INGI
Département
d'ingénierie
informatique

Relevant subgraph extraction from random walks in a graph

P. Dupont, J. Callut, G. Dooms, J.-N. Monette and Y. Deville

Research Report RR 2006-07
November 2006

Abstract

This paper describes novel methods for extracting a subgraph that best captures the relationships between k given nodes of interest (or seed nodes) in a graph. We introduce a betweenness measure based on random walks connecting distinct nodes of interest. Expected node and edge passage times along these walks can be efficiently computed. These quantities, which are defined here relatively to the choice of seed nodes, overcome limitations of shortest paths or maximal flow approaches. The proposed technique applies both to directed or undirected connected graphs. Additional variants of our approach include the consideration of all walks of a given length, or up to a maximal length, and extension to groups of *a priori* related seed nodes rather than individual nodes.

UNIVERSITAS • CATHOLICA • LOVANIENSIS • SEDES • SAPIENTIAE • 1425

Faculté des sciences appliquées

UCL Université catholique de Louvain

Relevant subgraph extraction from random walks in a graph

P. Dupont[1,2], J. Callut[1,2], G. Dooms[1], J-N. Monette[1], Y. Deville[1]

Research Report UCL/FSA/INGI RR 2006-07

[1]Department of Computing Science and Engineering (INGI)
Université catholique de Louvain
Place Sainte Barbe, 2
B-1348 Louvain-la-Neuve - Belgium
Corresponding author: Pierre.Dupont@uclouvain.be
[2] UCL Machine Learning Group
http://www.ucl.ac.be/mlg/

Abstract

This paper describes novel methods for extracting a subgraph that best captures the relationships between k given nodes of interest (or seed nodes) in a graph. We introduce a betweenness measure based on random walks connecting distinct nodes of interest. Expected node and edge passage times along these walks can be efficiently computed. These quantities, which are defined here relatively to the choice of seed nodes, overcome limitations of shortest paths or maximal flow approaches. The proposed technique applies both to directed or undirected connected graphs. Additional variants of our approach include the consideration of all walks of a given length, or up to a maximal length, and extension to groups of *a priori* related seed nodes.

1 Introduction

We address here the problem of extracting a subgraph that best explains the relationships between k (≥ 2) given nodes of interest in a graph. We are considering for instance a large metabolic network which can be represented as a directed connected graph[1] of reactions and bioentities involved in these reactions.

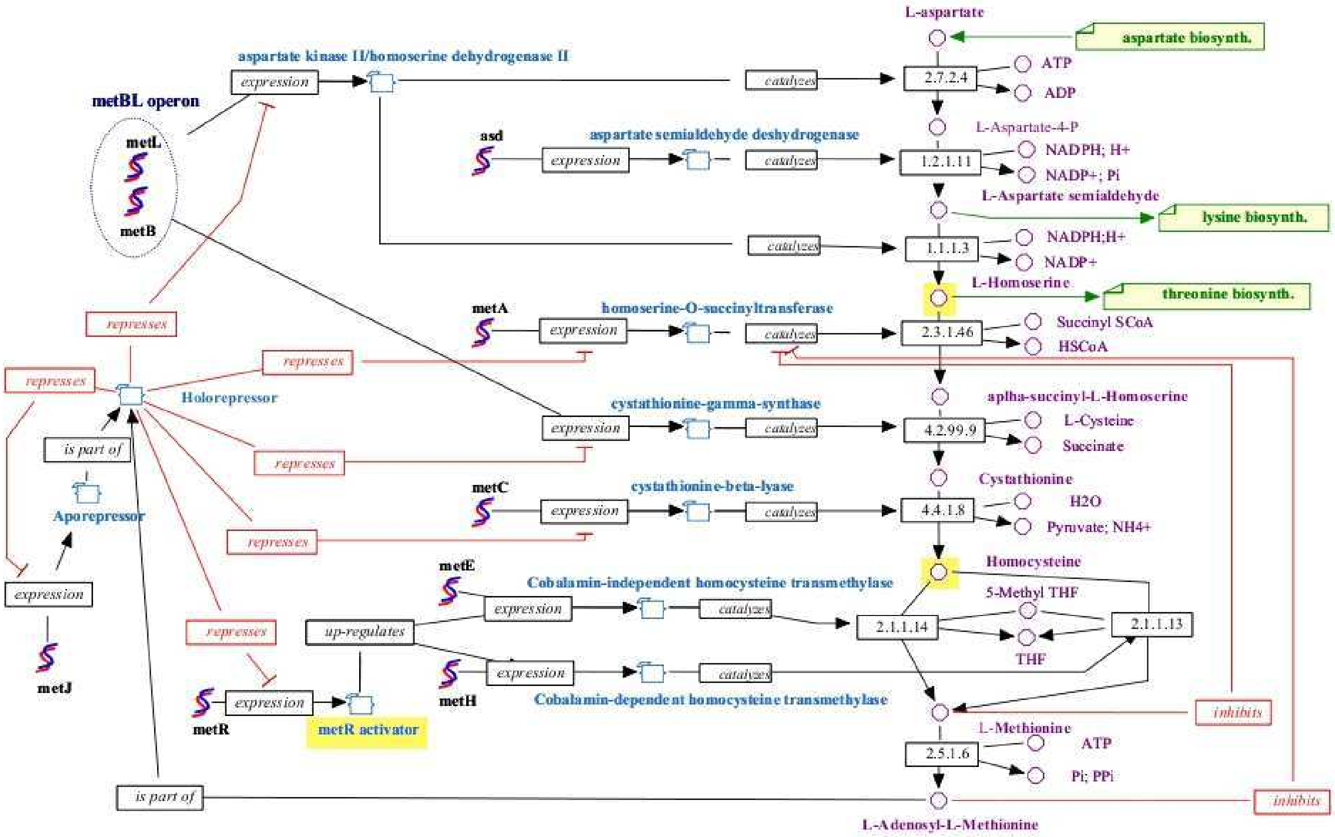

Figure 1: Methionine Biosynthesis of E. Coli.

More specifically, we could analyze the Methionine Biosynthesis of *Escherichia Coli*. A simplified view of this metabolic pathway[2] is depicted on Figure 1. One could study in particular the influence of the *metR Activator* on the production of *Homocysteine* from *L-Homoserine* in this pathway. In this case, there are 3 nodes of interest (depicted in yellow) and we would like

[1] Several representations exist for metabolic networks (see *e.g.* [Deville, 03]). The approach described here can be applied whether the network is represented as a directed or undirected graph, bipartite or not, and as long as it is connected. The unconnected case is briefly discussed in section 5.

[2] This figure has been kindly provided to us by J. Van Helden from the *Service de Conformation des Macromolécules Biologiques et de Bioinformatique* of the *Université Libre de Bruxelles*.

to extract a relevant subgraph explaining the relationships between these 3 nodes.

Another typical application of the proposed methods is the traffic analysis of vehicles running between several locations. The road network can be modeled as a graph in which nodes represent locations. Each road between any pair of adjacent locations can be characterized by an edge weight. The higher the weight between two nodes the easier the immediate connection between them. A connection weight is typically inversely proportional to the time to travel from one node to the other or to the distance between them. According to the chosen modeling hypothesis, the corresponding weighted graph can either be directed or undirected. In the latter case, the weight also corresponds to the notion of *conductance* in an electrical network as detailed in section 7.

While analyzing the traffic over the whole network, one would like to extract the most relevant routes to connect a predefined subset of k, non-necessarily adjacent, locations. The extracted subgraph should not only reflect the shortest path(s) between any pair of locations of interest but how the overall traffic can be distributed while connecting them.

Consider, for instance, the graph[3] depicted on Figure 2. The nodes represent particular locations on a roadmap. Edge weights correspond here to the inverse distance between nodes. Since distances are symmetric the graph is undirected. In this particular example, the weight of any (existing) edge is assumed to be equal to 1. For the simplicity of the reasoning, we consider firstly only two nodes of interest (1 and 9). They typically identify two distinct locations, each one belonging for example to a specific suburb (a highly connected cluster of locations). The best route between 1 and 9 corresponds here to the shortest distance path 1-3-6-9 or, equivalently, 9-6-3-1.

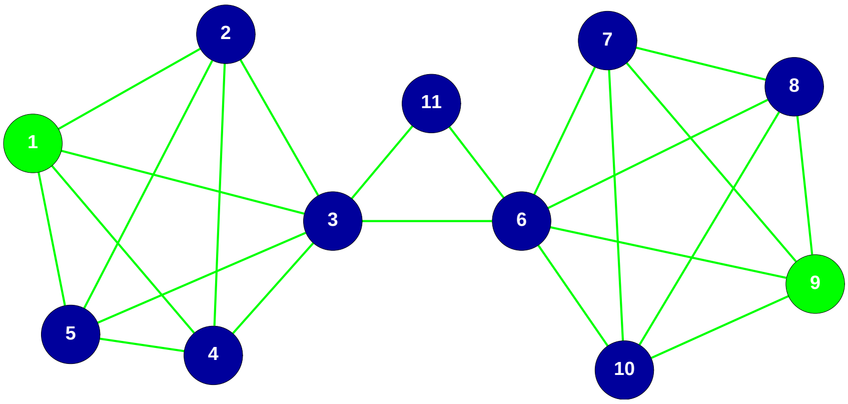

Figure 2: A roadmap describing the possible routes between a set of locations.

Suppose we would like to know the second best route (in case of a potential traffic jam along the 3-6 edge, for instance). If one considers the second

[3] This example is inspired from [Newman, 05].

shortest path(s), there are many routes having the same total distance (equal to 4 in this case): 1-4-3-6-9, 1-3-6-8-9, 1-3-11-6-9, … One can thus apply an algorithm for finding the m-shortest paths [Eppstein, 99, Jiménez, 99] but this algorithm would consider all routes of length 4 as equally important.

The unique second best route in this particular example is arguably 1-3-11-6-9 as it forms the unique best alternative to the direct edge 3-6 connecting two cut nodes. In other words, a random walker starting from node 1 and reaching eventually node 9 (or the converse) is more likely to go through node 11 than any other node as an alternative to the best route. Conversely, a random walker following the 1-2 edge instead of 1-3 is less likely to choose 2-3 as the next edge, simply because there are many other options to leave node 2. The proposed approach is precisely based on the expected number of times a given node or a given edge is used along any random walk connecting the nodes of interest.

