, “Search in power-law networks,” *Physical Review E*, vol. 64, no. 4, p. 046135, 2001.
- [39] B. Huberman, P. Pirollo, J. Pitkow, and R. Lukose, “Strong regularities in World Wide Web surfing,” *Science*, vol. 280, no. 5360, pp. 95–97, 1998.
- [40] R. West and J. Leskovec, “Human wayfinding in information networks,” in *Proc. of the International Conference on World Wide Web (WWW 2012)*. Lyon, France: ACM, 2012, pp. 619–628.
- [41] D. Lamprecht, F. Geigl, T. Karas, S. Walk, D. Helic, and M. Strohmaier, “Improving recommender system navigability through diversification: A case study of IMDB,” in *Proc. of the International Conference on Knowledge Technologies and Data-driven Business, I-KNOW ’15*, Graz, Austria, 2015, pp. 21:1–21:8.
- [42] E. Leicht, P. Holme, and M. Newman, “Vertex similarity in networks,” *Physical Review E*, vol. 73, no. 2, p. 026120, 2006.
- [43] N. Higham, *Functions of matrices: theory and computation*. SIAM, 2008, vol. 104.
- [44] E. Estrada, N. Hatano, and M. Benzi, “The physics of communicability in complex networks,” *Physics reports*, vol. 514, no. 3, pp. 89–119, 2012.
- [45] K. Foster, S. Muth, J. Potterat, and R. Rothenberg, “A faster Katz status score algorithm,” *Computational & Mathematical Organization Theory*, vol. 7, no. 4, pp. 275–285, 2001.
- [46] K. Das and P. Kumar, “Some new bounds on the spectral radius of graphs,” *Discrete Mathematics*, vol. 281, no. 1–3, pp. 149–161, 2004.
- [47] D. Stevanovic, *Spectral radius of graphs*. Academic Press, 2015.
- [48] M. Heath, *Scientific Computing*. McGraw-Hill New York, 2002.
- [49] G. Han and H. Sethu, “Closed walk sampler: An efficient method for estimating the spectral radius of large graphs,” in *Proc. of the IEEE International Conference on Big Data*. Boston, USA: IEEE, 2017, pp. 616–625.
- [50] N. Young, “The rate of convergence of a matrix power series,” *Linear Algebra and its Applications*, vol. 35, pp. 261–278, 1981.
- [51] D. Knuth, R. Graham, and O. Patashnik, “Concrete mathematics,” Addison Wesley, 1989.
- [52] M. Popolizio and V. Simoncini, “Acceleration techniques for approximating the matrix exponential operator,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 657–683, 2008.
- [53] Y. Saad, “Analysis of some Krylov subspace approximations to the matrix exponential operator,” *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 209–228, 1992.
- [54] S. Guttel, “Rational krylov approximation of matrix functions: Numerical methods and optimal pole selection,” *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 8–31, 2013.
- [55] J. Kunegis, “Konec: the Koblenz network collection,” in *Proc. of the International Conference on World Wide Web (WWW 2013)*. Rio de Janeiro, Brazil: ACM, 2013, pp. 1343–1350.



Pasquale De Meo is associate professor of Computer Science at the Department of Ancient and Modern Civilizations at the University of Messina, Italy. His main research interests are in the area of social networks, recommender systems, and user profiling. His PhD thesis was selected as the Best Italian PhD thesis in Artificial Intelligence by the AI*IA (Italian Association for Artificial Intelligence). He has been Marie Curie Fellow at Vrije Universiteit Amsterdam. He serves as Associate Editor for IEEE Transactions

on Cybernetics and IEEE ACCESS. More information about his research can be found at http://dblp.uni-trier.de/pers/hd/m/Meo:Pasquale_De.



Fabrizio Messina received his PhD in Computer Science from Department of Informatics and Mathematics, University of Catania in 2009. He is currently Assistant Professor in the same department. His main research interests are Grid and Cloud Computing, trust and recommender systems in federated cloud and grid, software platforms and simulations Autonomous vehicle control and flocking.



Mark Levene received his PhD in Computer Science in 1990 from Birkbeck College, University of London, having previously been awarded a BSc in Computer Science from Auckland University New Zealand in 1982. Following his PhD he was a faculty member in the Department of Computer Science at University College London (UCL). In 2001 he joined Birkbeck College as a Professor of Computer Science, where he is the head of department and a member of the Experimental Data Science research group. His

main research interests are web search and navigation, web data mining and applied machine learning, and human dynamics/sociophysics. He has published extensively in these areas, and has recently published a book called *An Introduction to Search Engines and Web Navigation*.



Alessandro Provetti studied Informatics at Milan, Imperial College and Bologna. His main research results have been in Computational logic and more recently in Computational Social Science. He has held academic positions at U. of Texas El Paso, Milan, Messina and now Birkbeck, University of London. He is currently the director of the Birkbeck Institute for Data Analytics, an institute that supports interdisciplinary research.