The resulting graph is depicted on Figure 3 where each edge width denotes its relative frequency of use along these walks. This relative frequency is interpreted as its *relevance* to explain the relationships between the nodes of interest. Discarding any edge the relevance of which falls below a threshold defines a relevant subgraph. A more stringent threshold would result in a smaller subgraph, *e.g.* the subgraph induced by the nodes $\{1, 3, 6, 9, 11\}$. The resulting subgraph obtained by such a thresholding needs not be connected however. This connectivity issue is further discussed in section 5.

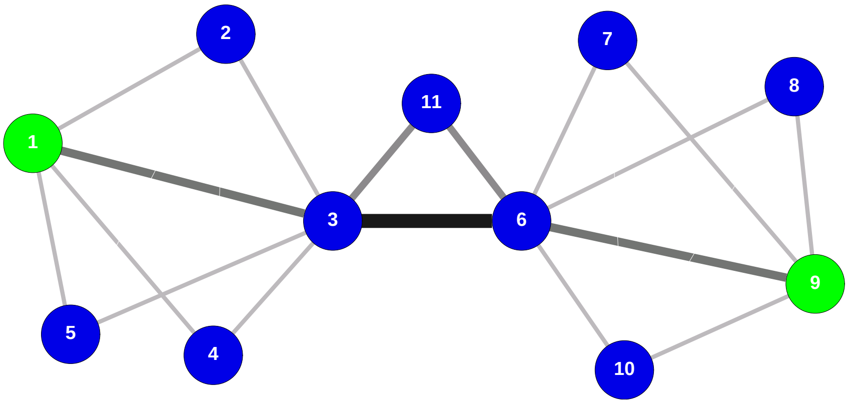

Figure 3: Relative edge relevance based on random walks between nodes 1 and 9.

The random walk method proposed here also overcomes limitations introduced by a maximal flow approach. Consider for instance another roadmap[4] depicted on Figure 4. We are interested in relevant ways to connect nodes 1 and 8, each one belonging to a separate cluster[5]. All (existing) edge weights are again assumed to be equal to 1. These weights can also be interpreted as

[4] This example is also inspired from [Newman, 05].

[5] These clusters form cliques in the present case but this is not a mandatory feature.

the flow capacity between any pair of adjacent nodes. There are alternatives routes to connect the two clusters but there is a maximal flow of 2 units from one cluster to the other. Nodes such as 12 or 15 can each one capture one unit of flow (for instance, from left to right) and thus would be considered as important to connect the two nodes (or clusters) of interest. The node 17 would not be considered as important since flowing through this node would limit the total flow between the two clusters to 1 unit. This is not necessarily appropriate since there are routes through 17 which can be considered as good alternatives to the "straight" routes. In particular, 1-3-11-17-16-10-8 also defines a relevant shortest path.

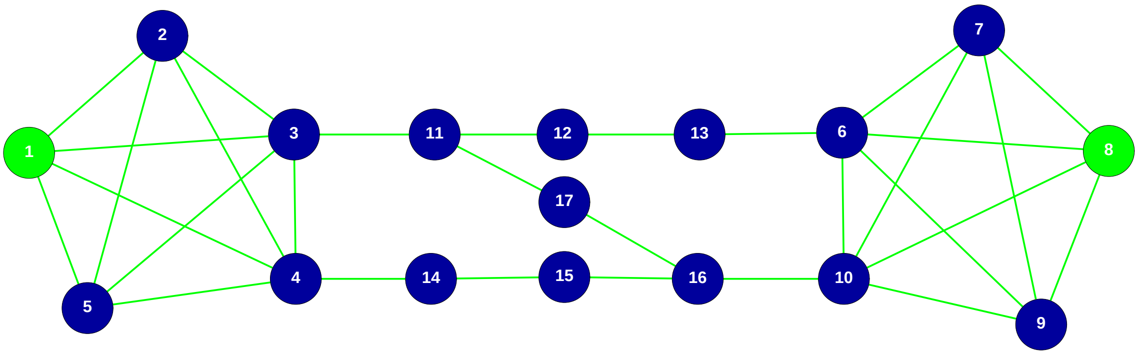

Figure 4: Another roadmap example.

Relative edge relevances based on the proposed random walk approach is depicted on Figure 5. The edges 3-11 and 16-10 are the most important since any left-to-right route between the two clusters must at least pick one of them, sometimes both. The edges 11-17 and 17-16 are non negligible as they are part of important alternative routes.

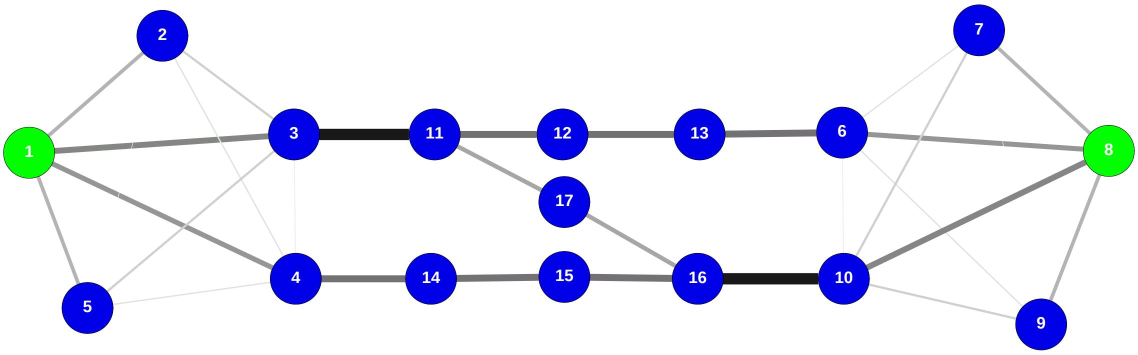

Figure 5: Relative edge relevance based on random walks between nodes 1 and 8.

The proposed approach relies on an interpretation of the graph as a Markov chain. The states of such a model correspond to the nodes in the original graph and the transition probabilities are defined proportionally to the edge weights. Markov chain theory [Kemeny, 83, Norris, 97] allows one to compute the nodes and edges most frequently visited while performing random walks between any pair of distinct nodes of interest. The walks are random but the probability distribution over all possible walks is generally far from uniform. Hence the likelihood of any given walk actually matters in the relevance computation. These notions are formally presented in section 2 detailing the proposed *k-walk approach*.

The edge or node relevance measures used here are *betweenness centrality indices* [Brandes, 05]. Shortest-path betweenness has been proposed in [Freeman, 77]. Random walk betweenness [Newman, 05] was introduced more recently. Our approach can be considered as offering several extensions to Newman's work. In particular, we do not restrict our attention to undirected unweighted graphs. More importantly, edge and node relevances are here evaluated *relatively* to the choice of k (≥ 2) nodes of interest rather than defined as a fixed characterization of a given graph. For instance, Figure 6 depicts the edge relevances of the graph of Figure 2 when nodes 3, 7 and 10 are selected as nodes of interest. The result is clearly distinct from the graph depicted on Figure 3.

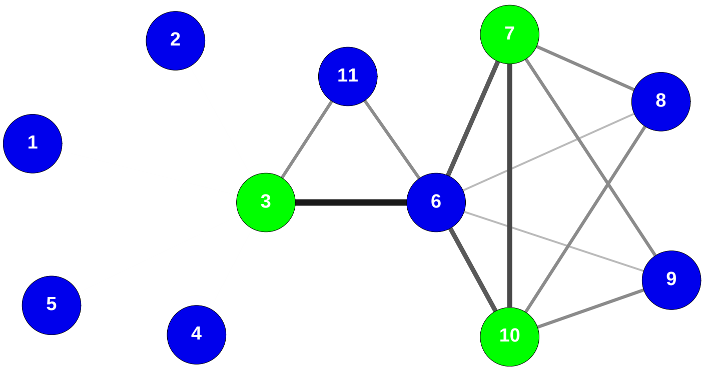

Figure 6: Edge relevances relatively to the nodes 3, 7 and 10.

Moreover, as pointed by Newman, shortest-path betweenness and random walk betweenness can be considered at the opposite end of a spectrum of possibilities in terms of the number of walk steps. Indeed a random walker has no idea of where she is heading to while a shortest-path walker intends to follow a straightest path from source to destination[6]. Section 3 presents the *limited k-walk approach*, which can be considered as an intermediate along this spectrum. Here the walker travels also at random but only those walks of a prescribed length L are considered. A slightly distinct variant considers all random walks up to a maximal length L_{max}. When L_{max} tends to infinity this methods becomes equivalent to the original k-walk approach.

[6] Whenever unit weights are assigned to the edges, shortest distance paths are also minimal in number of steps.

Both algorithms are however distinct from a computational viewpoint as discussed in section 3.

Section 4 describes another extension where the nodes of interest are no longer considered individually but as groups of *a priori* related nodes.

We discussed so far how to define a relevance measure on the edges and nodes of a graph to best explain the relationships between k nodes of interest. Keeping only those edges the relevance of which is above a prescribed threshold is a direct way to extract a non-necessarily connected subgraph. Section 5 elaborates on the subgraph extraction itself.

A relevant subgraph should be, in many cases, as small as possible while capturing most of the information to explain the relationships between the nodes of interest. In other words, one would like to discriminate between important and less important edges (or nodes). The k-walk or limited k-walk approaches are not necessarily highly discriminant since their relevance measures are based on a large, possibly infinite, set of walks in the graph. Many walks in this set can largely overlap while partially sharing the captured information. Whether or not the extracted subgraph is small relative to the initial graph also depends on the graph topology, the initial edge weights and the chosen nodes of interest. In any case, one can enforce more discrimination by *inflating* the differences between the edges as detailed in section 6.

An inspiring approach to the problem of extracting a relevant subgraph is described in [Faloutsos, 04]. To the best of our knowledge, this is the only previous work where the problem of extracting a relevant subgraph from a given set of nodes of interest has been stated. The problem was however restricted to 2 nodes of interest in an undirected graph. In that approach, the 2 nodes of interest are respectively considered to be the source and the sink of an electrical current. The algorithm searches the paths followed by the current flow and maximizes the sum of current flow in the extracted subgraph. In addition, each node includes some current loss in order to penalize long paths and very highly connected nodes (hubs). A comparison with this approach is further discussed in section 7 where we present an electrical equivalent of the k-walk approach for undirected graphs.

Practical evaluations of the k-walk and limited k-walk approaches are detailed in section 8. We conclude our discussion and present the perspectives of this work in section 9.

2 K-walks in a graph

Let $G = (V, E)$ denote a graph formed by a set V of vertices (or nodes) and a set E of edges, with $E \subseteq V \times V$. Let $n = |V|$ denotes the graph order. We consider in particular connected graphs represented by their weighted $n \times n$ adjacency matrix $\mathbf{A}$. The a_{ij} entry of $\mathbf{A}$ denotes the weight of the edge connecting node i to node j. Weights are assumed to be zero if and only if their corresponding edge does not belong to the graph. Otherwise, the weight between any pair of connected nodes is assumed strictly positive. Moreover edge weights should be defined such that the larger the weight a_{ij} the easier the connection (or communication or information flow) from i to j. We consider both directed and undirected graphs. An undirected graph is simply represented by a *symmetric* matrix $\mathbf{A}$. In the directed case, the graph is assumed weakly connected and there must exist a directed path from any node to at least one node of interest.

The diagonal degree matrix is defined as $\mathbf{D} = \text{diag}(d_1, \ldots, d_n)$ with $d_i = \sum_{j=1}^{n} a_{ij}$. Whenever $\mathbf{A}$ reduces to a binary adjacency matrix (weights are either 0 or 1), d_i simply denotes the degree[7] of node i. In general, d_i is interpreted as the weighted degree of node i. A related quantity is the *graph volume* $D_G = \sum_{i=1}^{n} d_i$.

Given a weighted adjacency matrix $\mathbf{A}$ associated to a graph $G = (V, E)$ and a subset $K \subseteq V$ ($|K| \geq 2$) of nodes of interest, we define relevance indices on the edges or nodes of G. They measure how much each node or edge contributes to the relationships between the nodes of K. The number $k = |K|$ of nodes of interest is typically much smaller than n. Formally, we define a *node relevance* function $nr_{\mathbf{A},K} : V \to \mathbb{R}^+$ which maps any node to its relevance. We define an *edge relevance* function $er_{\mathbf{A},K} : E \to \mathbb{R}^+$, similarly.

As motivated in section 1, the relevances should rely on all possible ways to connect the k nodes of interest (each way having a certain likelihood) and not only shortest distance or maximal flow paths. Technically, we propose to define the relevance of a node or an edge to be proportional to the *expected number of times* it is used when randomly walking through the graph, starting from one node of interest and eventually reaching a distinct node of interest. The theory of *absorbing Markov chains* allows us to compute these quantities efficiently [Kemeny, 83].

A random walk in a graph can be modeled by a Markov chain describing the sequence of nodes visited during the walk. A state of the Markov chain is associated with each node of the graph. Hence the terms *nodes* and *states* are used interchangeably in the rest of this paper. A random variable $X(t)$ represents the current state of the Markov chain at time t. The probability of transiting to state j at time $t+1$, given that the current state is i at time t, is given by:

$$P[X(t+1) = j | X(t) = i] = p_{ij} = \frac{a_{ij}}{d_i}. \tag{1}$$

Thus, from any state i, the (stationary) probability to jump to state j is proportional to the weight a_{ij} of the edge from i to j. The transition matrix $\mathbf{P} = [p_{ij}]$ of the Markov chain is related to the degree and adjacency matrices as $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$. Note that even when $\mathbf{A}$ is symmetric, $\mathbf{P}$ is generally asymmetric since the degrees of adjacent nodes need not be equal. $\mathbf{P}$ is a *row-stochastic* matrix since, by construction, $0 \leq p_{ij} \leq 1$ and $\sum_{j=1}^{n} p_{ij} = 1$.

We restrict first our attention to random walks starting from a given

[7] d_i denotes the *outgoing* degree of node i whenever the graph is directed.

node x of interest and ending in any other nodes of interest $K \setminus \{x\}$. Equation (6) below defines the same computations weighted over all possible starting nodes.

A state of a Markov chain is *absorbing* if and only if any walk reaching this state will stay forever on this state with probability 1. Let ${}^{x}\mathbf{P}$ denote a modified transition matrix for which all nodes of interest but x have been transformed to be absorbing. It is defined from $\mathbf{P}$ as follows.

$$ {}^{x}\mathbf{P}_{ij} = \begin{cases} 1 & \text{if } i \in K \setminus \{x\} \text{ and } i = j, \\ 0 & \text{if } i \in K \setminus \{x\} \text{ and } i \neq j, \\ \mathbf{P}_{ij} & \text{otherwise.} \end{cases} \tag{2} $$

As the original graph is assumed to be connected[8], there is no absorbing state in the Markov chain defined by the original transition matrix $\mathbf{P}$. Hence, only the states of interest but x are absorbing according to ${}^{x}\mathbf{P}$. The other states, including x itself, forms the set V_T of *transient* states from which there is a strictly positive probability to leave. Without loss of generality, the states can be reordered such that ${}^{x}\mathbf{P}$ has the following canonical block structure.

$$ {}^{x}\mathbf{P} = \begin{bmatrix} {}^{x}\mathbf{Q} & {}^{x}\mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{3} $$

Here ${}^{x}\mathbf{Q}$ denotes the $(n-k+1) \times (n-k+1)$ transition submatrix between transient states, $\mathbf{I}$ is the identity matrix of order $k-1$ and ${}^{x}\mathbf{R}$ is a $(n-k+1) \times (k-1)$ matrix. In particular, ${}^{x}\mathbf{R}_{ir}$ denotes the probability of a walk being in a transient state i to be absorbed in state r in one step.

In an absorbing Markov chain, the *node passage time* $n(x,i)$ is defined as the number of times a random walk starting in x goes through state i before getting absorbed. This quantity only depends on the transition matrix ${}^{x}\mathbf{Q}$ between transient states[9]. Indeed, since one is not interested in which state r the walk is absorbed, only the sum of the probabilities of absorption ${}^{x}\mathbf{R}_{i} = \sum_{r} {}^{x}\mathbf{R}_{ir}$ matters to define the total probability of absorption from a given state i. Moreover, since ${}^{x}\mathbf{P}$ is row-stochastic, this sum is equal to $1 - \sum_{j} {}^{x}\mathbf{Q}_{ij} = 1 - {}^{x}\mathbf{Q}_{i\cdot}$. The probability of transiting from state x to state i in a single step is given by ${}^{x}\mathbf{Q}_{xi}$. Let $({}^{x}\mathbf{Q})^{l}$ denote the l^{th} power $(l \geq 0)$ of the matrix ${}^{x}\mathbf{Q}$. The quantity $[({}^{x}\mathbf{Q})^{l}]_{xi}$, which denotes the xi entry of the matrix $({}^{x}\mathbf{Q})^{l}$, defines the probability of transiting from state x to state i in l steps.

[8] In the directed case, the graph is assumed weakly connected. Moreover, there must exist a path from any node to at least one node of interest. If, for a particular node of interest x, there exist only directed paths to itself but not to any other node of interest, we do not consider the transformed MC ${}^{x}\mathbf{P}$ and simply iterate the construction over the other nodes of interest.

[9] The discussion in this paragraph is valid for any absorbing Markov chain with a transition matrix $\mathbf{Q}$ between transient states. We use here the notation ${}^{x}\mathbf{Q}$ to refer explicitly to the Markov chain defined in equations (2) and (3).

Let $\{X_l = i | X_0 = x\}$ denote the random event of visiting state i in the l^{th} step of a random walk starting in x. The expectation of $n(x,i)$ is the expected number of times such events occur, in other words their respective probabilities, for any time index l:

$$E[n(x,i)] = \sum_{l=0}^{\infty} P[X_l = i | X_0 = x] = \sum_{l=0}^{\infty} \left[({}^x\mathbf{Q})^l\right]_{xi} = \left[\mathbf{I} - {}^x\mathbf{Q}\right]_{xi}^{-1} \tag{4}$$

The infinite sum of the successive powers of the matrix ${}^x\mathbf{Q}$ in equation (4) is known as the Neumann series and the last equality follows from the theory of stochastic matrices [Meyer, 00]. The matrix ${}^x\mathbf{N} = \left[\mathbf{I} - {}^x\mathbf{Q}\right]^{-1}$ is called the *fundamental matrix* of an absorbing Markov chain. In particular, ${}^x\mathbf{N}_{xi}$ is the expected number of times a walk starting in the node of interest x goes through state i before getting absorbed in any node of $K \setminus \{x\}$, that is, before reaching a distinct node of interest. Finally, the expected length n_x of any walk starting in x and ending in another node of interest is also the node passage times summed over all transient states:

$$n_x = \sum_{i \in V_T} {}^x\mathbf{N}_{xi} \tag{5}$$

The previous computations are restricted to a particular node of interest x being used as starting state of the random walks. Considering k nodes of interest as starting nodes reduces to computing the same quantities separately for each possible starting node. Typically, an initial probability distribution, represented by the vector ι, is defined with $\iota_x > 0 \Leftrightarrow x \in K$. This probability distribution allows to encode some *prior* preference between the nodes of interest. By default, all nodes of interest are assumed equally important and $\iota_x = \frac{1}{k}, \forall x \in K$. The node relevances are given globally by the *mean node passage times* $nr : V \longrightarrow \mathbb{R}^+$ defined as follows:

$$\forall i \in V, nr(i) = \begin{cases} \sum_{x \in K} \iota_x \, {}^x\mathbf{N}_{xi} & \text{if } i \in V \setminus K, \\ \iota_i \, {}^i\mathbf{N}_{ii} & \text{otherwise.} \end{cases} \tag{6}$$

The sum in the above formula is replaced by a single term whenever i is a node of interest since it is considered only once as a transient state, precisely when it is used as starting state.

The *edge passage time* $e(x,i,j)$ is defined as the number of times a random walk starting in x is using the transition from i to j. The expectation of this quantity directly follows from the above computations:

$$E[e(x,i,j)] = \begin{cases} {}^x\mathbf{N}_{xi} \, {}^x\mathbf{P}_{ij} & \text{if } i \in V \setminus K, \\ {}^i\mathbf{N}_{ii} \, {}^i\mathbf{P}_{ij} & \text{if } x = i \text{ and } i \in K, \\ 0 & \text{if } x \neq i \text{ and } i \in K. \end{cases} \tag{7}$$

The edge relevances are given by the *mean edge passage times* $er : E \rightarrow \mathbb{R}^{+}$, defined according to the directed or undirected nature of the original graph:

$$\forall (i,j) \in E,$$
$$er(i,j) = \begin{cases} \sum\limits_{x \in K} \iota_x \; E[e(x,i,j)] & \text{if } G \text{ is directed,} \\ \sum\limits_{x \in K} \iota_x \; |E[e(x,i,j)] - E[e(x,j,i)]| & \text{if } G \text{ is undirected.} \end{cases} \tag{8}$$

The absolute difference guarantees that edge relevances are symmetric in the undirected case. It is also motivated by an electrical interpretation detailed in section 7.

To sum up, the application of the k-walk approach to compute node and edge relevances essentially amounts to interpret the graph as a Markov chain. The transition probability matrix $\mathbf{P}$ of this chain can be computed in $\Theta(m)$ time with $m = |E|$. Next, k particular submatrices ${}^{x}\mathbf{Q}$ (one matrix for each node x of interest) need to be considered. For each of them, the fundamental matrix ${}^{x}\mathbf{N}$ is computed. This amounts to invert the matrix $\mathbf{I} - {}^{x}\mathbf{Q}$, which can be done in $\mathcal{O}(n^3)$. This is the core of the computational load of this approach. Node and edge relevances can be derived from the fundamental matrices in $\Theta(m)$ time. Overall, the time complexity of the k-walk approach is $\mathcal{O}(kn^3)$.

3 Limited K-walks in a graph

In the k-walk approach, presented in section 2, any walk connecting the nodes of interest is considered, no matter its length. Alternatively, the relationships between the nodes of interest can be explained by walks of a fixed length L or up to a maximal length L_{max}. This is the purpose of the limited k-walk approach.

Any node of interest is again considered in turn as the unique starting node in a transformed Markov chain characterized by the ${}^{x}\mathbf{P}$ transition matrix. We are now interested in the *conditional* expectations $E[n(x,i)|L]$ and $E[e(x,i,j)|L]$, the expected number of times the node i, respectively the edge (i,j), is visited while starting the walk in x *given* that the walk length is L. Conditional expectations can be computed on the edges as follows:

$$E[e(x,i,j)|L] = \sum_{l=0}^{L-1} \frac{P[X_l = i, X_{l+1} = j, L|X_0 = x]}{P[L|X_0 = x]} \tag{9}$$

In equation (9), X_l is a random variable denoting the state visited at step l of the walk. $P[L|X_0 = x]$ denotes the probability of a walk of length L given that the walk started in state x. Similarly, $P[X_l = i, X_{l+1} = j, L|X_0 = x]$ denotes the joint probability of visiting the edge (i,j), between step l and step $l+1$, and having a total walk length of L, given that the walk started in state x.

These probabilities can be computed from the ${}^x\mathbf{Q}$ and ${}^x\mathbf{R}$ matrices associated to ${}^x\mathbf{P}$ (see the block structure of ${}^x\mathbf{P}$ described in equation (3)). In particular, the probability of a walk of length L starting in x is given by

$$P[L|X_0 = x] = \sum_{r \in K \setminus \{x\}} \left[({}^x\mathbf{Q})^{L-1} \, ({}^x\mathbf{R}) \right]_{xr}, \tag{10}$$

since such a walk transits $L-1$ times through transient states before getting absorbed in any state r in $K \setminus \{x\}$. The probability of visiting edge (i,j) in such a walk, if j is a transient state, is given by

$$P[X_l = i, X_{l+1} = j, L|X_0 = x] = \sum_{r \in K \setminus \{x\}} \left[({}^x\mathbf{Q})^{l} \right]_{xi} [{}^x\mathbf{Q}]_{ij} \left[({}^x\mathbf{Q})^{L-l-2} \, ({}^x\mathbf{R}) \right]_{jr}. \tag{11}$$

Indeed, such a walk transits l times through transient states while reaching the transient state i in the l^{th} step, transits from state i to state j with probability $[{}^x\mathbf{Q}]_{ij}$, transits again $L-l-2$ times through transient states before getting absorbed in any state r in $K \setminus \{x\}$. Whenever the destination state j of the edge (i,j) is absorbing, the probability of visiting this edge is given by

$$P[X_{L-1} = i, X_L = j, L|X_0 = x] = \left[({}^x\mathbf{Q})^{L-1} \right]_{xi} [({}^x\mathbf{R})]_{ij}, \quad \forall j \in K \setminus \{x\} \tag{12}$$

since, for the walk length to be equal to L before getting absorbed, such an edge can only be visited at the last step of the walk.

The *limited mean edge passage times* $E[e(x,i,j) \mid L \leq L_{max}]$ are defined for a maximal walk length L_{max}:

$$E[e(x,i,j) \mid L \leq L_{max}] = \sum_{L=1}^{L_{max}} E[e(x,i,j)|L] \tag{13}$$

Globally, the edge relevances also depend on the definition of an initial probability distribution ι weighting the relative importance of each node of interest ($\iota_x > 0 \Leftrightarrow x \in K$). Edge relevances are computed according to equation (8) by replacing the unconditional expectations $E[e(x,i,j)]$ by conditional expectations. If one is interested only in the relationships between the nodes of interest for a fixed walk length L, the expectations $E[e(x,i,j)|L]$ are considered. For all walks up to length L_{max}, the expectations $E[e(x,i,j) \mid L \leq L_{max}]$ are used instead.

Node relevances are assigned according to the *limited node passage times* computed as the sum of the limited passage times on all outgoing edges from each node:

$$\forall i \in V, nr(i) = \sum_{x \in K} \sum_{L=1}^{L_{max}} \iota_x E[n(x,i)|L], \text{ with } E[n(x,i)|L] = \sum_{j \in V} E[e(x,i,j)|L]. \tag{14}$$

One can also consider limited walks while letting L_{max} tends to infinity. This offers an alternative way to compute the non limited k-walks since

$$E[n(x,i)] = \sum_{L=1}^{\infty} E[n(x,i)|L] \;\; P[L|X_0 = x] \tag{15}$$

Practical computation of the limited k-walks can be performed efficiently with two recurrences similar to those of the forward-backward algorithm used to estimate the parameters of a Hidden Markov Model [Rabiner, 93].

The forward recurrence computes the probability $\alpha(i,l)$ of starting the walk in x and reaching state i in l steps. It uses a left-to-right lattice structure with $|V| = n$ lines and L_{max} columns, each column being associated with a specific time index. The value of the i^{th} line at time l is precisely $\alpha(i,l)$. The transient states are assigned to the first $n-k+1$ lines in this lattice, with $\alpha(i,l) = \left[({}^x\mathbf{Q})^l\right]_{xi}$ in this case. Any such $\alpha(i,l)$ only depends on the previous column since $({}^x\mathbf{Q})^l = ({}^x\mathbf{Q})^{l-1}\,({}^x\mathbf{Q})$. The absorbing states are assigned to the last $k-1$ lines and the corresponding entries are given by $\alpha(i,l) = \left[({}^x\mathbf{Q})^{l-1}\;{}^x\mathbf{R}\right]_{xi}$. These entries also depend only on the previous column. Since the walks must start in state x, the basis of the forward recurrence is $\alpha(x,0) = 1$ and $\alpha(i,0) = 0$ if $i \neq x$.

A second lattice structure of the same size is used to compute a backward recurrence $\beta(i, L-l)$, which represents the probability of getting absorbed in $L-l$ steps, if the process is in state i after l steps. Hence $\beta(i,L-l) = \sum_{r \in K\setminus\{x\}} \left[({}^x\mathbf{Q})^{L-l-1}\,({}^x\mathbf{R})\right]_{ir}$. The recurrence is computed backward in time, from $l = L$ to $l = 0$. Since the walks must end in an absorbing state, the basis of the backward recurrence is $\beta(i,0) = 1$ if $i \in K \setminus \{x\}$, and $\beta(i,0) = 0$ otherwise.

The time complexity of the forward and backward recurrences for one transformed Markov chain is $\Theta(mL_{max})$, with $m = |E|$. Equation (9) can be reformulated using these recurrences:

$$E[e(x,i,j)|L] = \begin{cases} \dfrac{\sum_{l=0}^{L-1} \alpha(i,l)\;[{}^x\mathbf{Q}]_{ij}\;\beta(j,L-l-1)}{\beta(x,L)} & \text{if } j \in \{x\} \cup V \setminus K \\[2ex] \dfrac{\alpha(i,L-1)\;[{}^x\mathbf{R}]_{ij}}{\beta(x,L)} & \text{if } j \in K \setminus \{x\} \end{cases} \tag{16}$$

Equation (16) can be evaluated, from the two lattices, in $\Theta(mL)$ which, when summed over all possible lengths up to L_{max} (equations (13) and (14)), can also be performed globally in $\Theta(mL_{max})$. Finally, as the above computations need to be repeated for k Markov chains, the overall time complexity of the limited k-walk approach is $\Theta(kmL_{max})$. An equivalent upper bound is $\mathcal{O}(kn^2L_{max})$, but this upper bound is tight only if the graph is dense.

4 Groups of nodes of interest

The nodes of interest were considered so far independently of each other. In a more general setting, the elements of K can be partitioned into p clusters: $K = \{C_1, \ldots, C_p\}$ with $2 \leq p \leq k$. The node or edge relevances should now explain the relationships between the clusters rather than between the individual nodes.

If the k-walk approach is run while ignoring the cluster structure, node and edge relevances depend on the relationships between nodes of interest belonging to the same cluster, which is generally inappropriate. An alternative would be to construct a modified graph in which all nodes belonging to the same cluster are merged. This modified graph can however be an overly simplified view of the original graph.

A simple extension of the k-walk approach considers the cluster structure explicitly. A transformed Markov chain is built relatively to each cluster rather than each node of interest. When the cluster C_x is considered, all nodes belonging to $K \setminus C_x$ are transformed to be absorbing. Hence the walks start in any node of C_x and ends in any node of $K \setminus C_x$. Let ${}^{C_x}\mathbf{N}$ denote the fundamental matrix of this transformed Markov chain. The mean node passage times are now defined as:

$$\forall i \in V, nr(i) = \begin{cases} \sum_{x \in K} \iota_x \, {}^{C_x}\mathbf{N}_{xi} & \text{if } i \in V \setminus K, \\ \iota_i \, {}^{C_i}\mathbf{N}_{ii} & \text{otherwise.} \end{cases} \tag{17}$$

The expected edge passage times are defined analogously (see equation (7)). A similar reasoning can also apply to the limited k-walk approach.

5 Subgraph extraction

The k-walk and limited k-walk approaches essentially amount to define edge (or node) relevances proportionally to their frequencies of use along walks connecting the nodes of interest. A *relevant subgraph* can subsequently be extracted in several ways.

The simplest approach requires a positive *edge relevance threshold* θ_e and includes the edge (i, j) in the extracted subgraph whenever $er(i, j) > \theta_e$. This procedure defines a subgraph induced by the selected edges, the larger θ_e the smaller the induced subgraph.

A positive *node relevance threshold* θ_n can also be defined to include the node i whenever $nr(i) > \theta_n$. The use of edge and node thresholds gives more flexibility. In particular, a specific node may be selected even though none of its incident edges are selected. Whether such a situation is desirable depends on the application context.

In the k-walk approach, the extracted subgraph need not be (weakly) connected even when thresholding is applied only on the edges. Whenever connectedness is required, one can search for the maximal θ_e such that the induced subgraph is connected. Since the original graph is assumed connected, there must exist a critical threshold θ_e^* such that the extracted subgraph is connected for any $\theta_e \leq \theta_e^*$. Fixing automatically θ_e to this critical value defines a parameter-free subgraph extraction approach.

The above approach is a greedy edge selection procedure which is efficient and locally minimal, in terms of the number of edges of the induced subgraph. In the undirected case, one could also define edge costs inversely proportional to their respective relevances and look for the subgraph connecting the nodes of interest with a minimal total edge cost. This is a particular instance of the *Steiner tree problem* for undirected weighted graphs [Hwang, 92]. The optimal subgraph is indeed necessarily a tree in this case. This problem was shown NP-complete [Garey, 79] but exact algorithms exist in restricted cases [Warme, 00].

The limited k-walk approach offers another alternative to construct a connected subgraph. Only the subset of walks of a given length L, or up to a maximal length L_{max}, are considered. In any case, a given edge relevance is strictly positive if and only if it is used in at least one walk of this subset. One can thus simply discard all edges with a null relevance. The resulting subgraph, if not empty, is connected by construction. It is also the smallest subgraph representing all walks of the prescribed length(s) between the nodes of interest.

So far, the (limited) k-walk approaches require the original graph to be connected. In the directed case, the graph is assumed weakly connected and there must exist a directed path from any node to at least one node of interest. A possible extension would consider separately each connected component of the original directed or undirected graph and restrict the analysis in each component to the nodes of interest belonging to it.

6 Inflation

The k-walk approach offers a global view of the ways nodes of interest are connected between each other. It is however not necessarily very discriminative between highly relevant edges (or nodes) versus less relevant ones. The relevance measures are indeed based on a large, possibly infinite, set of walks. Many of these walks overlap at least partially and the captured information is spread between them. This may be not optimal if the goal is to extract a small subgraph summarizing most of this information.

To some extent the limited k-walk approach is more discriminative, especially for low L_{max}. In particular when $k = 2$, if L_{max} is fixed to the minimal number of steps between the 2 nodes of interest, only nodes or edges belonging to shortest paths, in terms of number of steps, have a strictly positive relevance. The larger L_{max} the more walks are considered with a less discriminant result. Tuning L_{max} is thus a way to control discrimination.

In general, the discriminative character of the k-walk approach (or the limited k-walk approach for a large L_{max}) depends on the edge weights in the original graph, the chosen nodes of interest and the graph topology. If necessary, it is easy to inflate the differences between edge (or node) relevances by applying the k-walk approach recursively. This notion of inflation is inspired from [vD, 00] where it used for defining graph clusters. The very simple way inflation is implemented in our approach is however different.

Figure 7 illustrates this idea. In the original graph on top, all edge weights are assumed equal to 1. The graph obtained after one execution of the k-walk approach is depicted in the middle. Here, the edge widths are their relative edge relevance. Edge relevances can be considered as new weights associated to edges and the same approach can be applied recursively. The result obtained after the second iteration is depicted at the bottom of Figure 7.

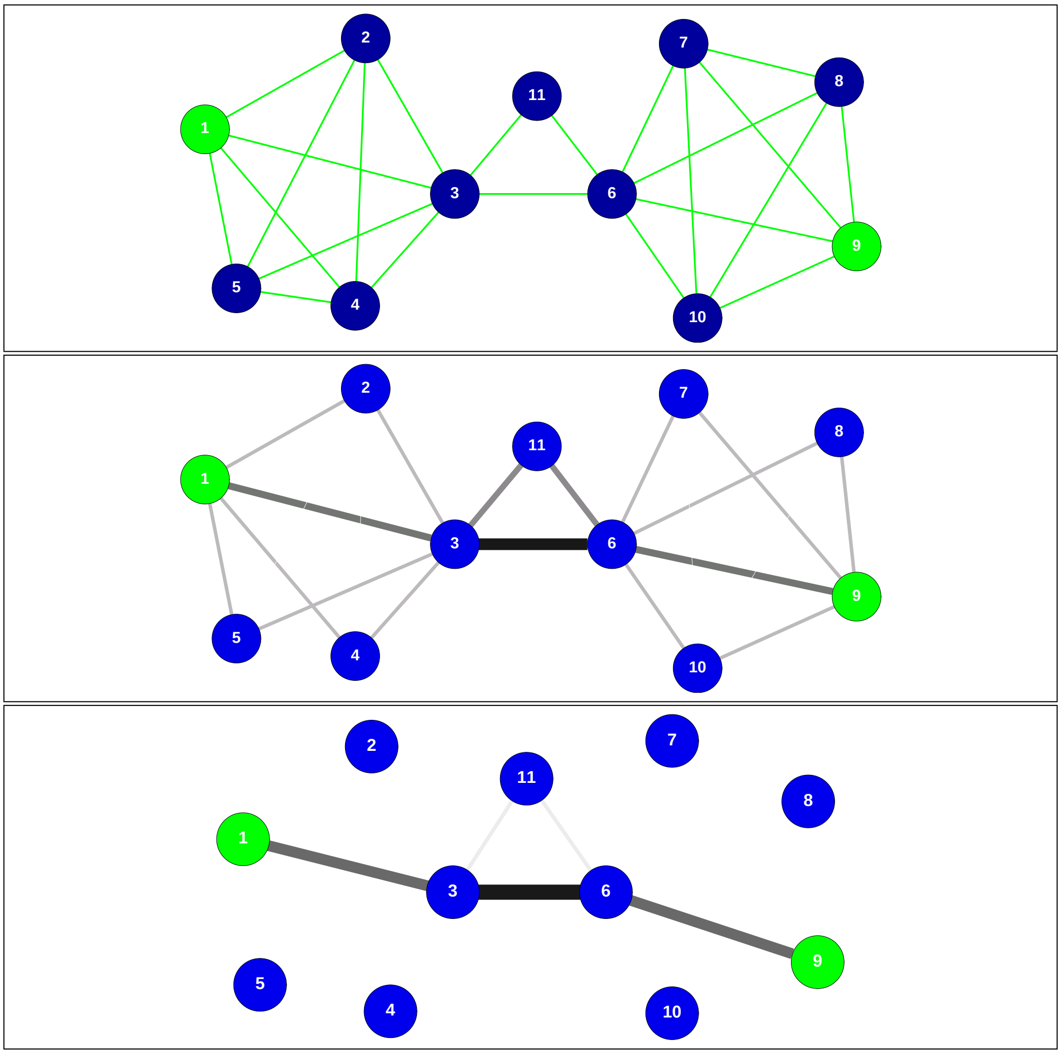

Figure 7: Inflating the relative edge relevances.

7 An electrical interpretation

An undirected graph is represented by a symmetric weighted adjacency matrix $\mathbf{A}$. In this case, the graph can be seen as an electrical network, the weight a_{ij} being the conductance between nodes i and j [Doyle, 84]. According to equation (1), this symmetry implies a relationship between the transition probabilities (in the associated Markov chain) assigned to each direction for traversing an edge:

$$d_i \mathbf{P}_{ij} = d_j \mathbf{P}_{ji}. \tag{18}$$

The degree d_i is now interpreted as the sum of the conductances of the edges incident to node i.

When a node of interest x is used as starting state of the walks, the network is represented by a transformed Markov chain with transition matrix ${}^x\mathbf{P}$. Positively charged particles are assumed to enter the network at node x and to leave the network at any node in $K \setminus \{x\}$. The node x is thus assumed to be the source of an electrical current and the other nodes of interest form current sinks. It could be even more realistic to consider negatively charged particles flowing in the opposite direction towards x but the reasoning below is essentially equivalent. The currents observed between any nodes i and j are directly related to the edge passage times in the k-walk approach as described below.

By Ohm's Law, the current I_{ij} from i to j is determined by the voltages V_i and V_j and the conductance between i and j:

$$I_{ij} = (V_i - V_j) a_{ij} \tag{19}$$

Kirchoff's Current Law requires that the total current flowing through any node (but the current source and sinks) is 0:

$$\forall i, j \in V \setminus K, \ \sum_j I_{ij} = 0. \tag{20}$$

The relationships between node voltages follows from the combination of equations (19) and (20):

$$\forall i, j \in V \setminus K, \ \ V_i = \sum_j V_j \, {}^x\mathbf{P}_{ij} \tag{21}$$

In the k-walk approach, ${}^x\mathbf{N}_{xi}$ is the expected number of times node i is visited for a walk starting in x. This quantity can also be defined from the expected number of times any node j is visited:

$${}^x\mathbf{N}_{xi} = \sum_j {}^x\mathbf{N}_{xj} \, {}^x\mathbf{P}_{ji} \tag{22}$$

The relationship between these expected times and the voltages follows from the combination of equations (18) and (22):

$$\forall i, j \in V \setminus K, \ \ \frac{{}^x\mathbf{N}_{xi}}{d_i} = \sum_j \frac{{}^x\mathbf{N}_{xj}}{d_j} \, {}^x\mathbf{P}_{ij} \tag{23}$$

Equations (21) and (23) are exactly equivalent when $V_i = \frac{{}^x\mathbf{N}_{xi}}{d_i}$ showing that the voltages in this electrical network correspond to the node passage times normalized by the node degrees. Moreover, by equation (19), the current from i to j is given by

$$I_{ij} = (\frac{{}^x\mathbf{N}_{xi}}{d_i} - \frac{{}^x\mathbf{N}_{xj}}{d_j})a_{ij} = {}^x\mathbf{N}_{xi}\ {}^x\mathbf{P}_{ij} - {}^x\mathbf{N}_{xj}\ {}^x\mathbf{P}_{ji} \tag{24}$$

The current I_{ij} is the current along the edge (i, j) actually flowing from i to j whenever it is positive, or flowing in the opposite direction otherwise. Hence the net current along the edge (i, j), independently of its direction, is the absolute value $|I_{ij}|$. This absolute value is exactly the net edge passage times considered in equation (8) in the case of an undirected graph. The edge relevances in the k-walk approach correspond to the sum of the net currents obtained for k electrical networks when each node x of interest is used in turn as a current source of ι_x current unit.

The electrical interpretation of the k-walk method illustrate some links and differences with the method proposed in [Faloutsos, 04]. The latter is restricted to 2 nodes of interest, respectively called the source and the destination, in an undirected graph. A one unit voltage is applied to the source and the destination is grounded. The voltages in all other nodes are computed by solving the system of linear equations (21). The electrical analogy is slightly different in the k-walk approach since a unit current is injected in the source node rather than fixing its voltage. The respective currents and voltages are however identical in both electrical circuits up to a multiplicative constant equal to the effective resistance of the network [Doyle, 84]. In other words, the expected edge passage times are identical in relative terms, for one starting node. The k-walk approach generalizes this methodology to any number of nodes of interest by considering k electrical networks.

Both approaches also differ when considering the actual subgraph extraction. The algorithm proposed by Faloutsos et al. searches the paths followed by the current flow, from source to destination, and maximizes the sum of current flow in the extracted subgraph. This approach is based on the extraction of successive paths using dynamic programming. Such an approach is more difficult to extend to any number of nodes of interest since the paths to consider should be defined according to the k nodes.

In contrast, the k-walk approach does not compute explicitly some paths but defines edge relevance according to their use in all possible walks. Walks, as opposed to paths, also include possible round trips along a given edge but the edge relevances are defined, for an undirected graph, as the net differences in both directions. Note that if the graph is directed or, more precisely, if $\mathbf{A}$ is no longer symmetric, the electrical analogy presented here does not hold anymore since physical laws require the conductance to be symmetric. The k-walk approach can still be applied however.

A universal sink node, which absorbs a fraction of the current delivered to each node, is also introduced in [Faloutsos, 04]. The purpose of this current loss is to penalize very long paths between the two nodes of interest and paths going through highly connected nodes (hubs). The (limited) k-walk approaches do not include such current losses. Discarding very long walks is the purpose of the limited k-walks approach. We argue that this is a more direct and principled way of limiting the maximal walk length considered between the nodes of interest. Discarding hubs can be obtained by defining the weighted adjacency matrix in such a way that any edge connected to a hub has a low conductance.

Finally, solving the system of linear equations (21) can be done in $\mathcal{O}(n^3)$ but iterative methods can often perform faster for sparse graphs. As detailed in section 3, the limited k-walk approach can be seen as an approximation to the k-walk approach for a large L_{max}. The resulting time complexity reduces to $\Theta(mL_{max})$ for a single source node. The possible sparsity of the graph directly pays off here since this complexity is linear with m, the number of graph edges.

8 Experiments

An experimental study of the (limited) k-walk approaches is conducted in the present section. The objectives of this study are summarized hereafter:

1. evaluate the discrimination efficiency of the proposed methods, that is, their ability to extract small relevant subgraphs,

2. observe the benefits of inflation (see section 6) with respect to the discrimination efficiency,

3. measure practical run times and validate them against the theoretical complexity,

4. evaluate how well the limited k-walk approach can approximate the k-walk approach for increasing L_{max} values.

Randomly generated graphs are considered as well as two strongly connected components of the metabolic network representing the KEGG LIGAND database [Goto, 00]. Random graphs were generated using the algorithm presented in [Viger, 05]. This technique allows one to generate an undirected and unweighted graph with a prescribed degree sequence drawn from a power law[10]. In our experiments, the exponent of the power law γ is set to 2.5 and the μ parameter is tuned such that the mean degree equals 4, which are typical parameters used in [Viger, 05]. Graphs were generated with sizes ranging from 100 to 20,000 nodes. The two strongly connected components extracted from KEGG contain respectively 7,695 and 18,297 nodes and their respective mean degrees are 2.7 and 2.9. Randomly generated graphs

[10] The probability that a node has a degree equal to d is $P[d] = (d + \mu)^{-\gamma}$.

are undirected while the KEGG network forms a directed graph. Furthermore, we have defined weights on the graph edges such that $a_{ij} = \frac{2}{d_i+d_j}$ where d_i and d_j are respectively the out degrees of nodes i and j.

Each experiment is carried out using random sets of nodes of interest of size $|K| = 2, 5, 10$ and 20. For each size, 10 sets are sampled out in order to produce averaged results with standard deviations.

8.1 Discrimination efficiency

The (limited) k-walk approaches outputs relevance weights on the edges and nodes of the input graph. The total amount of information is defined as the sum of these relevances over all edges. Section 5 proposes several techniques to extract a relevant subgraph. In our experiments, a subgraph is extracted by keeping every edges with a relevance above a given threshold θ_e. The information captured by a subgraph is simply defined as the sum of the relevances over all edges in the subgraph. Dividing this quantity by the total information provides the *relative* captured information. Similarly, the subgraph relative size is obtained by dividing the number of edges in the subgraph by the number of edges in the input graph. Figure 8 shows the relative captured information for increasing relative subgraph sizes using the k-walk method.

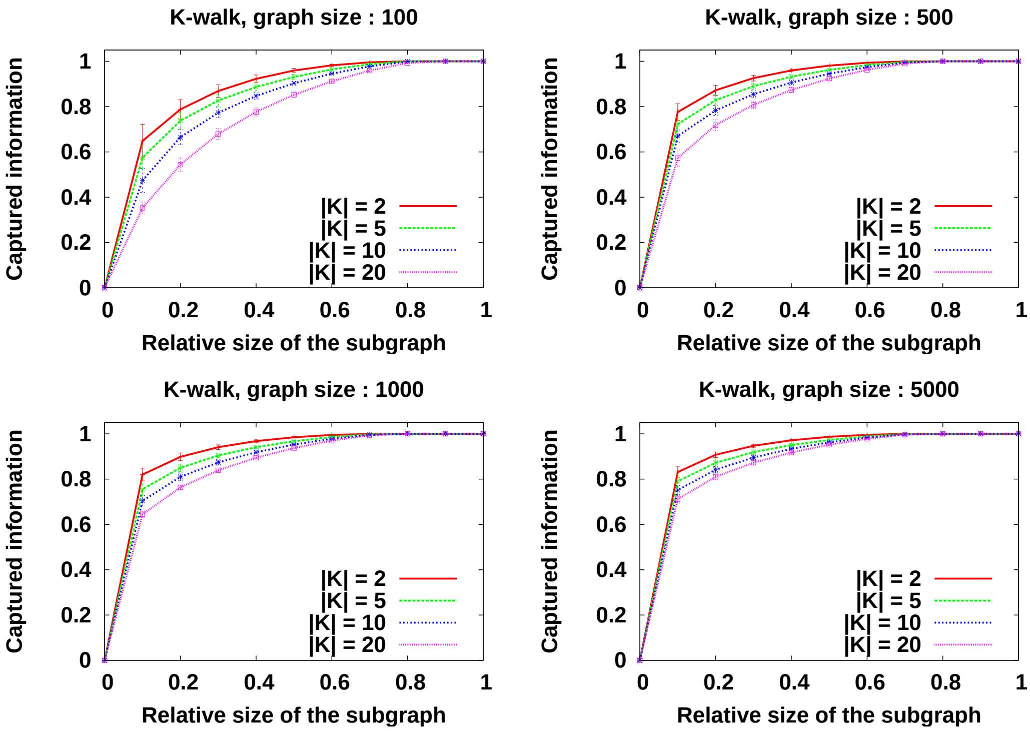

Figure 8: The relative captured information averaged over 10 random sets of nodes of interest for increasing relative subgraph sizes using the k-walk approach.

The k-walk technique seems to offer a good discrimination efficiency since a large amount of information can be captured with small subgraphs. This is especially true when a small set of nodes of interest is considered. For instance, for an input graph with 1,000 nodes and $|K| = 2$, a 10% subgraph captures in average 82% of information. In contrast, when $|K| = 20$, it only captures 64% of information. The rationale is that a larger set of (randomly chosen) nodes of interest is more likely to cover many regions of the input graph, spreading the information in the graph. Furthermore, considering larger input graphs also reduces the covering of the nodes of interest, making the approach more discriminative.

We consider now the limited k-walk approach. Figure 9 shows the relative discrimination efficiency for several L_{max} values using an input graph with 1,000 nodes. This technique is more discriminative for small L_{max} values. Indeed short walks only explore frequently a small portion of the input graphs, while concentrating the captured information. Note that using $L_{max} = 50$ provides almost the same results as the standard k-walk approach (see the bottom left of Figure 8). Using L_{max} greater than 50 does not modify the discrimination efficiency. As for the k-walk approach, a better discrimination is obtained with a smaller number of nodes of interest.

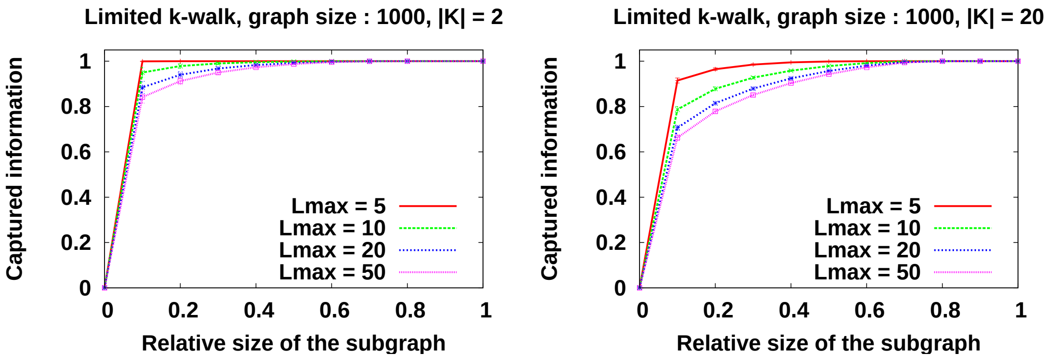

Figure 9: The relative captured information averaged over 10 samples of seed nodes for increasing L_{max} values using the limited k-walk approach.

We consider next the KEGG directed network. The plot of captured information against extracted subgraph relative size using the k-walk approach is presented on the left side of Figure 10. Relatively small portions of the input graph already contain a majority of the information of the graph. For instance, 10% of the input graph captures 67% of the total information. Note also that the captured information does not vary with the number of nodes of interest. We interpret this result as a consequence of the strong connectedness of the component extracted from the whole network. The information rapidly diffuses throughout such a graph and limited walks should reveal much better the influence of the specific nodes of interest chosen. The right side of Figure 10 illustrates that the captured information increases significantly faster with $L_{max} = 10$, with a slight dependence on $|K|$. The behavior of the limited approach tends rapidly to the unlimited case for larger L_{max} values. The same phenomenon is observed for the KEGG strongly connected component containing 18,297 nodes.

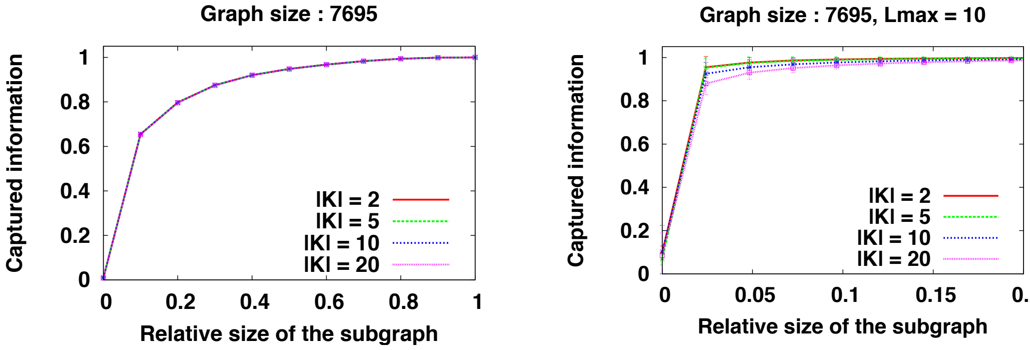

Figure 10: The relative captured information in a strongly connected component of the KEGG network containing 7,695 nodes. Left: The relative captured information using the k-walk approach. Right: The relative captured information using the limited k-walk approach with $L_{max} = 10$.

8.2 Impact of inflation

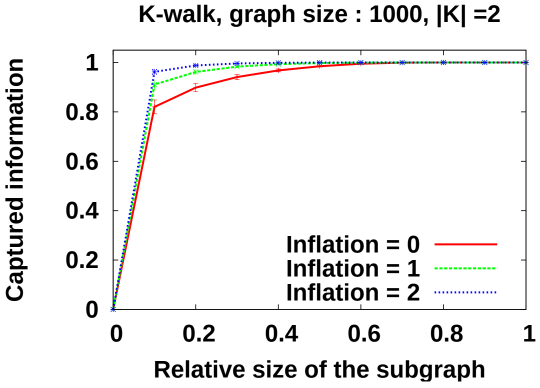

Figure 11: The influence of inflation on the discriminative efficiency using the k-walk approach. Curves are provided using no inflation and using 1 and 2 inflation iterations.

Figure 11 shows the influence of inflation on the k-walk approach using a random graph of size 1,000 and two nodes of interest. As expected, inflation allows one to improve the discriminative efficiency. For instance, a 10% subgraph captures in average 82% of information. Inflating the k-walk approach once or twice captures respectively 91% and 96% of information with the same subgraph size. Similar observations are made when studying the impact of inflation on the limited k-walk approach.

8.3 CPU time

The time complexity of the k-walk approach is $O(|K|n^3)$ (see section 2). We have assessed that our implementation fits this theoretical complexity. The left side of Figure 12 presents running times per node of interest for growing random graph sizes. The plotted times are computed by dividing the CPU time by the number of nodes of interest. Since the expected complexity is linear in $|K|$, these normalized times should be cubic in n. The plot presented on the left side of Figure 12 is a cubic curve fitting very well our experimental measures.

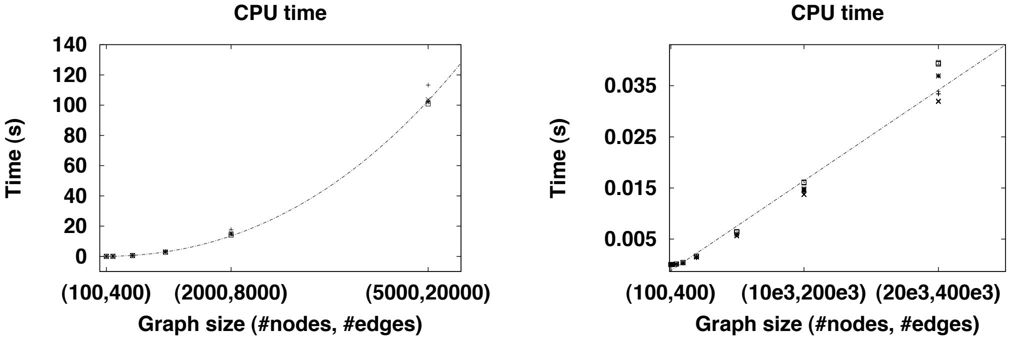

Figure 12: Left: running time results for the k-walk approach. The running time divided by $|K|$ is presented for various values of $|K|$ and for increasing graph sizes. Right: Running time results for the limited k-walk approach. The running time divided by $|K|L_{max}$ is presented for various values of $|K|$ and L_{max} for increasing graph sizes.

The time complexity of the limited k-walk approach is $\Theta(|K|mL_{max})$. The right side of Figure 12 presents the CPU time normalized per walk step and number of nodes of interest. The right side of Figure 12 presents a least-square fit of a linear function to our experimental data.

In a nutshell, the limited k-walk approach can handle an input graph of 100,000 nodes, with 10 nodes of interest and $L_{max} = 1,000$ in about 30 minutes on a standard PC. The actual useful maximal length to consider is often significantly smaller as discussed in the next section.

8.4 Approximating k-walks with limited k-walks

As mentioned in the previous paragraph, the limited k-walk offers a significant reduction of CPU time as compared to the k-walk approach. Such a reduction allows one to handle much larger graphs. We study now whether the limited k-walk approach can approximate well the subgraph extracted by the k-walk approach. The limited approach is guaranteed to become equivalent to the unlimited case when L_{max} tends to infinity (see section 3). We shall see that a good approximation is already obtained in practice with relatively small L_{max} values.

A first indicator of the quality of the approximation is the cumulated absorption probability:

$$\frac{1}{|K|} \sum_{x \in K} \sum_{L=1}^{L_{max}} P[L|X_0 = x]$$

For a given starting node x, the inner sum computes the cumulative probability of walks up to length L_{max}. When all walk lengths are considered (i.e. $L_{max} \rightarrow \infty$), this quantity sums up to one. This cumulative probability is averaged over all starting nodes in K. Hence, if this average is close to one for a finite L_{max} value, the limited k-walk is expected to approximate well the k-walk approach.

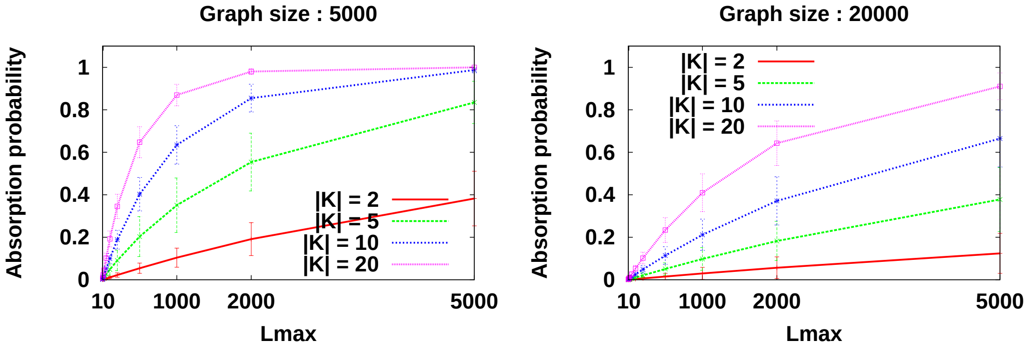

Figure 13: The cumulated absorption probability for increasing L_{max} values, using the limited k-walk approach.

Figure 13 displays the cumulated absorption probability with respect to the L_{max} value for two input graphs containing respectively 5,000 and 20,000 nodes. A higher mass of absorption probability is obtained with a larger number of nodes of interest. For instance, for an input graph containing 20,000 nodes, an absorption probability of 0.91 is obtained with $L_{max} = 5,000$ and 20 nodes of interest while this probability mass is only 0.12 for 2 nodes of interest. The rationale is that the distance, in terms of number of steps, between two distinct nodes of interest becomes smaller in average with a larger number of nodes of interest. Therefore, smaller walk lengths are required to get a high absorption probability mass.

The above analysis shows that only a small fraction of the total probability mass may be captured even for relatively large L_{max} values. The extracted subgraph is however not necessarily very different with respect to the unlimited case. Indeed, *relative* edge relevances may be already well approximated with a small L_{max}. The following experiment confirms this statement.

Several subgraphs are extracted with the k-walk approach. Each subgraph corresponds to a fixed proportion (5%, 10% and 20%) of the same input graph. They serve as reference subgraphs. Next, the limited k-walk approach is used to extract subgraphs. Figure 14 reports precision/recall figures of the edges included in these subgraphs with respect to the references. The curves are obtained while varying the selection threshold of the edge relevances computed with the limited approach. The plots correspond to several L_{max} values with an input graph of 5,000 nodes and 5 nodes of interest.

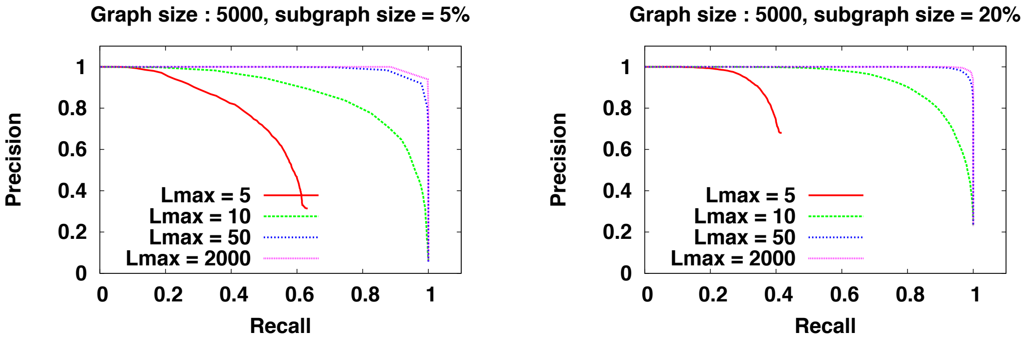

Figure 14: Measure of the quality of the approximation of the k-walk approach by the limited k-walk approach by comparing the extracted subgraphs. Subgraphs of relative sizes 5% (left) and 20% (right) are considered for the k-walk approach. Precision/recall curves are computed by increasing the threshold on the edge relevance weight obtained with the limited k-walk.

Unsurprisingly, the higher the L_{max} value, the better the precision/recall. A very good precision/recall curve is already obtained using $L_{max} = 50$. This curve is almost ideal while the cumulated absorption probability is only equal to 0.2 in this case. In conclusion, the standard k-walk approach can be approximated accurately with the limited k-walk approach using a small L_{max} value and at a much lower computational cost.

9 Conclusion and research perspectives

This work describes novel methods to extract a relevant subgraph that best captures the relationships between k given nodes of interest in a connected graph. The number k of nodes of interest is typically assumed much smaller than the graph order n. The proposed approaches are based on random walks between the nodes of interest to define node and edge relevances proportionally to their expected frequency of use along these walks. Such quantities can be computed efficiently after transforming the graph successively into k

absorbing Markov chains. In particular, the time complexity of the k-walk approach is $\mathcal{O}(kn^3)$ since it essentially amounts to invert k matrices of size $\mathcal{O}(n) \times \mathcal{O}(n)$.

The k-walk approach relies on walks of any length between the nodes of interest. In contrast, a method relying only on shortest paths between these nodes would offer a much more restrictive view. The limited k-walk approach offers an intermediate view (in number of walk steps) since it considers random walks up to a maximal length L_{max}. It can be efficiently implemented with two recurrences similar to those of the forward-backward algorithm used for estimating the parameters of a Hidden Markov Model [Rabiner, 93]. The resulting time complexity is $\Theta(kmL_{max})$, where m denotes the number of graph edges.

The limited k-walk method offers an efficient way to approximate unlimited k-walks even when a relatively small L_{max} is chosen. The quality of this approximation is discussed in section 8. Practical experiments also illustrate that the proposed methods are well suited to extract relatively small subgraphs while capturing most of the information between the nodes of interest. Even more discriminative results can be obtained while inflating edge relevances.

The limited k-walk approach is clearly superior to the unlimited k-walk with respect to CPU time. It can process a graph of 100,000 nodes with 10 nodes of interest and a maximal walk length of 1,000 steps in 30 minutes on a standard PC. Relevant subgraphs of the largest connected component (18,297 nodes and 53,248 edges) of the KEGG metabolic network [Goto, 00] can be extracted in 3 seconds, while considering 5 randomly chosen nodes of interest and $L_{max} = 50$. This walk length is also shown to approximate very well the unlimited case.

Additional variants of the proposed methods consider clusters of nodes of interest. Our future work includes several aspects described below.

Some of the practical experiments described in section 8 were conducted on metabolic networks. These experiments answer questions related to the amount of captured information in the extracted subgraphs or the respective results of the limited versus unlimited k-walk methods. Average results were reported here with nodes of interest chosen at random. From a biological viewpoint the nodes of interest are far from random though. The proposed methods could be tested with carefully chosen nodes of interest. The extracted subgraphs could then be compared with known metabolic pathways. Interestingly, these pathways are not necessarily restricted to single paths but several alternatives are often considered to define a biologically relevant subgraph. The initial edge weights should be defined in a meaningful way in this context, possibly according to the enthalpy of the various reactions represented by the nodes of the network. Such an approach would then help to predict relevant subgraphs which are not yet identified as known pathways.

The limited k-walk approach is efficient since it scales linearly with k, the graph size and the maximal walk length. Yet it may require to be adapted for very large graphs by decomposing the computation over several predefined subgraphs and by combining the results. One could investigate whether some edges with very high relevances can be rapidly detected and serve as seeds of this process.

Another open issue is the sensitivity of the k-walk approach to the insertion or removal of an edge in the graph or, more finely, to some modification of its weight. Fast computation of the extracted subgraph after such a modification is a related issue.

Up to now, the nodes of interest in the original graph are assumed to be given. The sensitivity analysis mentioned above could determine for which nodes of interest the results are more or less stable. A different but related problem is to find which nodes of interest have the highest influence overall in the graph. Some existing works precisely address this issue, motivated by the design of viral marketing strategies in social networks [Richardso, 02]. Finding the most influential nodes, according to standard models of information diffusion in social networks, is known to be a NP-hard optimization problem. However an efficient greedy strategy providing a good approximate solution has been proposed [Kempe, 03]. It would be interesting to study whether the k-walk approach could be adapted to this objective.

Information diffusion in graphs has also been studied with kernel methods [Kondor, 02]. Such a kernel defines a similarity measure between graph nodes. This kernel is related to a *lazy* random walk model. The random walker in this model has a certain probability, at each time step, to stay in place (even if the graph does not include self loops) rather than pursuing her walk. One could possibly extend this approach by defining a similarity measure between graph nodes which would become relative to some nodes of interest.

A diffusion kernel is also strongly related to the graph Laplacian which plays a central role in spectral graph theory [Chung, 97]. In particular, Shi and Malik introduced a *normalized cut* criterion to define an optimal bipartitioning of a graph [Shi, 00]. They proposed a spectral segmentation approach based on the normalized Laplacian to approximate this problem. Meila and Shi also present links between this spectral segmentation and random walks in a Markov chain [Meila, 01]. Random walks are also used in the MCL algorithm for graph clustering [vD, 00]. Another line of work would extend these works by considering an optimal partitioning or clustering of a graph relative to some predefined nodes of interest.

References

[Brandes, 05] Brandes (U.) et Erlebach (T.) (édité par). – *Network Analysis: Methodological Foundations.* – Springer, 2005, *Lecture Notes in Computer Science.*

[Chung, 97] Chung (F.R.). – *Spectral graph theory.* – American Mathematical Society, 1997.

[Deville, 03] Deville (Yves), Gilbert (David), van Helden (Jacques) et Wodak (Shoshana J.). – An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, vol. 4, n° 3, 2003, pp. 246–259.

[Doyle, 84] Doyle (P.) et Snell (J.). – *Random walks and electric networks.* – The Mathematical Association of America, 1984.

[Eppstein, 99] Eppstein (D.). – Finding the k shortest paths. *SIAM Journal on Computing*, vol. 28, n° 2, 1999, pp. 652–673.

[Faloutsos, 04] Faloutsos (C.), McCurley (K.) et Tomkins (A.). – Fast discovery of connection subgraphs. *10th ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 118–127. – August 2004.

[Freeman, 77] Freeman (L.C.). – A set of measures of centrality based upon betweenness. *Sociometry*, vol. 40, 1977, pp. 35–41.

[Garey, 79] Garey (M.R.) et Johnson (D.S.). – *Computers and Intractability: A guide to the theory of NP-Completenes.* – San Francisco, Freeman and Company, 1979.

[Goto, 00] Goto (S.), Nishioka (T.) et Kanehisa (M.). – Ligand: chemical database of enzyme reactions. *Nucleic Acids Research*, vol. 28, n° 1, 2000, pp. 380–382.

[Hwang, 92] Hwang (F.K.), Richards (D.S.) et Winter (P.). – *The Steiner Tree Problem.* – Elsevier, North-Holland, 1992, *Annals of Discrete Mathematics*, volume 53.

[Jiménez, 99] Jiménez (Victor M.) et Marzal (Andrés). – Computing the k shortest paths: A new algorithm and an experimental comparison. *Algorithm Engineering: 3rd International Workshop, WAE'99.* pp. 15–29. – London, UK, 1999.

[Kemeny, 83] Kemeny (J.G.) et Snell (J.L.). – *Finite Markov Chains.* – Springer-Verlag, 1983.

[Kempe, 03] Kempe (D.), Kleinberg (J.) et Tardos (E.). – Maximizing the spread of influence trhough a social network. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 137–146. – Washington, DC, USA, 2003.

[Kondor, 02] Kondor (R.) et Lafferty (J.). – Diffusion kernels on graphs and other discrete input spaces. *Proc. of the Nineteenth International Conference on Machine Learning*, pp. 316–322. – 2002.

[Meila, 01] Meila (M.) et Shi (J.). – A random walks view of spectral segmentation. *Proc. of AISTATS.* – 2001.

[Meyer, 00] Meyer (Carl D.). – *Matrix analysis and applied linear algebra.* – Society for Industrial and Applied Mathematics, 2000.

[Newman, 05] Newman (M.E.J.). – A measure of betweenness centrality based on random walks. *Social networks*, vol. 27, 2005, pp. 39–54.

[Norris, 97] Norris (J. R.). – *Markov Chains.* – United Kingdom, Cambridge University Press, 1997.

[Rabiner, 93] Rabiner (L.) et Juang (B.-H.). – *Fundamentals of Speech Recognition.* – Prentice-Hall, 1993.

[Richardso, 02] Richardson (M.) et Domingos (P.). – Mining knowledge-sharing sites for viral marketing. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 61–70. – 2002.

[Shi, 00] Shi (J.) et Malik (J.). – Normalised cuts and image segmentation. *IEEE Transactions on Pattern Matching and Machine Intelligence*, vol. 22, August 2000, pp. 888–905.

[vD, 00] van Dongen (S.). – *A cluster algorithm for graphs.* – Technical Report n° INS-R0010, Amsterdam, Netherlands, Centrum voor Wiskunde and Informatica, May 2000.

[Viger, 05] Viger (Fabien) et Latapy (Matthieu). – Efficient and simple generation of random simple connected graphs with prescribed degree sequence. *COCOON*, pp. 440–449. – 2005.

[Warme, 00] Warme (D.M.), Winter (P.) et Zachariasen (M.). – *Advances in Steiner Tree*, chap. Exact Algorithms for Plane Steiner Tree Problems: A Computational Study, pp. 81–116. – Kluwer Academic Publishers, 2000.